



**INSTITUTO POLITÉCNICO NACIONAL**



**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA**  
**SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**  
**MAESTRÍA EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

*“Development of a smart metering unit designed to identify energy theft on distribution networks”*

# TESIS

**Que para obtener el grado de maestro en ciencias  
en ingeniería eléctrica**

*Presenta*

**ING. DAVID JONATHAN SEBASTIÁN CÁRDENAS**

*Director de tesis*

**DR. RICARDO MOTA PALOMINO**

**MÉXICO D.F., ENERO DE 2015**

**DISCLAIMERS:**

PARTS OF THIS WORK CONTAIN SENSITIVE INFORMATION REGARDING ELECTRICAL UTILITIES METERING INFRASTRUCTURE, THE AUTHOR SHALL NOT BE RESPONSIBLE FOR MISUSE OR WRONGDOING DERIVED FROM THIS WORK. SIMILARLY, PARTS OF THIS WORK CONTAIN REVERSE ENGINEERING TECHNIQUES, WHICH MIGHT BE ILLEGAL DEPENDING ON THE JURISDICTION; THE AUTHOR ONLY PERFORMS THEM WITHIN EDUCATIONAL BOUNDS.

**COPYRIGHT NOTICE**

PARTS OF THIS WORK ARE UNDER A PATENTING PROCESS, USERS SHOULD CHECK WITH AUTHOR BEFORE CREATING DERIVED WORKS.



# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

### ACTA DE REVISIÓN DE TESIS

En la Ciudad de México D. F. siendo las 13:00 horas del día 5 del mes de Diciembre del 2014 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de E.S.I.M.E. – ZAC. para examinar la tesis titulada:

**DEVELOPMENT OF A SMART METERING UNIT DESIGNED TO IDENTIFY THEFT  
ON DISTRIBUTION NETWORKS**

Presentada por el alumno:

**SEBASTIÁN**

**CÁRDENAS**

**DAVID JONATHAN**

Apellido paterno

Apellido materno

Nombre(s)

Con registro: 

B	1	2	0	9	0	6
---	---	---	---	---	---	---

aspirante de:

**MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

Después de intercambiar opiniones, los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISIÓN REVISORA

Director(a) de tesis

**DR RICARDO OCTAVIO MOTA PALOMINO**

Presidente

Secretario

**DR. DANIEL OLGUÍN SALINAS**

Segundo Vocal

**DR. DAVID ROMERO ROMERO**

Tercer Vocal

**DR. MOISÉS SALINAS ROSALES**

**DR. RAÚL ÁNGEL CORTÉS MATEOS**

PRESIDENTE DEL COLEGIO DE PROFESORES

**DR. MAURO ALBERTO ENCISO AGUILAR**





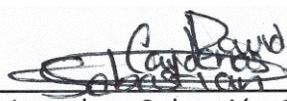


**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA DE CESIÓN DE DERECHOS**

En la Ciudad de México D. F., el día 15 del mes de Diciembre del año 2014, el que suscribe David Jonathan Sebastián Cárdenas, alumno del Programa de Maestría en Ciencias en Ingeniería Eléctrica con número de registro B120906, adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME-Zacatenco del IPN, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Ricardo Octavio Mota Palomino y cede los derechos del trabajo titulado Development of a smart metering unit designed to identify energy theft on distribution networks, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección **jonathans@live.com y/o rmotap@ipn.mx**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

  
\_\_\_\_\_  
David Jonathan Sebastián Cárdenas



## RESUMEN

En la última década las tecnologías de medición inteligente se han hecho presentes en las redes de distribución. En específico las tecnologías de medición inteligente han sido instaladas para permitir el monitoreo de la energía y ejecución de servicios de valor añadido por medio de redes de comunicación. Por otro lado, en los últimos años ha surgido el interés sobre la privacidad de datos y asuntos relacionados con la seguridad cibernética de estos dispositivos. Las cuestiones de privacidad de datos y la necesidad de elevar la eficiencia energética han hecho que los medidores inteligentes sean un tema de controversia, entre otras, debido a nuevas aplicaciones basadas en la minería de datos y posibles escenarios de ataque que han sido expuestos año tras año.

En esta tesis se presenta el diseño y desarrollo de un dispositivo de medición inteligente, el cual desde su planeación incluyó mecanismos para cuidar la seguridad de la información y permitir la detección del robo de energía mediante la inclusión de hardware específico. El prototipo implementado cumple con el estándar IEEE 1459, que se utiliza para cuantificar los efectos de las cargas no lineales en sistemas de medición residenciales. También cumple con los requerimientos de la clase “M” para la operación en estado estable de una unidad de medición fasorial (IEEE C37.118) mediante la inclusión de una unidad receptora de GPS.

El dispositivo cumple parcialmente con el estándar IEEE 802.15.4g, este estándar está diseñado para regular la comunicación inalámbrica en redes de medición inteligente. Sin embargo, los límites de comunicación inalámbrica fueron excedidos con la finalidad de permitir las comunicaciones basadas en TCP/IP. Desde el punto de vista de la seguridad, este dispositivo implementa Transport Layer Security (TLS) basado en el uso de AES-128. También cuenta con mecanismos de protección de credenciales de seguridad almacenadas en el dispositivo, mediante la inclusión de una Physically Unclonable Function (PUF).

Finalmente, las características anteriormente mencionadas fueron fusionadas para desarrollar una unidad de medición de inteligente que sea capaz de identificar y cuantificar el robo de energía. El algoritmo propuesto emplea la descomposición armónica en tiempo real para realizar un balance de corrientes basado en el principio de Kirchhoff entre los medidores instalados en el lado del usuario y un agente observador ubicado en transformador de distribución. El algoritmo propuesto

fue probado considerando un conjunto de escenarios donde se realiza el análisis de patrones de consumo a lo largo de varios días. Los resultados obtenidos muestran que es posible identificar a los medidores alterados.

## **ABSTRACT**

Smart metering technologies have rapidly started to appear in distribution networks in the last decade. Specifically smart metering technologies have been installed on the user premises to enable network-based energy monitoring as well as deployment of associated services. However, on the last few years there has been a public interest for data privacy and cyber security relating these devices. These data privacy concerns and needs to raise energy efficiency have made smart meters a controversial topic, among others fueled by new data mining applications and theoretical attack scenarios being exposed each year.

In this thesis a smart meter device is designed and built. This unit since its planning stages has included mechanisms targeted to ensure information security and enable energy theft detection capabilities by including application-specific hardware. The implemented design complies with IEEE 1459 standard, a standard used to assess the effects of non-linear loads on residential metering installations. It also complies with class “M” of the steady state requirements for a PMU unit by including a GPS receiver unit (IEEE C37.118).

The device is partially compliant with IEEE 802.15.4g standard, a standard intended to provide data interoperability in smart metering networks. However, in this implementation the wireless data rate limit was overridden in order to enable fast TCP/IP communications. From the security viewpoint, the device implements Transport Layer Security (TLS) communications based on Advanced Encryption Standard (AES-128) primitives. It also includes a custom Physically Unclonable function (PUF) designed to protect the security credentials stored on the device.

Finally, the previously mentioned characteristics were fused together in order to develop a smart metering unit capable of identifying and assessing energy theft. The proposed energy theft algorithm relies on the harmonic decomposition to perform time-synchronized Kirchhoff-based current balancing between the energy meter located at the user premises and a central observer located on the distribution transformer. The developed algorithm was simulated under a variety of scenarios and is able to pinpoint altered meters by analyzing consumption patterns across several days. The results indicate that it is possible to identify altered meters by using the proposed methodology.



## ACKNOWLEDGMENTS

I would like to express my gratitude to all the members of the graduate section for providing an excellent and inspiring working atmosphere, especially the professors who have taught me along the way during my studies.

I would like to thank my thesis advisor, Dr. Ricardo Mota for his life-long support to my family. I am thankful for his aspiring guidance, invaluable constructive criticism, friendly advice and for his financial support during the project development.

I wish to thank the members of my thesis committee: Dr. Daniel Olguín, Dr. David Romero, Dr. Moisés Salinas, Dr. Raúl Cortés and Dr. Pablo Gómez for generously offering their time, support, guidance and valuable comments regarding this document.

I would also like to thank Dr. Pablo Gómez and Dr. Mohamed Badaoui for their extensive proofreading checks, without whose patience and careful reading, this thesis might have been still more unreadable to English readers.

I am also very grateful to CONACYT and IPN for supporting my studies during all of these years, as well as companies that donated equipment like Microchip™ and Texas Instruments.

Finally, I would like to thank my parents for their emotional and financial support during my entire life, especially to my dad who has been deeply involved during the course of this work.



## TABLE OF CONTENTS

RESUMEN	vi
ABSTRACT	viii
List of Figures	xviii
List of Tables	xxiv
Lists of Used Abbreviations	xxvi
Publications from the Study	xxx
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Presentation	1
1.2 Motivation	2
1.2.1 Energy Losses	2
1.2.2 Secure Data Communications	3
1.3 Objective	4
1.4 Secondary Objectives	4
1.5 Thesis Scope and Limitations	4
1.6 Contributions	5
1.7 Thesis Outline	6
1.8 State of the art	7
1.8.1 Smart grids	7
1.8.2 Smart grid objectives	8
1.8.3 Smart grid projects	10
1.8.3.1 Smart grid –EPRI sponsored Projects	10
1.8.3.2 Smart grid – E2SG sponsored Projects	13
1.8.3.3 Smart grid – Non sponsored Projects	14
1.8.4 Prior Works on Energy Theft Detection	17
1.8.4.1 Introduction	17
1.8.4.2 AMR based theft detection	18
1.8.4.3 Theft detection in Smart Meters	19
1.8.4.4 Theft detection based on power balance	20
1.9 Overview of Technologies Required For the Smart Grid Metering	21
1.9.1 Advanced Meter Infrastructure (AMI)	21
1.9.2 Communication standards	22
1.9.3 Information security	22
1.10 Smart Meter Components	22
<b>2. SIGNAL ACQUISITION AND ENERGY METERING FOR SMART METERS</b>	<b>25</b>
2.1 Analog to Digital Conversion	25
2.2 Types of ADC	25
2.2.1 Ramp ADC	25
2.2.2 Successive Approximation ADC	26
2.2.3 Delta Sigma ADC	26
2.3 Filters	30
2.3.1 Low pass filters	31
2.3.2 Butterworth	31
2.3.3 Bessel	31
2.3.4 Active filters	32
2.3.5 Key-Sallen topology	32
2.4 Fourier Series	34
2.5 Harmonics	35
2.6 Discrete Fourier Transform	36
2.7 Fast Fourier Transform	37
2.8 Energy Metering	38

<b>2.9</b>	<b>IEEE 1459</b>	<b>39</b>
2.9.1	RMS Voltage- Discrete Time domain	40
2.9.1	RMS Current- Discrete Time domain	40
2.9.2	Harmonic current and voltage components	41
2.9.3	Total Harmonic Distortion (THD)	41
2.9.4	Active Power	41
2.9.5	Reactive power	42
2.9.6	Fundamental apparent power ( $S_1$ )	42
2.9.7	Non-fundamental apparent power ( $S_N$ )	42
2.9.8	Other quantities described by IEEE 1459	42
2.9.9	Apparent power	43
2.9.10	Fundamental power factor	43
2.9.11	Power factor	43
2.9.12	Vector apparent power (3-phase systems)	44
<b>2.10</b>	<b>Phasor Measurements Units</b>	<b>44</b>
2.10.1	Hardware components	45
2.10.2	Phasor signal representation	45
2.10.3	IEEE C37.118	46
<b>3.</b>	<b>DATA SECURITY IN COMMUNICATIONS.</b>	<b>49</b>
<b>3.1</b>	<b>Introduction</b>	<b>49</b>
<b>3.2</b>	<b>Crypto Elements Implemented for This Thesis</b>	<b>50</b>
<b>3.3</b>	<b>Cryptographic Terms</b>	<b>51</b>
<b>3.4</b>	<b>Cryptography Modes</b>	<b>51</b>
3.4.1	Symmetric-key cryptography	51
3.4.1.1	Crypto Elements of symmetric-key cryptography	52
3.4.1.2	Advanced encryption standard	52
3.4.2	Asymmetric-Key cryptography	53
<b>3.5</b>	<b>Attacks on Cryptographic Security Implementations</b>	<b>54</b>
3.5.1	Incorrect cipher mode of operation	54
3.5.2	Not using MAC or authentication cipher modes.	55
3.5.1	Timing attacks	56
3.5.1.1	MAC verification timing attack	56
3.5.1.2	Cache memory attacks	57
3.5.2	Reverse engineering	58
3.5.2.1	Software attacks	58
3.5.2.2	Hardware attacks	60
<b>3.6</b>	<b>Smart Meter Security</b>	<b>62</b>
<b>3.7</b>	<b>Proposal for securing key storage in microcontrollers.</b>	<b>65</b>
3.7.1	Key management	65
3.7.2	PUF in Microcontrollers	66
3.7.3	PUF extraction function	67
3.7.3.1	Error repetition codes	68
3.7.4	Experimental development of the PUF function.	69
3.7.4.1	Proposed PUF extraction algorithm	72
3.7.4.2	Results evaluation	74
3.7.4.3	Further improvements on the PUF function	78
<b>4.</b>	<b>DIGITAL COMMUNICATIONS IN SMART METERING NETWORKS</b>	<b>79</b>
<b>4.1</b>	<b>Introduction</b>	<b>79</b>
<b>4.2</b>	<b>Wireless Communications</b>	<b>80</b>
4.2.1	Fundamentals of RF	80
4.2.1.1	Power measuring in RF	81
4.2.1.2	The channel	82

4.2.1.3	The modulation	83
<b>4.3</b>	<b>Issues with Respect Wireless Communications</b>	<b>83</b>
4.3.1	Clear Channel Assessment	84
4.3.2	The hidden node problem	84
4.3.3	Carrier Sense Multiple Access with Collision Avoidance	84
<b>4.4</b>	<b>The Open Systems Interconnection model</b>	<b>85</b>
<b>4.5</b>	<b>The IEEE 802.1x Family of Standards</b>	<b>87</b>
4.5.1	The IEEE 802.15 family of standards	88
4.5.1.1	The IEEE 802.15.4 Standard	88
4.5.2	The 2.4 GHz ISM band	90
4.5.3	Physical layer properties	91
4.5.3.1	Physical layer terms	91
4.5.4	The data link layer properties	92
4.5.4.1	The MAC Frame on IEEE 802.15.4g	92
4.5.4.2	The LLC layer	95
4.5.4.3	The superframe format	97
<b>4.6</b>	<b>Wireless Data Communication Interface Development.</b>	<b>97</b>
4.6.1	Introduction	98
4.6.2	Electrical design considerations	99
4.6.2.1	PCB Design Guidelines	100
4.6.3	Software based radio handling	101
4.6.3.1	Compatibility of the proprietary MAC header and IEEE MAC header.	103
<b>4.7</b>	<b>Wired Communications Interface Development</b>	<b>104</b>
<b>4.8</b>	<b>TCP/IP IMPLEMENTATION</b>	<b>106</b>
<b>4.9</b>	<b>TLS</b>	<b>107</b>
<b>4.10</b>	<b>Attacks on Smart Meter Network Infrastructure.</b>	<b>108</b>
<b>5.</b>	<b>LOSS ASSESSMENT ON DISTRIBUTION NETWORKS</b>	<b>109</b>
<b>5.1</b>	<b>Introduction</b>	<b>109</b>
<b>5.2</b>	<b>The Mexican Electricity Market</b>	<b>109</b>
<b>5.3</b>	<b>Energy Theft in the Mexican Market</b>	<b>110</b>
<b>5.4</b>	<b>An Overview of Energy Theft Techniques</b>	<b>113</b>
5.4.1	Attacks on the security seals	114
5.4.2	Alteration of feeding circuits	122
5.4.2.1	Preamble	122
5.4.2.2	The current inversion method	123
5.4.2.3	Opened return line	124
5.4.2.4	Current coil bypass.	124
5.4.3	Mechanical meter tampering	125
5.4.3.1	Lowering the rotational speed.	126
5.4.3.2	Disc breaking methods	127
5.4.3.3	Dynamic disc breaking	128
5.4.4	Electronic meter tampering	128
5.4.4.1	TC Burden modification	129
5.4.4.2	RTC clock alterations	130
5.4.4.3	Master clock alterations	130
5.4.4.4	Communication channel alterations	131
5.4.5	Smart meter attacks-the future.	131
<b>5.5</b>	<b>A Central Observer Technique Based On the Harmonic Content, For Energy Theft Assessment</b>	<b>132</b>
5.5.1	Introduction	132
5.5.2	Algorithm development	133
5.5.3	Simulation preamble	139
5.5.3.1	Energy theft simulation procedure	143

5.5.4	Simulation Results	145
5.5.4.1	Case 1. More than 20 users connected to a monophasic transformer, with three altered units.	146
5.5.4.1	Case 2. More than 15 users connected to a monophasic transformer, with two altered units.	148
5.5.4.2	Case 3. More than 15 users connected to a single phase, with one altered unit.	149
5.5.4.3	Case 4. More than 45 users connected to a single phase, with 5 altered units	150
<b>6.</b>	<b>PROPOSED SMART METER ARCHITECTURE</b>	<b>153</b>
<b>6.1</b>	<b>Introduction</b>	<b>153</b>
<b>6.2</b>	<b>Hardware Selection</b>	<b>153</b>
6.2.1	Election of a Computing Unit for a Smart Meter	154
<b>6.3</b>	<b>MIPS Architecture</b>	<b>155</b>
6.3.1	MIPS instruction set	155
6.3.2	Pipelining	156
6.3.2.1	Pipeline Problems	157
6.3.3	Memory architecture	159
6.3.3.1	Cache memory	159
<b>6.4</b>	<b>The PIC32MZ Architecture</b>	<b>160</b>
6.4.1	The PIC32MZ pipeline	161
6.4.2	PIC32MZ interrupts	163
6.4.3	PIC32MZ Assembly language	164
6.4.4	Cache memory in the PIC32MZ	164
6.4.5	Direct Memory Access (DMA)	164
6.4.6	Bus access groups	165
6.4.7	Description of commonly hardware modules	166
6.4.8	PIC32MZ limitations.	167
<b>6.5</b>	<b>Proposed Smart Meter Architecture</b>	<b>167</b>
6.5.1	Dual microcontroller setup	168
6.5.2	Analog to Digital conversion hardware implementation.	169
6.5.2.1	Accurate voltage reference	170
6.5.2.2	Differential voltage signal acquisition	172
6.5.2.3	Differential current signal acquisition	174
6.5.3	Frequency measurement-Hardware Implementation	175
6.5.3.1	Analog buffer	176
6.5.3.2	Sallen Key Filter	176
6.5.3.3	DC component Removal	178
6.5.3.4	Square wave generator	179
6.5.4	Dynamic signal sampling mechanism-Hardware Implementation	179
6.5.5	DFT implementation	181
6.5.5.1	Analysis of the required operations needed by FFT/DFT implementations.	181
6.5.5.2	Optimization of a cross correlation function to compute the DFT	183
6.5.5.3	DFT based signal decomposition-The square root problem	185
6.5.5.4	DFT validation (Digital simulation)	188
6.5.6	IEEE 1459-2000 implementation	191
6.5.7	IEEE C37.118 implementation	191
6.5.7.1	GPS time signal characterization	192
6.5.7.2	Phasor measurement unit testing procedure.	194
6.5.8	I/O control	197
6.5.9	PUF generation and recovery algorithm	199
6.5.10	AES implementation	200
6.5.11	Dynamic frequency measurement-Signal Validation	202
6.5.11.1	Frequency measurement circuit	202
6.5.11.2	PLL validation	207
6.5.12	Microcontroller operating system.	210

6.5.13	Smart meter communications platform.	211
6.5.13.1	Embedded HTTP server tests.	213
6.5.13.2	Assembled Hardware Modules	216
<b>7.</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>219</b>
7.1	Conclusions	219
7.2	Future work ideas	222
<b>A.</b>	<b>SMART METER CIRCUITS</b>	<b>233</b>
<b>B.</b>	<b>ADC TERMS USED ON THIS WORK</b>	<b>243</b>
<b>C.</b>	<b>PHASE LOCK DEVICES</b>	<b>245</b>
C.1.	Clocks	245
C.1.1.	Clock generation	246
<b>D.</b>	<b>COMMON TERMS USED IN CRYPTOGRAPHY</b>	<b>249</b>
<b>E.</b>	<b>ADVANCED ENCRYPTION STANDARD.</b>	<b>251</b>
E.1.	AES Galois field( $2^8$ )	251
E.2.	AES general algorithm description	252
E.2.1.	AES setup process	252
E.2.1.1.	Message parsing	252
E.2.1.2.	Key Setup	253
E.2.1.3.	Key scheduling	253
E.2.2.	AES round operation	254
E.2.2.1.	SubBytes function	254
E.2.2.2.	ShiftRows	255
E.2.2.3.	MixColumns	255
E.2.2.4.	Key XORing	255
E.2.2.5.	Round operation epilogue	256
E.3.	AES optimization on matrix multiplication	256
<b>F.</b>	<b>BLOCK CIPHER ENCRYPTION MODES</b>	<b>259</b>
F.1.	Electronic Code Book	259
F.2.	Cipher-Block Chaining	259
F.3.	Counter Mode	260
<b>G.</b>	<b>AUTHENTICATION FUNCTIONS</b>	<b>263</b>
G.1.	Message Integrity	263
G.1.1.	Hash functions	263
G.1.1.1.	One way functions	264
G.1.1.2.	Collision resistance	266
G.1.2.	Secure cryptographic hash functions.	267
G.1.2.1.	Message Digest Algorithm-5	268
G.1.2.2.	Secure Hashing Algorithm-1 (SHA 1)	270
G.1.2.3.	Secure Hashing Algorithm-2 (SHA-2)	271
G.1.3.	Message Authentication Codes	273
G.1.3.1.	Hash based Message Authentication Codes	274
G.1.3.2.	Block cipher authentication modes	275
G.1.3.3.	Cipher-based message authentication	275
<b>H.</b>	<b>DIGITAL COMMUNICATIONS</b>	<b>277</b>
H.1.1.	Digital modulation	277
H.1.1.1.	Preamble	278
H.1.1.2.	Amplitude-Shift Keying	279
H.1.1.3.	Phase shift keying	280
H.1.1.4.	Digital demodulation	284
H.1.1.5.	The correlation receiver	284
H.1.1.6.	Direct-Sequence Spread Spectrum	287

<b>I. IEEE 802.15.4G PHYSICAL LAYER FORMAT</b>	<b>289</b>
I.1. The physical layer frame format	289
I.1.1. Forward Error Correction (FEC)	290
I.1.2. Interleaving	292
I.1.3. Bit Differential Encoding (BDE)	294
I.1.4. DSSS mapping	294
I.1.5. Pilot insertion	295
I.2. Clear Channel Assesment (CCA)	296
<b>J. COMPUTER ARCHITECTURE</b>	<b>297</b>
J.1. CPU Types	297
J.1.1. Desktop	297
J.1.2. Microcontrollers.	297
J.1.3. System on Chip	298
J.2. System buses	298
J.3. Random Access Memory (RAM)	298
J.4. Static Random Access Memory, or Read Only Memories (SRAM-ROM)	298
J.5. Von Newman Architecture	298
J.6. Harvard Architecture	298
J.7. Performance	299
J.8. Interrupts	299
<b>K. THE SIMPLEX ALGORITHM</b>	<b>301</b>
K.1. Linear Programs in Standard Form	301
K.2. Generalities for solving linear problems by using a matrix based approach	302
K.3. The simplex method algorithm by using the Tableau Format	303
K.3.1. Additions to the basic simplex algorithm	304

## List of Figures

Figure 1-1 E2SG smart grid vision projects, adapted from [20] .....	14
Figure 1-2 Smart meter architecture deployed by CFE in Mexico City, adapted from [26].....	16
Figure 1-3. The power balance diagram used by the authors on [27] .....	20
Figure 1-4 Functional block of a smart meter, adapted from [2].....	23
Figure 1-5 Basic Elements of a smart meter, adapted from [45] .....	23
Figure 2-1 Ramp ADC theory of operation. ....	26
Figure 2-2 SAR ADC theory of operation. ....	26
Figure 2-3 Components of a Sigma-Delta ADC, adapted from [46] . ....	27
Figure 2-4 Sample Quantization of a Sigma-Delta ADC, integrator vs input signal.....	27
Figure 2-5 Sample Quantization of a Sigma-Delta ADC, outputting 4 Bits.....	28
Figure 2-6 Sample Quantization of a Sigma-Delta ADC, outputting 5 Bits.....	29
Figure 2-7 Delta-sigma conversion speed limit, adapted from [46].....	30
Figure 2-8 Group delay characteristics, for eight order filters given a square input, adapted from [47].....	30
Figure 2-9 Fundamental first order active filter circuits.....	32
Figure 2-10 Basic Second order Sallen-Key filter topology .....	33
Figure 2-11 Modified third order Sallen-Key filter topology using a single Op-Amp .....	33
Figure 2-12 Square wave decomposition, first four odd integer frequencies, adapted from [56] .....	35
Figure 2-13. The 3D approach to explain S, according to IEEE 1459.....	43
Figure 2-14 Arithmetic apparent power ( $SA$ ) and Vector Apparent power ( $SV$ ) under unbalanced non-sinusoidal conditions, adapted from [61].....	44
Figure 2-15. Basic hardware components of PMU units. ....	45
Figure 2-16 The TVE concept, graphical representation. ....	47
Figure 3-1 Uses of encryption modes, and possible pitfalls.....	54
Figure 3-2 Attack Based on the ECB mode, or static/weak IV on CTR mode .....	55
Figure 3-3 CBC attack based on IV masking, for reference the internal CBC decryption mode of operation is shown. ....	56
Figure 3-4 MAC verification timing attacks, an attacker successively test bytes from the MSByte up to LSByte, if the byte is correct the response time increases. ....	57
Figure 3-5 Entropy of a sample program, containing an asymmetric key located at the peak entropy point, taken from [77].....	59
Figure 3-6 Communications sniffing, taken from [77].....	61
Figure 3-7 Professional vs open source sniffer visualization tools.....	61
Figure 3-8 Chemical etching to expose silicon die, on a PIC32MZ (left) and PIC32MX (right) microcontroller, adapted from [78] .....	62
Figure 2-9 Example of load identification, from a detailed demand profile for a single user, taken from [82], [83].....	64
Figure 3-10 Typical T6 SRAM layout, that uses balanced pair communications to improve noise characteristics. ....	67
Figure 3-11 Internal silicon layout for the PIC32MZ series device (left) and PIC32MX (right), showing the SRAM layout, known as T6-doughnut, taken from [78] .....	69
Figure 3-12 Methodology used to extract data from the PIC32MZ unit.....	70
Figure 3-13 Bit set count for each byte, in device 0.....	73
Figure 3-14 Bit set count for each byte, in device 1 .....	74
Figure 3-15 A visual representation of the generated pattern vs the one coming out of a PRNG. ....	76
Figure 3-16 Autocorrelation function for device #0.....	77
Figure 3-17 Autocorrelation function for device #1.....	77
Figure 3-18 Proposed PUF verification method. ....	78

Figure 4-1 Basic communication diagram showing the main stages of wireless communications. ....	81
Figure 4-2. The hidden node problem, node B can see both A and C; but C and A units cannot see each other. .	84
Figure 4-3 The CSMA/CA algorithm, Based on [10], [94]. ....	85
Figure 4-4 Data transmission path in the OSI reference model, adapted from [93]. ....	86
Figure 4-5 The IEEE 802.11 and 802.15.4 RF spectrum adapted from [98]. ....	90
Figure 4-6 The MAC Frame structure [99]. ....	92
Figure 4-7 The MAC header component of the frame structure [99]. ....	93
Figure 4-8 The IEEE 802.15.4g link configurations [99]. ....	96
Figure 4-9 The IEEE 802.15.4g superframe format [102] . ....	97
Figure 4-10 The MRF24XA Transceiver internal architecture, based on the architecture described on [100] .....	99
Figure 4-11 The MRF24XA electrical diagram, with external components attached, own design based on [103] .....	100
Figure 4-12. Manufactured PCB layout. ....	101
Figure 4-13 The MRF24XA unit mounted on custom made PCB.....	101
Figure 4-14 The MRF24XA register mapping, physical address space [103].....	102
Figure 4-15 Register-based configurations required to handle 802.15.4g thru the MPU .....	103
Figure 4-16 The MRF24XA proprietary MAC header in compatibility mode with IEEE 802.15.4g.....	104
Figure 4-17 The LAN8740A Ethernet PHY mounted in a plug-in module.....	105
Figure 4-18 The LAN8740A pins used to communicate with the microcontroller.....	105
Figure 4-19 The TCP/IP stack available on the cyclone TCP distribution .....	106
Figure 4-20 The TLS key interchange mechanism .....	107
Figure 5-1 Mexican electric utility reported losses through the years [108]. ....	111
Figure 5-2. An example of direct wire taping of air wires in many flea markets around the city of Mexico. ....	114
Figure 5-3 Due to their multipurpose nature, tamper-evident seals are widely available to the general public. ....	115
Figure 5-4 Tamper proof mechanism bypassing, method disclosed by a tamper-proof device manufacturer [112]. .....	115
Figure 5-5 A tamper-resistant padlock with a transparent enclosure. ....	116
Figure 5-6 A tamper-resistant padlock mechanism bypassed by inserting a pin through an inferior cavity.....	117
Figure 5-7. The bypassing of tamper-resistant padlock mechanism, by perpetrating minimum damage. ....	117
Figure 5-8 Reuse of tamper-resistant padlock mechanism, with no visible damage.....	118
Figure 5-9 Forgery of tamper proof seals, with the presence of valid ID fields. ....	119
Figure 5-10 A custom build tool to forge seal IDs, based on an open source QR generator. ....	119
Figure 5-11 A bolt type tamper-proof seal, with printed ID.....	120
Figure 5-12 A mechanical barrel lock mechanism for meter locking rings, with its associated tool. ....	120
Figure 5-13. An electronic authorization key mechanism for barrel locks.....	121
Figure 5-14 A next generation boxed meter lock (pre-production sample), with RFID key based mechanism, side view. ....	121
Figure 5-15 RFID boxed meter lock (pre-production sample), front view. ....	122
Figure 5-16. Internal wiring of a single-phase watt-hour meter, based on the S1 form.....	123
Figure 5-17 A correctly wired metering circuit, where * denotes the instantaneous polarity. ....	123
Figure 5-18. An inverted metering circuit with an additional return path.....	124
Figure 5-19. An open neutral return circuit causes most-meters to measure a zero consumption.....	124
Figure 5-20 Jumper based current coil bypass.....	125
Figure 5-21 A current bypass method, hidden behind the terminal block. ....	125
Figure 5-22 Mechanical wattmeter tampering by means of disorientation. ....	126
Figure 5-23 Mechanical alteration that seeks to break the disk free rotation. ....	127
Figure 5-24 A destructive mechanical alteration that seeks to break the disk free rotation, by modifying the dial mechanism. ....	127

Figure 5-25. The presence of a hole in the meter case indicates a possible tampering case .....	128
Figure 5-26 Typical architecture of an electronic meter .....	128
Figure 5-27 The PCB layout of a production electronic meter. ....	129
Figure 5-28 Current transformer burden alteration. ....	130
Figure 5-29 RTC crystal alteration. ....	130
Figure 5-30 A clear sign of oscillator tampering.....	131
Figure 5-31. The proposed method diagram uses current balance. ....	133
Figure 5-32. The proposed method assumes that current waveform is composed of three user types.....	134
Figure 5-33 Modified Simplex algorithm used to obtain the $\beta_i$ factors .....	138
Figure 5-34 Weekly load profile obtained from a feeder in a southern state in Mexico.....	140
Figure 5-35 Sample daily current profile for three customers, with different average current consumption. ....	140
Figure 5-36 Sample daily P.F profile for three customers, where the existence of unique profiles can be observed.....	141
Figure 5-37 Sample customer current components for a single day current profile.....	142
Figure 5-38 Sample customer current angles during a single day. ....	143
Figure 5-39 Current waveform for two customers in two different days.....	143
Figure 5-40 Current waveform data capture, with added noise typical of a class 0.5% precision device (with a 2.5A testing current).....	144
Figure 6-1 Dual microcontroller smart meter architecture, adapted from [41].....	154
Figure 6-2 Pipeline approach as a parallel execution mode.....	157
Figure 6-3 Simplified pipeline architecture showing pipeline stalls .....	158
Figure 6-4 NUMA Memory hierarchy available on PIC32MZ a type of MIPS microcontroller, showing different memory capabilities. ....	159
Figure 6-5 Core elements of a PIC32MZ microcontroller .....	160
Figure 6-6 Detailed hardware architecture of the PIC32MZ unit.....	161
Figure 6-7 Simplified MIPS pipeline embedded in the PIC32MZ, adapted from [129]. ....	163
Figure 6-8 Cache coherency problems due to the DMA unit. ....	165
Figure 6-9 Simultaneous access group, i.e. initiators do not require bus arbitration schemes.....	166
Figure 6-10 Proposed smart meter architecture, featuring a dual microcontroller setup. ....	168
Figure 6-11 Typical shunt voltage reference architecture, adapted from [135] .....	171
Figure 6-12 Dual supply voltage references employed by the ADC circuit .....	172
Figure 6-13 ADC ranges under different operational modes .....	172
Figure 6-14 Ladder circuit used for differential voltage acquisition .....	173
Figure 6-15 Differential measurement circuit used for TC current acquisition via a burden resistor. ....	174
Figure 6-16 Actual signals (from a non-linear load) measured by the differential current channel.....	175
Figure 6-17 Frequency measurement device components.....	176
Figure 6-18 Designed buffered architecture. ....	176
Figure 6-19 Architecture of the third order Sallen-Key filter topology used to filter out high frequency components.....	177
Figure 6-20. Ideal vs hardware constrained Bessel filter magnitude characteristics.....	178
Figure 6-21. Ideal vs hardware constrained Bessel filter group time delay characteristics.....	178
Figure 6-22 DC filter used on the frequency measurement circuit. ....	179
Figure 6-23 Sinusoidal waveform to square waveform conversion circuit.....	179
Figure 6-24 Angle variation by a real frequency and sampling frequency mismatch, different signals are sampled at 60 Hz.....	180
Figure 6-25 PLL driven dynamic clock reference used to drive the ADC unit. ....	180
Figure 6-26 Internal PLL values used to simulate the PLL response under the clock design tool available at [137]. ....	181

Figure 6-27 DFT calculation algorithm, featuring a loop unrolling optimization, $N = 128$ .....	185
Figure 6-28 Fast Square root algorithm based on successive multiplications .....	186
Figure 6-29 Sample current and voltage signals proposed by [61] .....	188
Figure 6-30 Discretized test signals given by [61], expressed on % respect the nominal ADC channel configuration. ....	189
Figure 6-31 Time synchronization measurements obtained from two GPS units, showing time differences.....	193
Figure 6-32 Time difference between two GPS units (absolute) organized through a histogram.....	193
Figure 6-33 Absolute position error histogram reported by the US government, for a large number of samples [143] .....	194
Figure 6-34 Time-stamped measurement flowchart .....	195
Figure 6-35 Sample voltage found at the university premises.....	196
Figure 6-36 ADC measured signal (digitally added signal, based on the obtained readings). ....	196
Figure 6-37 Streaming data to devices using a DMA Unit.....	198
Figure 6-38 Sample LCD display by employing a DMA based video driver .....	199
Figure 6-39 PUF generation algorithm (Actual Firmware implementation) .....	200
Figure 6-40 Proposed AES implementation flow chart algorithm.....	201
Figure 6-41 The proposed AES implementation features a constant execution time, but could be susceptible to power analysis attacks.....	202
Figure 6-42 Typical frequency ranges and variations for the Mexican Electrical Interconnected Grid, obtained from .....	203
Figure 6-43 Frequency sweep applied to third order Bessel and Butterworth filter .....	203
Figure 6-44 Frequency measurement errors recorded during the frequency sweep used to study the Bessel filter response. ....	204
Figure 6-45 Signal response characteristics of the proposed Bessel filter.....	204
Figure 6-46 Group delay characteristics of the proposed filter under non-ideal component values.....	205
Figure 6-47 Actual filter response characteristics of the proposed Bessel filter. ....	206
Figure 6-48 Sample frequency measurements as outputted by the frequency measurement hardware. ....	206
Figure 6-49 Simultaneous waveform captures at different frequencies, that shows correct filter performance.....	207
Figure 6-50 PLL loopback control. ....	208
Figure 6-51 PLL output before applying frequency control .....	208
Figure 6-52 Amplified PLL output after applying frequency control.....	209
Figure 6-53 ADC_CLOCK output compared with the PLL clock signal .....	210
Figure 6-54 Web services provided by meters in CFE Polanco AMI project [150] .....	212
Figure 6-55 Web services provided by the developed meter. ....	212
Figure 6-56 Data transfer requirements for the proposed metering architecture. ....	213
Figure 6-57 HTML Server log transmitted over the optical port installed on the developed metering unit. ....	213
Figure 6-58 HTML Server connection reliability results by using an open source benchmark tool.....	214
Figure 6-59 Network architecture used to determine connectivity to an end user device from a HTTP client. ..	214
Figure 6-60. Sample HTTP content provided by the developed unit. ....	215
Figure 6-61 Rear view of the metering unit configured as a data concentrator (with Ethernet module).....	216
Figure 6-62 Front view of the designed metering unit with LCD and optical port mounted.....	216
Figure 6-63 Front view of the designed metering unit mounted inside an ANSI 1S sized container.....	217
Figure A-1 Proposed ADC circuit (Electrical Diagram). ....	233
Figure A-2 Proposed ADC circuit (PCB layout).....	234
Figure A-3 Proposed ADC circuit (manufactured PCB). ....	234
Figure A-4 Proposed ADC circuit (mounted PCB). ....	234
Figure A-5 Proposed CPU circuit (Electrical diagram).....	235
Figure A-6 Proposed CPU circuit (PCB layout).....	235

Figure A-7 Proposed CPU circuit (manufactured PCB).....	236
Figure A-8 Proposed CPU circuit with components mounted.....	237
Figure A-9 Proposed GPS circuit (Electrical diagram).....	238
Figure A-10 Proposed GPS circuit (PCB layout).....	238
Figure A-11 Proposed GPS circuit (manufactured PCB).....	238
Figure A-12 Proposed GPS circuit (Mounted components).....	238
Figure A-13 Proposed GPS circuit (Electrical diagram).....	239
Figure A-14 Proposed GPS circuit (PCB layout).....	239
Figure A-15 Proposed GPS circuit (manufactured PCB).....	239
Figure A-16 Proposed GPS circuit (manufactured PCB).....	239
Figure B-1 Non-Linear ADC responses, adapted from [155].....	244
Figure C-1 Common clocking terminology.....	245
Figure C-2 PLL internal components.....	247
Figure E-1. State Matrix for a “Sample message” encoded as 0x534F4D452031323820424954204B4559	253
Figure E-2 Key state matrix for a “16 B SECRET KEY” .....	253
Figure E-3 Key scheduling algorithm. ....	254
Figure E-4 The S-BOX transformation concept, a form of substitution [70].....	254
Figure E-5 S-BOX transformation, with a nonlinear substitution table, adapted from [70].....	255
Figure E-6 Message ‘XORing’ with a key, the key scheduling algorithm provides different keys for each round. .....	255
Figure E-7 Message transformation, after the round key step.....	256
Figure F-8 ECB mode, adapted from [162], [161].....	259
Figure F-9 ECB mode, adapted from [162], [161].....	260
Figure F-10 CTR mode, adapted from [162], [161].....	261
Figure G-1 Basic XOR based hashing function.....	264
Figure G-2 Forging messages by adding data, due to a not size-dependent hash function .....	266
Figure G-3 Forging messages by exploiting collision resistant weakness, and/or exploiting weak one-way functions.....	267
Figure G-4 General the merkle-damgård construction, used by several hash functions.....	267
Figure G-5 Davies-Meyer ‘h’ compression function.....	268
Figure G-6 Matyas–Meyer–Oseas ‘h’ compression function.....	268
Figure G-7 MD5 single operation general diagram, where $\boxplus$ denotes 232 modular addition, adapted from [172], [171]. ....	269
Figure G-8 SHA-1 single operation general diagram, where $\boxplus$ denotes 232 modular addition, adapted from [177] .....	271
Figure G-9 SHA-256 single operation general diagram, where $\boxplus$ denotes 232 modular addition, adapted from [181] .....	273
Figure G-10 CBC-MAC raw mode of operation, at the TAG step additional functions can be implemented to improve security.....	275
Figure G-11 CMAC mode of operation, showing key usage according to message size.....	276
Figure H-1. The intervening blocks of digital communications.....	277
Figure H-2. The $I - Q$ representation used on digital communications.....	278
Figure H-3. A generic I-Q modulator. ....	278
Figure H-4. The ASK modulation shown in the I-Q plane.....	279
Figure H-5. The ASK time domain modulation for $n = 1$ .....	279
Figure H-6. The BPSK time domain modulation for the data sequence “101”.....	281
Figure H-7. The QPSK modulation expressed on the I-Q constellation [93]. ....	281
Figure H-8. The QPSK modulation on time domain based on the generic I-Q modulator.....	282

Figure H-9. Alternative QPSK modulation based on a rectangular representation. ....	283
Figure H-10. O-QPSK modulation based on a rectangular representation. ....	284
Figure H-11. Sample BPSK correlation demodulator, adapted from [99]. ....	285
Figure H-12. Using a correlation receiver to demodulate a radio signal with value {0b101001}, adapted from [99]. ....	286
Figure H-13. Using a correlation receiver to demodulate a radio signal (Zoomed version), adapted from [99].	286
Figure H-14 The DSSS modulation scheme block diagram.....	288
Figure H-15. Sample DSSS encoding for a BPSK modulated signal.....	288
Figure I-1 The IEEE 802.15.4g frame format [98]. ....	289
Figure I-2. The IEEE 802.15.4g frame processing. ....	290
Figure I-3 Pilot insertion on a standard IEEE 802.15.4g PHY frame [99]. ....	295

## List of Tables

Table 1.1 NIST smart grid proposed domains, adapted from [15].	9
Table 2.1 Decimation filter to obtain an ‘n’ bits output.	29
Table 2.2. Spectral decomposition formulas.	37
Table 2.3 Fundamental energy quantities described on IEEE 1459-2000	39
Table 2.4 Non-fundamental quality indicators described on IEEE 1459-2000	39
Table 2.5 Summary of energy quantities grouped by source, adapted from [61].	40
Table 3.6 IEEE C37.118 standard, class requirements for the stationary operation of PMU	46
Table 3.1 Cyber-Physical attacks scenario for smart grids, adapted from [9]	63
Table 3.2. Error correcting code applied to a PUF extracting function, as described by [68]	69
Table 3.3. Physical microcontroller identifiers for the devices used during the experimental phase of the PUF extraction function development.	70
Table 3.4. Sample bit sequence extracted from a PIC32MZ device using a custom firmware, with temperature variations. Device #0	71
Table 3.5. Sample bit sequence extracted from a PIC32MZ device using a custom firmware, with temperature variations. Device #1	71
Table 3.6. Set bits (0b1) distribution for each sample device	72
Table 3.7. Equidistribution test for the XORed string of the small window sample.	72
Table 3.8. Extracted bit values for device 0.	73
Table 3.9. Extracted bit values for device 1.	74
Table 3.10. Complete bit sequence extracted from device #0 (0x58FAF7E6881EDEAFFAC2A7CF0259941)	74
Table 3.11. Complete bit sequence extracted from device #1 (0xA054CBEF8AE074510716DEDAD41C543)	75
Table 3.12. Extracted IDs random properties, (Equidistribution test)	76
Table 3.13. Hamiltonian bit string equidistribution properties.	78
Table 4.1. Sample dB scale tied to 1mW of power.	82
Table 4.2. Part of the 802.1x family of standards, adapted from [96]	87
Table 4.3. IEEE 802.15.4 Physical Layer characteristics, summary taken from [10], [96] [97].	89
Table 4.4. IEEE 802.15.4g Physical Layer requirements for O-QPSK transceivers operating on 2.4 GHz, adapted from [10].	91
Table 4.5. “Frame type” field bit encoding, taken from [101].	93
Table 4.6. Destination and Source Address fields bit encoding, taken from [101]	94
Table 5.1. Reported metering alterations on the Mexico state municipality of Chimalhuacan, adapted from [110].	112
Table 5.2. Sample output from the simplex method for several rounds.	139
Table 5.3. Computed score from the sample given in Table 5.2.	139
Table 5.4. Common harmonic components found of common household/office environments, adapted from [123].	142
Table 5.5. Reported suspicion factors, for a 20+ users connected to a single phase.	147
Table 5.6. Interpreted suspicion factors, for a 20+ users connected to a single phase, (77.7% effective detection rate)	147
Table 5.7. Reported suspicion factors, for a 15+ users connected to a single phase, and two altered units.	148
Table 5.8. Interpreted suspicion factors, for a 15+ users connected to a single phase, and two altered units (91.6% effective detection).	149
Table 5.9. Reported suspicion factors, for a 15+ users connected to a single phase, and a single altered unit.	149
Table 5.10. Interpreted suspicion factors, for a 15+ users connected to a single phase, and a single altered unit (100% effective detection).	150

Table 5.11. Reported suspicion factors, for a 45 legally connected users plus 15 wire tappers to a single phase. .....	151
Table 6.1. R-Type instruction encoding.....	155
Table 6.2. I-Type instruction encoding.....	156
Table 6.3. J-Type instruction encoding.....	156
Table 6.4 Wait slots required before issuing data dependent instructions, adapted from [129].....	163
Table 6.5 Internal Register Configuration of the ADS131E08 for various ENOB, taken from [134].....	170
Table 6.6 Sallen-Key filter components.....	177
Table 6.7 Sallen-Key filter characteristics.....	177
Table 6.8 Comparison of the number of operations required to transform discrete time signals into frequency domain quantities by using DFT and FFT.....	182
Table 6.9 Comparison of the number of operations required to transform discrete time signals into frequency domain quantities by using $\mathcal{R}$ field operations.....	182
Table 6.10 Comparison of the number of operations required to transform discrete time signals into frequency domain quantities by using $\mathcal{R}$ field operations under a MAC enabled unit.....	183
Table 6.11 Sample values for testing the proposed method, showing the exact values.....	187
Table 6.12 Time and precision comparison of the proposed method against the Babylonian method.....	187
Table 6.13 ADC voltage and current configurations employed by the digital simulation.....	188
Table 6.14 Harmonic contents (magnitude) obtained from the sample voltage signal.....	189
Table 6.15 Harmonic contents (magnitude) obtained from the sample current signal. ....	190
Table 6.16 Execution times obtained by the FFT and DFT algorithms implemented on the PIC32MZ.....	190
Table 6.17 Comparative results of the IEEE sample calculations and those implemented by the author. ....	191
Table 6.18 Harmonic contents (angle) obtained from the sample waveforms given in [61] .....	192
Table 6.19 Angle LUT used to compute the phasor angles .....	192
Table 6.20 Time stamp differences between two GPS units and their associated electrical angle error.....	194
Table 6.21 Signal characteristics obtained from two PMU units that are connected to the same power outlet.....	197
Table 6.22 Total Vector Error computed from the readings obtained in Table 6.21 .....	197
Table 6.23 Constant execution times achieved by the developed algorithm .....	201
Table 6.24 Block encryption execution time for the developed algorithm.....	202
Table 6.25 PLL generated frequency characteristics.....	209
Table 6.26 PLL generated frequency characteristics (with loopback control). ....	210
Table A.1. Used pins in the communications microcontroller .....	240
Table A.2. Used pins in the metering microcontroller. ....	241
Table E.3. Sample operations under the Galois $2^8$ finite field.....	252
Table H.1. Various Barker codes organized by chip length, adapted from [93].....	287
Table I.1. IEEE 802.15.4g PHR field for the physical frame, adapted from [10].....	289
Table I.2. Standard IEEE 802.15.4-2011 PHR field for the physical frame, adapted from [101] .....	290
Table I.3. The IEEE 802.15.4g FEC stream generation pseudo-code, based on the equations described on [10]. .....	291
Table I.4. FEC code generation example for a PSDU field, based on the description given in [10]. ....	291
Table I.5. Generated bit positions for interleaving algorithm considering a 126-bit length PSDU field, ( $\lambda = 7$ ) [10]. .....	293
Table I.6. PSDU field interleaving example, highlighting the original vs new bit positions. ....	293
Table I.7. (8,4) DSSS bit to chip mapping, taken from [10] .....	294
Table I.8. (16,4) DSSS bit to chip mapping, taken from [10].....	295
Table I.9. Pilot sequence insertion characteristics for 2.4GHz carrier band, adapted from [10].....	296
Table I.10. CCA sampling time for various Frequency bands, taken from [10] .....	296

## Lists of Used Abbreviations

ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AES-ECB	Advanced Encryption Standard in Electronic Code Book mode
AM	Amplitude Modulation
AMI	Advanced Meter Infrastructure
AMR	Automated Meter Reading
ASICs	Application Specific Integrated Circuits
ASK	Amplitude Shift Keying
BDE	Bit Differential Encoding
BPSK	Binary Phase Shift Keying
CBC	Cipher Block Chaining
CCA	Clear Channel Assessment
CFE	Comisión Federal de Electricidad
CIM	Common Information Model
CISC	Complex Instruction Set Computer
CMAC	Cipher Based Message Authentication
CPHA	Clock Phase
CPOL	Clock Polarity
CPP	Critical Peak Pricing
CO <sub>2</sub>	Carbon Dioxide
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CRC	Cyclic Redundancy Check
CS-PRNG	Cryptographic Secure Pseudo Random Number Generator
CTR	Counter Mode
DAC	Digital to Analog Converter
DADP	Day Ahead Demand Pricing
DAP	Day Ahead Pricing
DDoS	Distributed Denial of Service
DDR	Develop a Dynamic Demand Response
DG	Distribution Generation
DGM	Distributed Generation Management
DES	Data Encryption Standard
3-DES	Triple Data Encryption Standard
DFT	Discrete Fourier Transform
DH	Diffie-Hellman
DMS	Data Management System
DOE	Department of Energy
DoS	Denial of Service
DSO	Distribution System Operator
DSSS	Direct Sequence Spread Spectrum

ECB	Electronic Code Book
ED	Energy Detection
EDF	ÈlectricitÈ de France
E2SG	Energy to Smart Grid
EPRI	Electrical Power Research Institute
ERT	Encoder Receiver Transmitter
FAPER	Frequency Adaptive Power Energy Rescheduler
$F_c$	Cutoff Frequency
FM	Frequency Modulation
EEMBC	Embedded Microprocessor Benchmark Consortium
FEC	Forward Error Correction
FERC	Federal Energy Regulatory Commission
FHSS	Frequency Hopping Spread Spectrum
$F_s$	Stop band frequency
$F_0$	Fundamental frequency
GIS	Geographical Information System
GSM	Global System for Mobil Communication
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HAN	Home Area Network
HECO	Hawaiian Electric Company
HMAC	Hash Based Message Authentication
HVAC	High Voltage Alternative Current
ICT	Information and Communication Technology
IMS	Industrial Medical and Scientific
IP	Internet Protocol
ISA	Instruction Set Architecture
ISO	International Organization of Standardization
IT	Information Technology
IV	Initialization Vector
JCP&L	Jersey Central Power Light
LAN	Local Area Network
LLC	Logical Link Control
LPF	Low Pass Frequency
LQI	Link Quality Indication
LR-WPAN	Low Rate Personal Area Network
LTE	Long Term Evaluation
LV	Low Voltage
LUT	Look Up Table
MAC	In cryptography: Message Authentication Code In communications: Media Access Control

	In computer architecture: Multiply-Accumulate unit/operation
MD5	Message Digest Algorithm -5
NAN	Neighborhood Area Network
NB-PLC	Narrow Band Power Line Communications
NIST	National Institute of Standards and Technology
NTL	Non-Technical Losses
NUMA	Non Uniform Memory Access
OMS	Outage Management System
O-QPSK	Offset Quadrature Phase Shift Keying
Op-Amp	Operational Amplifier
OSI	Open System Interconnection model
PCB	Printed Circuit Board
P.F	Power factor
PLC	Power Line Communications
PLL	Phase Locked Loop
PM	Phase Modulation
PHR	Physical Layer Header
PHY	Physical Layer
PSDU	Physical Layer Service Data Unit
PSK	Phase Shift Keying
PRNG	Pseudo Random Number Generator
PUF	Physical Unclonable Functions
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RE	Reverse Engineering
RF	Radiofrequency
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
RTP	Real Time Pricing
RSA	Rivest-Shamir-Adelman
RSSI	Receive Signal Strength Indicator
s	Complex frequency (complex plane)
SAIDI	System Average Interruption Duration Index
SAIFI	System Average Interruption Frequency Index
SAR	Successive Approximation Register
SCADA	Supervisory Control and Data Acquisition
SCL	Serial Clock Line
SDA	Serial Data Line
SDG&E	San Diego Gas and Electric
SFD	Start of Frame Delimiter
SGDI	Smart Grid Demonstration Initiative

SHA-1	Security Hashing Algorithm 1
SHA-2	Security Hashing Algorithm 2
SHR	Synchronization Header
SMUD	Sacramento Municipal Utility District
SNR	Signal to Noise Ratio
SoC	System on Chip
SPI	Serial Peripheral Interface
SS	Select Signal
TL	Technical Losses
TCP/IP	Transmission Control Protocol/ Internet Protocol
THD	Total Harmonics Distortion
ToUP	Time of Use Pricing
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver /Transmitter
VCO	Voltage Control Oscillator
VPN	Virtual Private Network
WAN	Wide Area Network
WE	Weekend
WD	Weekday (Monday-Friday)
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
	Other abbreviations
$\lambda$	Wavelength of a signal
$\omega$	Angular frequency
$df$	degrees of freedom

## Publications from the Study

1. David Jonathan Sebastián Cárdenas, David Sebastián Baltazar, Ricardo Mota Palomino, **“Secure Credential Storage in an MCU by Means of a PUF”**, VII Congreso Internacional Ingeniería Electromecánica y de Sistemas, Instituto Politécnico Nacional, SEPI-ESIME, México, D. F., October 27<sup>th</sup>-31<sup>st</sup> 2014.
2. Jonathan Sebastián Cárdenas, David Sebastián Baltazar, Ricardo Mota Palomino **“Diseño e implementación de un medidor de energía eléctrica con comunicación inalámbrica hacia un centro de facturación”**, RVP-AI/2014, Acapulco Gro, July 20<sup>th</sup>-26<sup>th</sup> 2014
3. Jonathan Sebastián Cárdenas, David Sebastián Baltazar, Ricardo Mota Palomino **“ Diseño de una unidad de medición fasorial compatible con la norma IEEE 1459-2000 e IEEE C37.118-2011 para redes de distribución”**., XIV Congreso Nacional de Ingeniería Electromecánica y de Sistemas, Instituto Politécnico Nacional, SEPI-ESIME, México, D. F., november 11<sup>th</sup>-15<sup>th</sup> 2013.
4. Jonathan Sebastián Cárdenas, David Sebastián Baltazar, Ricardo Mota Palomino **“Algoritmo Digital Para La Medición De Energía De Acuerdo A La Norma IEEE 1459-2000 Usando Un Microcontrolador MIPS”**, RVP-AI/2013, Acapulco Gro, July 7<sup>th</sup>-13<sup>th</sup> 2013



## CHAPTER 1

### 1. INTRODUCTION

#### 1.1 Presentation

Over the last century, electricity has changed the way human society interacts, giving all sorts of communicating possibilities; yet electricity itself has remained unchanged. Over the last 100 years it has remained as three different areas that are mutually interdependent, and these areas are generation, transmission, and distribution.

Residential distribution systems are the last step in energy delivery to end users. They are usually long medium voltage (MV) lines attached to distribution transformers, allowing low voltage (LV) clients to receive energy. These large networks have been designed in conservative ways for many years, often being over designed, and until recent years with very little monitoring, making it one of the most inefficient areas of power transmission. In the last century, utility companies have relied on human resources, to gather data around the distribution network, obtaining information, including meter readings, looking for faults, taking Quality of Service (QoS) statistics such as voltage and service continuity at consumer endpoints [1].

Environmental awareness on reducing carbon dioxide (CO<sub>2</sub>) emissions, cost-reducing strategies and an overall tougher regulation has resulted in the modernization of distribution systems. The electric market is transitioning from a centralized, one mass producer-controller network to a more distributed production system. Nowadays it is frequent to see small private generators connected to the grid. With the advances in power generation, homes might one day start selling energy produced by their solar cells or wind generators. All of this has led to think consumers not only as clients but as active market players, which could play important roles in reducing transmission losses, helping generation during peak demand hours, thus giving birth to the term “smart grid” [1].

The Smart Grid refers to an ample set of definitions; however, this work focuses on energy metering on distribution networks, known as Smart Metering Networks (SMUN). SMUN refers to systems that measure, collect, and analyze data, using Information and Communication Technology (ICT). One of the first steps taken towards this goal were the introduction of digital meters known as Automatic Meter Reading (AMR) devices [2], which allowed handheld devices to read energy consumption, and

later on process this data on the utilities servers to bill a user. Currently improved versions of these meters are being developed and put in mass scale service. These improved versions are called the Smart Meter [2]; they feature bidirectional communication systems, bringing the ability of monitoring the load on real time, and under certain conditions managing it, allowing new metering applications to emerge.

One field of interest concerning smart grid networks is lowering energy losses. There are two types of energy losses, technical and non-technical losses. Technical losses are an inherent problem of power distribution due to conductor resistance and the fact that energy consumption is generally far from generation plants. Non-technical losses (NTL) refers to energy losses caused due to theft or meter malfunctions resulting in loss of revenue [3].

Energy theft can be done in several ways, such as meter tampering, meter bypassing, or illegal connections to the distribution networks, and even altering energy fees by the utility employees. This work pretends to identify meter tampering or meter bypassing and to quantify energy theft due to illegal connections by using a central observer unit similar to one described in [3], using real time waveform data to solve a linear set of system equations, by employing the Kirchhoff current law.

## 1.2 Motivation

The newly approved Mexican Energy policy (ratified as of 2014) [4] has allowed open market conditions for the electrical sector. These policies enable energy generation on any scale, as well as transmission and distribution networks outsourcing, leaving the former electric utility Comisión Federal De Electricidad (CFE) with the task of operating the electrical network. Under this new scheme, the government owned CFE will make profits based in the amount of billed energy, and therefore must raise its overall billing indexes by means of reducing energy losses.

### 1.2.1 Energy Losses

In the electricity market, energy losses refer to the amounts of electricity generated, versus the amount of energy effectively billed to end consumers. As mentioned earlier, energy losses can be further divided into Technical Losses (TL) and Non-Technical Losses (NTL).

Technical losses are an inherent nature of energy transportation, for example, resistance in transmission lines, transformer efficiency, and even equipment operating temperatures. These losses are to be expected in all energy transportation companies; they can be minimized, can be planned, but never eliminated.

Non-technical losses are ideally due to malfunction metering equipment, high impedance ground faults, but in reality also account for energy theft. Energy theft refers to the net difference between the energy effectively supplied by the distribution utility and the effectively billed energy of the consumers [3]. High scale energy theft is mostly present on developing countries, but is also present in US, and UK markets, and it could be responsible for up to 30% of home fires that originate near the meter perimeter [5].

In Mexico, for example, people steal electricity with wires known as “*diablitos*”, or “little devils”, this stealing method can vary in complexity, ranging from direct tapping of distribution wires, hidden cables behind the meter, to sophisticated electronic manipulation of electronic meters [6]. Most energy theft is concentrated in low-income neighborhoods, or in areas of illegal street vendors, which usually hook to the nearest pole.

Mexican utility service CFE is considered on this study due to its national presence. CFE is a state owned electrical utility, responsible for generation, transmission and distribution of energy; it is the only authorized entity for selling energy to end users. CFE has estimated energy losses as high as 30% at the distribution level.

### **1.2.2 *Secure Data Communications***

Smart grid networks components are designed to transmit vast amounts of data to the utility data centers in order to provide near real time monitoring and control capabilities. Specifically smart meters allow detailed recording of power usage and remote management. Unauthorized use of these recordings could pose a privacy risk for the consumer [7]. Furthermore, due to the network architecture of smart grids and the use of public known protocols as their backbone infrastructure, meter networks can be attacked. Smart meters can be attacked in analogous manners to computer

networks; attacks can range from Denial-of-Service (DoS), meter recalibration to remote client disconnections [8]. AMI security is a major concern for the National Institute of Standards and Technology (NIST), which has suggested the establishment of a secure network protocols that allow certain level of trust. These security measures should be robust against information interception, information tampering, and unauthorized access [9].

### 1.3 Objective

The main objective of this work is to design and implement a smart meter capable of network communications, capable of assessing non-technical losses in low voltage distribution systems.

### 1.4 Secondary Objectives

- Create a smart meter design capable of complying with energy metering standards, equipped with wireless communication.
- Develop a communication network based on standardized communications protocols that allow fast inter-meter communication.
- Create a methodology designed for real time energy theft detection based on a central energy observer.
- Develop a web-based interface for utility personnel management with end-user oriented monitoring capabilities.

### 1.5 Thesis Scope and Limitations

As mentioned in the prior section two main goals are pursued in this thesis. The first one focuses on implementing a smart metering unit that meets standards related to measurement specifications and provides network communications based on standardized protocols. The proposed hardware should replicate some of the characteristics associated with commercial units including functional features, electrical design elements and overall physical design.

Although commercial implementations offer advanced system integrations with the utility data servers like customer database integration, automatic outage detection and network analysis tools these types of features were not implemented and only a proof of concept were provided where

applicable. Similarly, due to time limitations the produced prototype was only tested with respect hardware/software functionalities and no reliability tests were produced.

The second goal is to describe a methodology for detecting energy theft associated with individual customers, the proposed methodology was restricted to the capacities provided by the designed metering unit. The methodology tests were limited to computer simulations that represent the consumption patterns of customers in Mexico with typical energy losses.

## 1.6 Contributions

The main aim of this study was to develop a methodology capable of identifying individuals responsible for energy theft, as well as the quantification of total energy losses in low voltage (LV) distribution systems, based on the smart grid concept of network communications. The proposed methodology uses the characteristics provided by a smart metering unit that was co-designed to satisfy the methodology requisites.

The proposed smart meter complies with revenue grade metering and communication standards typically found on commercial smart meters. This meter implements security measures to prevent tampering or data modification intended to alter the amount of recorded energy. The designed meter meets specific prerequisites that allow it to execute the proposed methodology based on limited standards compliance like IEEE C37.118.

The developed smart meter uses a dual microcontroller architecture; each microcontroller is a MIPS based architecture, known as the MICROCHIP™ **PIC32MZ2048ECM**, paired with a TI™ **ADS131E06** a 24-bit delta-sigma ADC unit; it features a dual MICROCHIP™ **MRF24XA** IEEE 802.15.4 radio compliant module, capable of 2-Mbps wireless communications, plus a GPS unit, and Ethernet port. Several auxiliary devices were used including a programmable Voltage Controlled Oscillator and other Application Specific Integrated Circuits that are described in Chapter 6<sup>th</sup>.

At the time of writing IEEE published the standard IEEE 802.15.4g, allowing higher data throughput for smart metering utility networks (SMUN) [10], yet no hardware vendors were found to be fully

compliant during initial phases of this thesis, so a modification of a partially compliant IEEE 802.15.4g transceiver is used to enable data communications. The selected transceiver allows low-level Media Access Control (MAC) functions at raw data levels, allowing custom modifications of the protocol stack through software handling.

Finally, the results presented in this thesis are subject to a patenting process under an innovation project sponsored by the IPN under Grant No *SIP-20144682*.

## 1.7 Thesis Outline

The thesis is arranged as follows:

- Chapter 1. Provides motivation about this work and a general background of smart grids, introducing the reader into the thesis subject. A literature review of previous smart grid implementations and proposals involving smart metering on distribution area deployments is given.
- Chapter 2. Focuses on energy metering algorithms and measuring standards required for the development of an energy measurement compliant meter. Two IEEE standards involving electrical signal measurement are discussed.
- Chapter 3. Focuses on the security of communications by using cryptographic primitives, it gives details on the implementation of secure systems and its associated vulnerability, a Physical Unclonable Function is proposed as a key storage mechanism.
- Chapter 4. Gives an in depth introduction of wireless networks, providing details on the implementation, beginning with modulation schemes, and ending on Secure Socket Connections. At the end of the section, the designed communication interface is discussed.
- Chapter 5. Lists common lost energy causes and their classical solutions, it provides specific meter fraudulent practices and possible solution methods by the use of electronic meters. At the end of this chapter, a central observer based energy theft detection algorithm is proposed.
- Chapter 6. Follows the development of a measuring apparatus and its network implementation, aiming at real time energy theft detection. At the end of the chapter the

standards compliance is evaluated, and the pilot experiment results for energy theft detection are reported.

- Chapter 7. Provides conclusion of this thesis and future work ideas.

## 1.8 State of the art

Due to the dual intended scope covered by this work, the state of the art has been split into two sections. The first section encompasses the smart grid projects tailored at the consumer level (smart meters, load management devices), while the second part introduces the reader to energy theft detection techniques.

### 1.8.1 *Smart grids*

Energy transmission has existed practically since the beginning of modern electricity use. Its need for a reliable operation and complexity has required the adoption of the computer technology almost since its early days. In the latter half of the XX century, computer programs were developed to analyze the power system, helping engineers to plan according to the expected demand growth. Computer tools with the help of dedicated hardware gave birth to *supervisory control and data acquisition systems* (SCADA) in the early 1980's. SCADA systems were added to generation plants and transmission systems bringing real-time monitoring capabilities, helping transmission networks and generation units to become a very reliable and predictable system [11].

On the contrary, the distribution system is a very extensive network with very little monitoring (most real time monitoring occurs at substations). Its characteristics (long lines, radial network and low voltage buses) make it one of the most inefficient areas of power transmission; distribution networks were left aside from technological developments for several decades mainly due to their complexity [12]. In the last century, utility companies have relied on human resources, to gather data around the distribution network, including obtaining meter readings, looking for faults, taking Quality of Service (QoS) statistics such as voltage and service continuity on consumer endpoints [1].

The electric grid is the collection of lines, substations, and transformers that deliver electricity from the generation plants to end users (consumers). The “grid”, as it is known, was built in the late XIX century; it was slowly modernized in situ as new equipment became available. This grid from its

inception was designed only to deliver energy to consumers, but new technologies and modern legislation have allowed users to cogenerate energy, changing the traditional energy flow direction. The Smart Grid term was coined in order to describe expected grid technologies; these expected capabilities take into account the digital age, making extensive use of communications [1].

This new “grid” is required to automate and manage the increasing complexity of electrical networks by using “digital networks”. In other words, the smart grid uses digital technology, to allow two-way communication between the utility and its customers; the new grid involves the use of controls, computers, automation schemes, and new technologies that work together in order to improve energy use [1].

### 1.8.2 *Smart grid objectives*

Smart grids are being tested all over the world, often sponsored by regional groups. In North America, the Electric Power Research Institute (EPRI) has cosponsored a series of smart grid projects across the US and in some overseas countries. While the Energy to Smart Grid (E2SG) has been in charge of European countries. Most EPRI and E2SG projects are multiyear initiatives targeted at finding possible uses of smart grids, as well as evaluating proposed technologies in real market conditions, these projects seek to create a repeatable smart grid architecture [13].

According to EPRI smart grid resource center, a smart grid “*is one that incorporates information and communications technology into every aspect of electricity generation, delivery and consumption in order to minimize environmental impact, enhance markets, improve reliability and service, and reduce costs and improve efficiency*” [14]

It is important to note that the term *smart meter* is mistakenly used to describe the *smart grid*, the *smart grid* encompasses a much wider definition including generation, transmission and distribution into the picture, whereas the smart meter only includes a part of energy distribution. To prevent further misconceptions the U.S. National Institute of Standards and Technology (NIST) proposes that the term smart grid must be encompassed in a domain in order to be described [15].

NIST has suggested the use of seven domains; each domain allows actors (technologies, devices, systems, e.g. smart meters) to be specifically developed or tailored for the applications (automation, renewable energy generation, energy management) required in the domain. In Table 1.1 the NIST proposed domains are further described. NIST has also set a roadmap for standardization of the smart meter market by the creation of guidelines for a total of 97 areas; including communications, risks assessment, security handling, systems architecture, and testing for conformity of devices.

Table 1.1 NIST smart grid proposed domains, adapted from [15].

Domain	Description
Customers	Electrical energy end users, they might generate, store, and manage the use of energy.
Markets	Energy operators and participants in electricity markets.
Service Providers	Organizations providing services to electrical customers and utilities.
Operations	Organizations in charge of energy flow.
Bulk Generation	Generation units, they might store energy
Transmission	Bulk Energy carriers, that might operate generation units
Distribution	Final step of energy distribution to end consumers

According to the previous NIST designation, smart meters can be included in the customer and distribution domain. These domains seek to increase energy efficiency, reduce costs and improve power quality issues. Some others the fields requiring improvement are related to financial, environmental, system reliability, and customer service [16]. Listed below are some of the areas that could be improved by the use of smart grid technology on distribution systems:

**Financial:** On traditional distribution systems, most networks are over designed. With the help of smart grids, better load knowledge can be achieved, helping to install properly sized infrastructure, predict expansion, and optimize load flows, thus reducing operational costs. Furthermore, equipment can be regularly monitored, ensuring maintenance forecasting based on real world data, keeping out of service equipment to a minimum and reducing equipment-repairing costs.

**Reliability:** It is common for utilities (at the distribution level) to know about power failures only after a customer calls in. Real time measuring devices are typically located at substations, and

some sparse locations throughout the feeders. With the help of smart meters, the affected meters can send alarm messages, helping to pinpoint the fault.

**Quality:** Some distribution feeders suffer from voltage issues; fluctuating voltages can damage certain electronic devices, which adversely affect the operations of industries. Real time monitoring equipment and on-site generation can mitigate voltage issues, raising energy quality.

**Environmental:** The integration of environmentally friendly generation (wind and solar) at the consumer premises can reduce the CO<sub>2</sub> footprint of traditional energy generation, while providing economic benefits to the consumer.

### 1.8.3 *Smart grid projects*

An important aspect of current smart grid deployments is that they are often sponsored by third party organizations, although there are some utilities that are deploying units by their own initiative. In the next section, some representative smart grid projects involving smart metering are discussed.

#### 1.8.3.1 *Smart grid –EPRI sponsored Projects*

In 2008 EPRI launched a series of demonstrative projects that seek to design, deploy, and evaluate distributed energy resources in real life scenarios; at the time of writing of this document 24 utilities had participated [17]. Developed projects cover a wide range of topics, from generation to distribution; some projects also involve other energy sources such as coal and gas. EPRI projects pursue to create shared smart grid architecture, which can be reused [13]. The title “Smart Grid Demonstration Initiative” (SGDI) was selected as the all-encompassing project. A selection of interesting projects are listed below. Further information can be found at [14].

##### *I. Pre-Smart Meters*

Before smart meters were conceived, some utilities invested in automatic metering devices, which relied on radio transmitters (ERT radios) to transmit metering data to utility vehicles passing nearby. Early AMR meters were introduced to allow time of day tariffs, encouraging industrials to plan production according to market prices, and thus can be thought as early load control devices.

## II. *SmartSacramento™ Project*

Sacramento Municipal Utility District (SMUD) project involves system-wide deployment of smart metering systems, advanced asset monitoring tools, and end user monitoring tools. The goal is to make customers aware of energy consumption, improve management, reliability and efficiency of the grid. The technologies listed below are being evaluated.

**Communication Infrastructure:** Installation of wireless systems that allow two-way communication between field equipment and data management systems.

**Advanced Metering Infrastructure (AMI):** Massive installation of smart meters with improved accuracy and two-way communications provide enhanced load monitoring, outage management and theft detection.

**Time of Day Tariffs:** Customers are able to choose energy tariffs, between traditional, ToUP, CPP, or a combination of ToUP and CPP, helping SMUD assess the effectiveness of load management.

**Dynamic Demand Load Management:** Installation of load management devices, allowing the utility to shut them off according to the demand. These devices are installed on non-critical loads and are offered to customers as means to access financial incentives.

**Distribution automation systems:** Automated control of distribution equipment (Voltage regulators, switching devices, capacitor devices) acting upon system disturbances using constant feeder monitoring.

**Plug-In vehicles:** Charging stations where installed in order to assess the impact of battery charging in the distribution network.

## III. *PREMIO Project*

*Électricité de France* (EDF) is a participating member of the SGDI group; it aims to optimize distributed energy resources in southern France. The project has been named PREMIO a French acronym for “Integration and Optimization of Distributed Generation, Demand Side Management, and Renewable Energies”. The project was launched in late 2010 to fulfill the following goals:

**Develop a Dynamic Demand Response (DDR):** Peak power consumption puts an additional stress on system stability, increasing the risks of wide area blackouts, controlled load shedding (disconnection of enabled interruptible loads) and use of led dimmable public lighting can be used as backup method [18]. Moreover, demand pricing can discourage users from using excessive energy.

**Promote energy efficiency:** Energy saving awareness campaigns have been launched, promoting efficient energy use; customers are offered tips and advice on ways to reduce energy bills and protect the environment.

**Integrate DG and renewable energies:** Policies and financial incentives encourage customers to install renewable energy sources [19], not only providing energy savings for customers, but helping to cogenerate during peak hours, reducing peak power consumption.

**Reduce greenhouse gas emissions from power plants:** By using previously installed renewable sources located at local consumption points, the use of fossil fuel plants can be reduced, protecting the environment.

#### *IV. FirstEnergy Project*

Jersey Central Power & Light (JCP&L) has launched a multi-target smart grid demonstration project in New Jersey, Ohio, and Pennsylvania areas; as other projects it seeks to assess user response to time variant rates. Some of the technologies being evaluated are listed below.

**Communication Infrastructure:** Installation of fiber optic communication and radio frequency mesh networks with pole mounted data concentrators.

**Advanced Metering Infrastructure (AMI):** Installation of smart meters with enhanced load monitoring, improving load forecasting and faster outage management.

**Time-based Rate programs:** Customers are offered energy tariffs a day ahead with CPP, allowing them to plan their energy consumption by using in-home displays that control their devices.

**Dynamic Demand Load Management:** Installation of load management devices, High Voltage AC (HVAC) devices are controlled by the utility, modifying their use according to the demand; the consumers are offered financial incentives according to the extent HVAC systems are allowed to be controlled.

**Integrated Control Platform Visualization:** Development of a distribution system visualization tool providing near real-time information of field equipment. This computer tool also provides historical information for planning purposes, and plotting of detailed trend data available for each field device. Utility personnel can take the necessary steps in order to keep voltage regulation within limits, phase balancing and maintenance planning.

### 1.8.3.2 Smart grid – E2SG sponsored Projects

European ENIAC Joint Undertaking is a public-private partnership focusing on developing information and communication technologies; it has sponsored the Energy to Smart Grid (E2SG) project that aims at developing and demonstrating key technologies for smart grids [20].

#### I. Energy to Smart Grid (E2SG)

Energy to Smart Grid is a multinational project (Austria, Belgium, Germany, Italy, Netherlands, Portugal, Slovakia, Spain and United Kingdom) [21] intended to address the five main areas enlisted below.

**Node-grid interfaces:** Improvement of AC/DC link converters and measurement devices by using modern electronic devices, and establishment of standard practices.

**Grid-sensing/metering:** Integration of monitoring equipment into the grid.

**Over-the-grid communication:** Develop mediums to carry measurements and control information between entities.

**Grid-topology and control:** Use of local energy sources, management of local production/consumption and energy storage.

**Energy routing:** Develop mechanisms for energy transmission (AC/DC) according to operating conditions.

E2SG is supported by a number of electrical industries, which goal is to devise and design mechanisms and policies to assemble, monitor and control smart grids [22]. There are other focus areas, such as reducing CO<sub>2</sub> emissions, through energy efficiency improvement programs, and the development of microgrids that can operate in island modes during critical periods. These gives the consumer the ability to implement energy saving techniques, improving overall customer satisfaction. In Figure 1-1 the E2SG common vision diagram can be observed. This vision integrates all areas that are being sponsored for development.

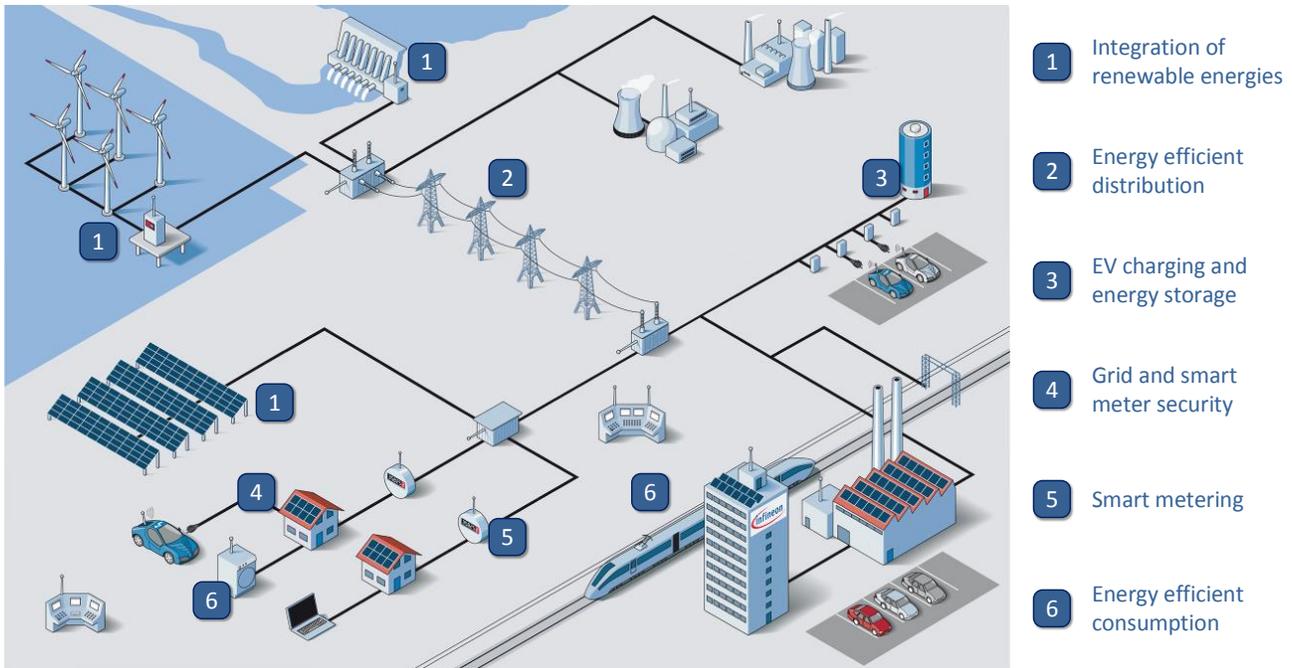


Figure 1-1 E2SG smart grid vision projects, adapted from [20]

### 1.8.3.3 Smart grid – Non sponsored Projects

#### I. Smart grid project in Italy

Italy's bidirectional AMR pilot test gave birth in 2001 to the *Telegestore* project, a pre-smart grid technology showcase-system. *Telegestore* Metering Infrastructure is based on PLC enabled areas that communicate with a local data aggregator. This local aggregator is equipped with *Global System for Mobile Communications* and *General Packet Radio Services* (GSM/GPRS) that allows the utility to send and receive data. Some of the main features provided by the *Telegestore* project are enlisted below.

- Remote energy consumption reading
- Upgradeable billing tariffs
- Remote contract modification of power demand
- Remote connection/ disconnection of power supply
- Energy theft/tamper detect alarm transmission

#### II. Smart grid projects in Mexico

Mexico State owned utility has deployed smart grid technology on an experimental basis, some of these projects are further described in the next paragraphs.

### **Queretaro smart meter deployment**

In late 2013 a set of smart meters developed by the “*Instituto de Investigaciones Eléctricas*” (IIE) was laid on the city of Queretaro. A total of 600 units were installed to analyze the network reliability and overall product performance, these meters feature the following characteristics [23] [24].

- ZigBee enabled wireless communications.
- Energy theft/tamper detect alarm transmission ( By registering reverse current flow)
- Remote energy consumption reading
- Power quality readings (voltage, current and power)
- IEC-61850 compliant communications

### **Cozumel Inteligente project**

Starting in 2012 a series of AMR units were deployed in the city of Cozumel as part of the “*Cozumel Inteligente*” project. These units were designed to operate in a pre-pay basis as a way to reduce client debt and to deter energy theft. On top of this, a set of distribution automation schemes were laid that operate under wireless communications, the main objectives of this automation schemes are provide [25]:

- Real time load balancing
- Fault detection with automatic service isolation and service restoration
- Automatic voltage var control

### **Smart grid project in Mexico City**

The Polanco AMI project is a showcase system featuring a network of smart meter units and distribution monitoring tools. Its main target is to improve energy delivery metrics, including service restoration, power quality and energy theft detection.

The project uses off the shelf solutions acquired from Elster and ENERI, plus custom data bridges that link the utilities data servers with the recorded measurements (see Figure 1-2). CFE has installed 29660 meters as of 2013 [26], but plans to install up to 60000 units in the near future.

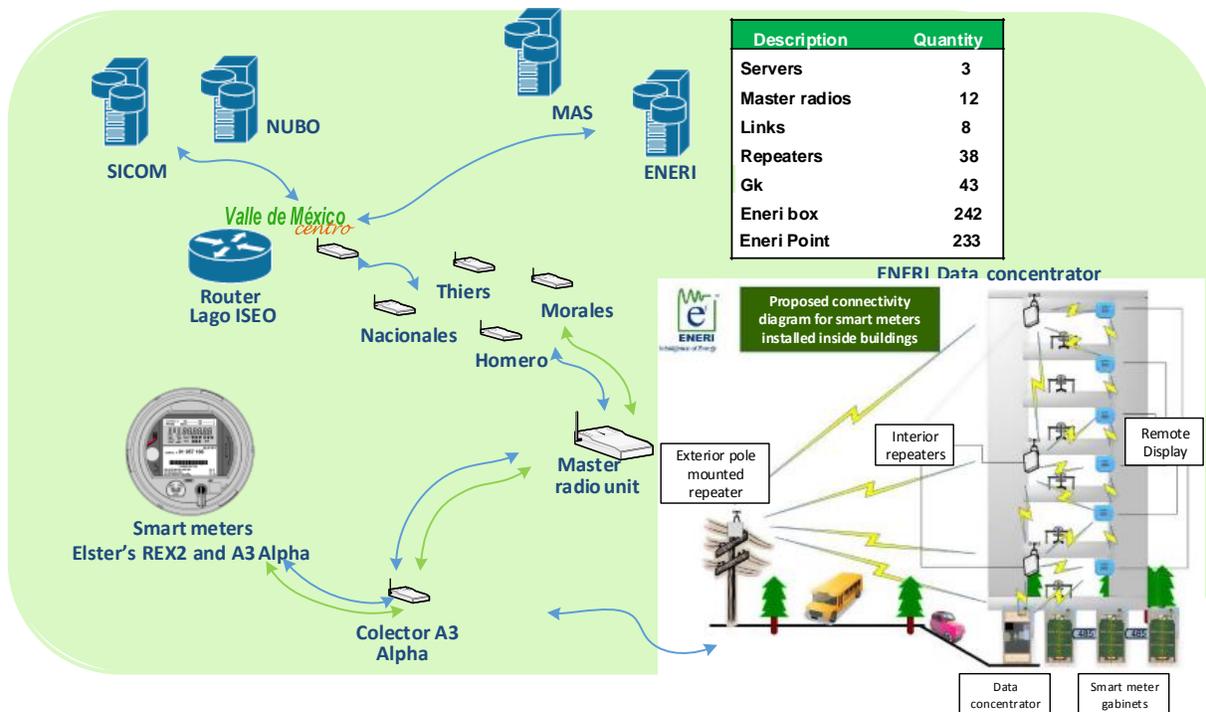


Figure 1-2 Smart meter architecture deployed by CFE in Mexico City, adapted from [26].

The deployed meters feature the following characteristics

- ZigBee enabled wireless communications.
- Energy theft/tamper detect alarm transmission ( By registering reverse current flow)
- Power quality readings (voltage, current and power)
- Remote client disconnection/reconnection
- Energy portal where users can visualize power consumption

The associated distribution network modernization has also allowed the utility to perform the following actions.

- Removal of previously installed “diablitos”
- Client mapping
- Distribution automation
- Per zone energy loss detection based on power balance, similar to the one reported by [27]

Future goals for the smart grid deployed in Polanco

- Dynamic load control
- Smart building control
- Client interaction

- Bidirectional energy metering (PV arrays, wind)

This project has partially been replicated in Mexicali as of 2014 [28].

### *III. Smart metering projects in the university campus*

A smart metering unit requires a wide set of technologies to perform its metering and communication purposes, in the following paragraphs a summary of related works done in the university premises are described.

- In [29] the author details the hardware implementation of a watt-hour-meter capable of obtaining core parameters, such as fundamental frequency, RMS voltage and current as well as Total Harmonic Distortion (THD); these variables are obtained through a time domain analysis by means of the Discrete Fourier Transform (DFT).
- A meter with synchrophasor capabilities is given in [30]. It employs a GPS unit to perform time-synchronized measurements according to IEEE C37.118 Standard, reporting the measured data through a serial port. It performs all the time domain computations based on the DFT, it is a ground breaking work on PC-Microcontroller interfaces (at the EE program), with the introduction of GUI.
- Lastly, in [31] the author develops a smart meter with wireless communications based on the ZigBee protocol and a proprietary monitoring network developed by Digi™ International Inc. This meter also features a GPS synchronization platform but it does not report its measurement capabilities.

#### **1.8.4 Prior Works on Energy Theft Detection**

##### *1.8.4.1 Introduction*

Energy theft has possibly existed since the beginning of electricity metering in the early XX century, with mechanisms designed to avoid it appearing in the early 1950, usually by means of security seals. With the wide adoption of the computer, and use of billing servers, some energy thieves were detected based on energy consumption patterns changes. Later on, state estimation was used to identify areas of great losses based on readings from equipment installed along the feeding circuits.

However, strictly speaking most energy theft was identified by means of inspection teams sent to the field, this has slowly started to change with the appearance of tamper-aware devices, which are discussed in the next sections.

Modern meters, such as AMR meters, provide visual alarms to field personnel upon routine visits, these alarms, depending on the manufacturer, can detect tamper related operations, such as meter removal, meter tilting, or abnormal signal patterns. Newer smart meters designs perform a similar set of features, reporting the tamper events to the utilities management systems by automated communication means. There also have been proposed methods for the identification of energy theft, mostly based on consumption patterns [3], the proposed methods are discussed in detail in the next sections.

#### *1.8.4.2 AMR based theft detection*

The first AMR device appeared in the early 1980; it was a modernization of the electromechanical meter into a digital meter, some with rudimentary display systems. The first AMR meters were deployed to industrial consumers; they enabled power consumption recording according to the time of day (ToD), allowing time variant tariffs, these meters, were read manually by field personnel. At the same time, early tampering techniques appeared some of the techniques involved opening the meter in order to modify the time keeping circuits, altering the displays, or simply bypassing the meter. These tampering techniques were detected by field personnel, and were addressed in newer models; tamper detection methods/devices listed below are commonly found in AMR meters.

**Accelerometer:** Detects a change in orientation in the meter, as well as fast movements. Typical maneuvers while installing bypass cables, or trying to remove the meter from its base, it records an alarm signal when the event happens.

**Proximity sensor:** Detects changes in position, measures the proximity of the meter to its receptacle [32] (base and supporting structure), it records an alarm signal when the event happens.

**Reverse Current Flow:** Detects a change in power flow between the incoming power and load in the terminal block. If one phase is reversed in the 3 phase service, the meter runs in a forward direction whenever three phase motor or three phase load is switched on, but records only 1/3rd consumption [33].

**Low consumption current:** Low power consumption, as well as zero consumption levels for prolonged periods of time, could indicate a faulty unit, or altered wiring (tampering).

**Disconnection event:** A disconnection event while voltage signals are present, indicates a meter removal incident, since meters are usually secured to their bases, it could indicate tamper related maneuvers.

**Meter Case:** A set of micro switches is installed in the inner case, capable of detecting internal tamper attempts; a backup battery is responsible for monitoring these kinds of events when the meter is disconnected.

#### *1.8.4.3 Theft detection in Smart Meters*

Smart meters appeared as an improvement over AMR devices, making extensive use of communications networks, this enables them to send and receive data, most existing tamper detection methods were carried over into smart meters from AMR devices, with some models introducing enhanced alarm signals that are transmitted to the utilities data centers [34].

“Tamper enabled” warning signs on meters deter most users, but fail when the user knows the inner workings of such devices, or the utility personnel modifies meter settings in an act of venality. In order to prevent settings overrides by inner personnel, AMI devices often have two level authentication, one for reading, and one for writing configuration tables, they also contain access logs that keep traces of the intrusion, discouraging most field personnel from attempting it. Nevertheless computer programs specially designed to expose flaws in smart meters, have been exposed on the internet [35], allowing determined users to access the meter, these computer tools with the help of hardware, and electronics knowledge, can render tamper detection technology useless.

Most smart meters comply with IEEE C12.19 standard; this standard allows reading of stored values through an optical port; the port is known as ANSI Type-2 optical connector; this port is designed for in field configuration, and vendor-specific capabilities, but wide area smart meter deployments, have exposed security breaches in the meter optical port. Due to the aforementioned vulnerability, new energy theft detection techniques have evolved, some of these techniques involve the use of neighborhood meters, or a master meter that audits a single meter. The following sections describe proposed techniques found on literature.

#### 1.8.4.4 Theft detection based on power balance

The power balance methods traditionally relied on comparing the amount of supplied energy vs the amount of billed energy, at the feeder level on a monthly basis; with the introduction of smart meters, these basic techniques have progressed from evaluations done monthly to daily reports.

With the introduction of smart metering technologies some granularity has been achieved, on [27] the authors describe a methodology in which nearby measurements are grouped into a single load, and for each of this loads a non-technical losses estimation is given, the authors goal is to identify the critical areas where most of energy theft is done. The technique is based on the power balance diagram that is shown in Figure 1-3, authors assume that no connectivity diagram is available and as such estimate the amount of technical losses for each user (which is dependent on the distance to the transformer). With the technical losses estimate, plus the actual recorded consumptions on the end user devices an overall balance calculation can done, those concentration points with a significant balance variation are said to be possible energy theft points.

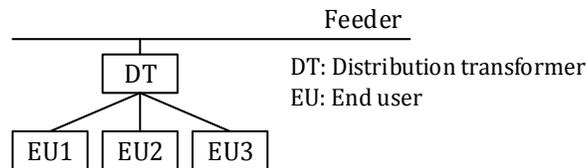


Figure 1-3. The power balance diagram used by the authors on [27]

Another published article, which was ahead of his time, is given on [3], where the authors propose a central observer meter located on the distribution transformer to identify problematic meters, which could be malfunctioning or could be bypassed. For this algorithm to work, each of the customers load profile is compared with a missing energy result by means of a linear set the equations, the costly matrix inversion procedure is reduced to statistical methods, by means of least linear squares. Although the solution is able to pinpoint altered meters it requires that a linear independence exists on the matrix model.

In [36] the authors propose the use of a Support Vector Machine (SVM) to evaluate the presence of altered meters, a SVM is a binary classification algorithm that is used to assign an object to a particular class, the classification is done accordingly to decision boundary (“hyperplane”), which

is adjusted according to training sets [37]. The Vector term in SVM's refers to the fact that the decision is based on the distance between the class boundary and the calculated object value, a good introduction of SVM's can be found on [38]. Continuing with [36] the authors propose an energy theft classifier that is trained with several consumption patterns that mimic the effect of several forms of energy theft and trustworthy consumers, in total 135 classification patterns are proposed. Once the SVM has been trained, a particular customer demand profile is fed into the classifier, if the customer profile is flagged as suspicious then a series of additional test are done to eliminate false positives, in this case positive cases are queue for physical inspection. Although the results are promising, the main drawback is the necessity of configuring input parameters that depend on many customer-specific factors.

Finally, in [39] the authors propose a privacy-preserving energy-theft detection tool based on the solution of a linear system of equations, which are also based on the work done in [3]. However, the solution of the equations is done in a distributed environment, which each metering device determining its honesty coefficients, the solution of the matrix inversion is done according to an LU decomposition, which improves the stability of the system. The overall results are promising and this thesis uses this work as the basis of the proposed algorithm.

## **1.9 Overview of Technologies Required For the Smart Grid Metering**

Two-way communication technologies allow utilities to obtain near real-time readings into control centers from field devices, but require a common infrastructure that guaranties interoperability, reliability and future proof technologies (utilities expect a 5-15 life span for smart meters [40]). Regarding future proof technology, field devices must be upgradable to support new standards or functionalities expected for a future point in time [41].

### **1.9.1 *Advanced Meter Infrastructure (AMI)***

An important piece of monitoring equipment for end-user consumption are advanced meters and their supporting infrastructure (AMI). According to the Federal Energy Regulatory Commission (US agency) report "Assessment of Demand Response and Advanced Metering", the AMI term can be defined as "An Advanced metering system that records customer consumption [and possibly other

parameters] hourly or more frequently and that provides daily or more frequent transmission of measurements over a communication network to a central office” [42].

### **1.9.2 *Communication standards***

Currently deployed meters are mostly based on proprietary communication technology, ranging from ZIGBEE™, Wi-Fi™, WI-MAX™, Encoder receiver transmitter (ERT), cellular networks (LTE, GSM), power line communications (PLC), and fiber optics [43]. Data stacks (data encoding) have even a wider range of variability (Many vendors implement their own software stack). It is likely that proprietary communications schemes will continue to dominate the market in the near future [44].

Due to the potential risk associated with proprietary protocols, governments and standardization offices have begun to steer the market into a more standardized network. In the US, NIST has targeted an “open, interoperable network based on the internet protocol” [15].

### **1.9.3 *Information security***

Once the data interchange mechanisms have been established, the AMI data flow must be protected from attacks that can range from minor single user consequences to system wide inoperability. To prevent this attacks must be prevented by using standardized security suites that enable the grid to resist deliberate attacks, an in depth discussion of smart meter security will be discussed on chapter 3.

## **1.10 Smart Meter Components**

Smart meters can be thought as energy metering devices that incorporate network communication hardware that allows them to measure, record and transmit energy consumption to a central data server. These processes are often split across several hardware/software modules that are illustrated by Figure 1-4 in a simplified manner.

In Figure 1-5 an elaborated version of a typical smart meter architecture is shown. This block diagram contains most of the required components needed to build a functional unit, most of these components will be discussed during this thesis.

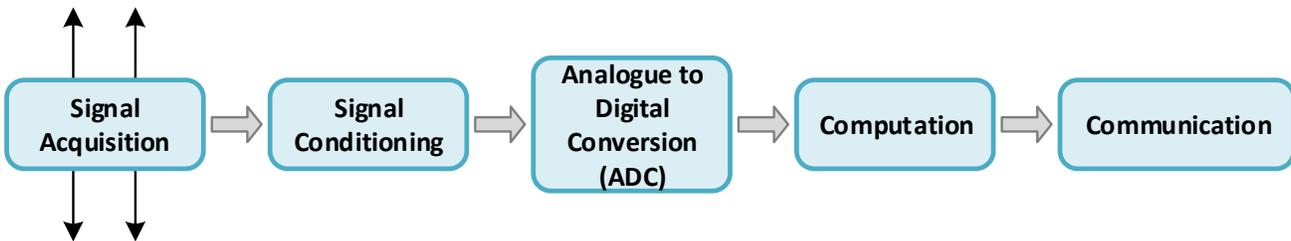


Figure 1-4 Functional block of a smart meter, adapted from [2]

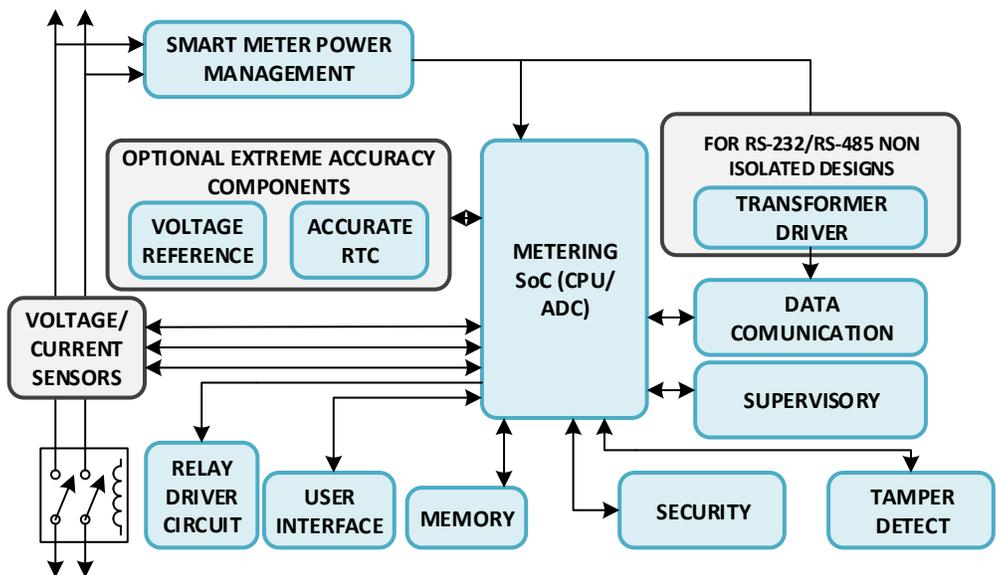


Figure 1-5 Basic Elements of a smart meter, adapted from [45]



## CHAPTER 2

### 2. SIGNAL ACQUISITION AND ENERGY METERING FOR SMART METERS

#### 2.1 Analog to Digital Conversion

Real world signals represent continuous physical quantities, ideally requiring an infinite number of states to accurately characterize their amplitude and unlimited bandwidth to reproduce their rate of change. This contrasts with digital systems that operate in a finite number of states, at discrete time intervals. For a computer-based digital system, only two states can be defined. These finite states (called binary) can be grouped together to symbolize different data patterns, such as characters, numbers or signal values.

Analog to digital converters are used to constrain a continuous set of values into a discrete set of numbers, also known as quantization. These devices have discrete time sampling properties that limit their bandwidth use, as well as resolution limits (quantization levels) that impede full signal reconstruction. Analog to Digital Converters (ADC) are implemented in a number of architectures, often dictating specific parameters, but most implementations share a set properties that describe their signal quantization characteristics. These properties are mentioned on annex B of this thesis.

#### 2.2 Types of ADC

There are several ADC architectures that have been developed over the years, some of the most common architectures are described in the next sections.

##### 2.2.1 Ramp ADC

The ramp ADC works by comparing an input signal to an internally generated reference signal, the reference signal is obtained by feeding a Digital to Analog Converter (DAC) with a binary value. This binary value is generated through an “n-bit” counter that continually increments, once the reference signal exceeds the input signal the counter value is considered as the ADC value. The general principle of operation can be seen in Figure 2-1, an important detail about the ramp ADC is the requirement for a reference clock, and sample-and-hold circuit. The benefits of using a ramp ADC signal are low overall cost and simplicity, but the greatest drawback of the ramp ADC is the variable conversion time, which renders this convertor useless for frequency dependent measurements.

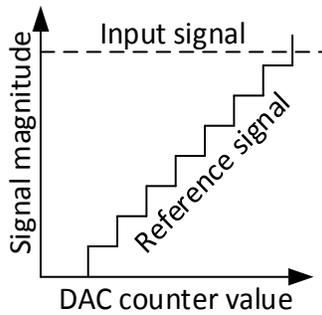


Figure 2-1 Ramp ADC theory of operation.

### 2.2.2 Successive Approximation ADC

The successive approximation ADC works in a similar manner to the ramp ADC, i.e. by comparing a reference and input signal. In this case, the reference signal is generated dynamically by successively comparing it and deciding if digits should be set or cleared, in a progressive manner, as shown in Figure 2-2. The unit responsible for these decisions is called, the Successive Approximation Register (SAR), in a similar way to the ramp ADC, it also requires a clock input to perform operations, but the number of operations is only equal to the number of resolution bits desired. Since the time to generate the actual ADC value is constant and low, this ADC architecture has been commonly used for time sensitive circuits such as energy metering applications.

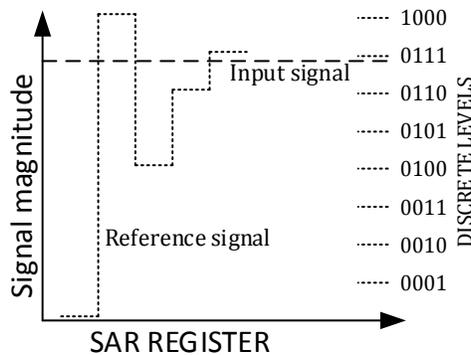


Figure 2-2 SAR ADC theory of operation.

### 2.2.3 Delta Sigma ADC

The delta-sigma ADC, uses a mostly digital architecture, only requiring an analog integrator and comparator, it is based on a single bit ADC. Single bit ADCs can be thought as outputting a “1” if the signal exceeds a certain threshold and “0” otherwise. On delta-sigma ADC this threshold is

dynamically adjusted by integrating the error signal between the input and the calculated value, this calculated value is set to either a  $+V_{ref}$  and  $-V_{ref}$  value according to the ADC output, creating a loop back signal. This loopback architecture can be observed in Figure 2-3, the output signal of a single bit ADC, is worthless in most cases, but if a stream of 1 bit conversions is stored, the pattern can be interpreted into an 'n' bit output by averaging, this is also known as delta modulation [46].

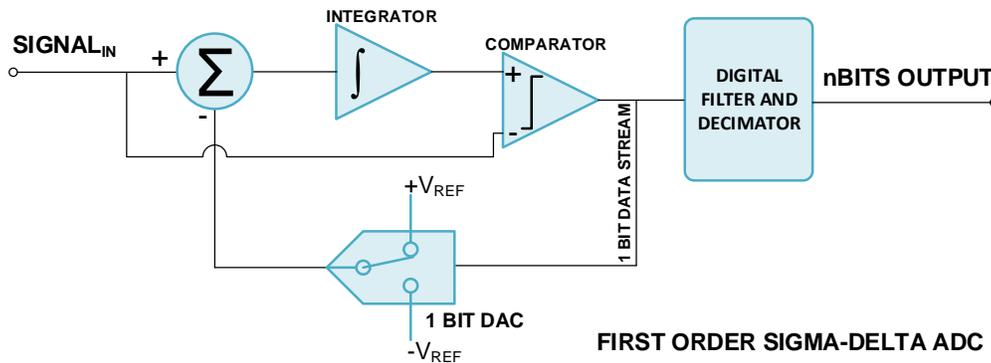


Figure 2-3 Components of a Sigma-Delta ADC, adapted from [46].

Delta modulation allows storing signal changes, or in this case error feedback, this data must be grouped and averaged (filtered) to give a meaningful value, in Figure 2-4 a simple delta-sigma ADC was modeled in Matlab™ by the author, considering a symmetrical  $V_{ref}$ , and a fixed input signal. The red colored signal represents the integrator output, oscillating around the input signal; this integrator integrates the difference between the DAC signal and the signal input, see Eq. 2.1. The integrator output is compared with the input signal once again to output a bit stream, the bit stream is 1 if the integrator is greater or equal to reference signal and 0 otherwise, the comparator output is shown in Figure 2-5, where the red line represents the discrete steps (i.e. 0's and 1's)

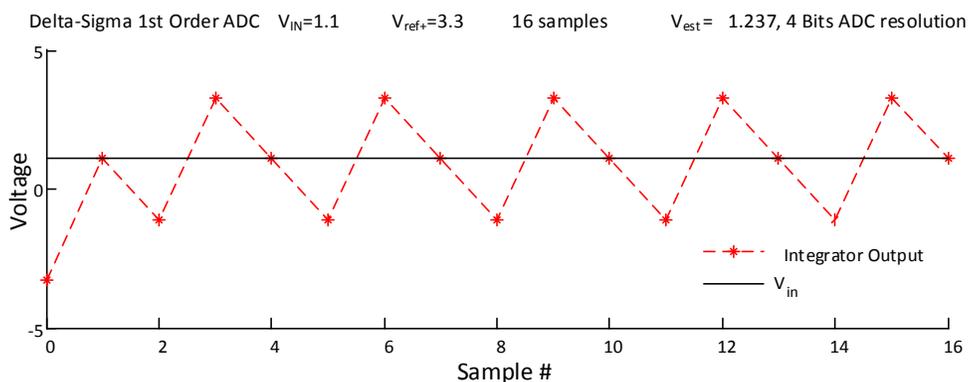


Figure 2-4 Sample Quantization of a Sigma-Delta ADC, integrator vs input signal.

$$\int_{input} = Signal_{in} - DAC_{output} \quad \text{Eq. 2.1}$$

The decimator unit works as an “n bit” window averaging filter, that constantly samples the comparator output stream and generates the average value of an “X number” of window samples, for example in Figure 2-5 the delta-sigma comparator output is discretized to obtain a stream of ‘0’ and ‘1’. This stream is then fed into a 4-bit sized window-averaging filter, after several sample windows; the “n bit resolution” quantized value of the input signal is generated. This underlying technique allows to generate any “n bit” resolution value, as long a high number of window samples are available, thus delta-sigma ADCs rely on oversampling techniques and only work for relative low speed signals, often hundreds of times the Nyquist frequency. In practice, other filters are inside the delta-sigma modulators, which enable to filter noise and to reduce the number of required window samples to output a value within statistical confidence.

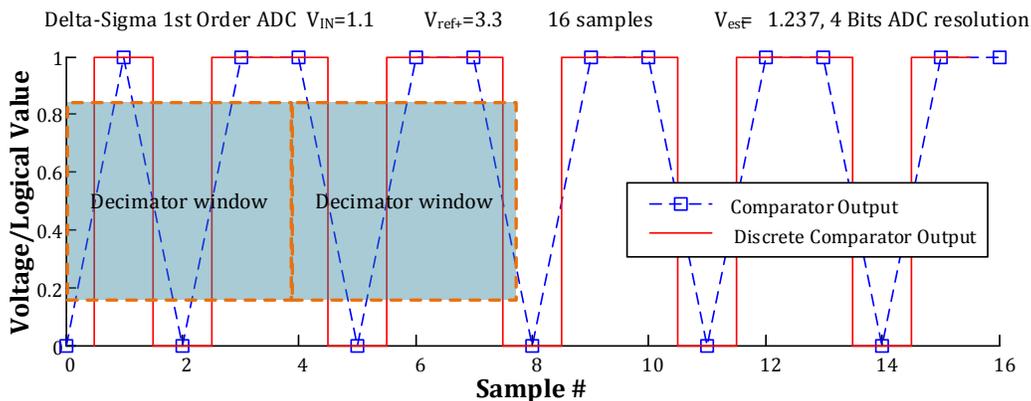


Figure 2-5 Sample Quantization of a Sigma-Delta ADC, outputting 4 Bits

As mentioned earlier, increasing the decimator window size increases the resolution output levels and hence precision of the ADC (if enough data windows are available). In Figure 2-6 a 5-bit decimator is exemplified, the principles are the same as the previous 4-bit example, but with improved precision, following the same principles Table 2.1 was obtained, it shows the decimator window size (ADC bits) and the ideally measured value. Since ADCs have  $\pm\frac{1}{2}$  digit quantization error the obtained value actually represents a possible range that depends on the signal step, this uncertainty range is given on columns 4 and 5 of Table 2.1.

Table 2.1 Decimation filter to obtain an 'n' bits output

ADC Bits	Digital Output	IDEAL OUTPUT	Uncertainty range	
			MIN VALUE	MAX VALUE
2	3/4	1.65	0.825	2.475
3	5/8	0.825	0.4125	1.2375
4	11/16	1.2375	1.0313	1.4437
5	21/32	1.0313	0.9281	1.1344
6	43/64	1.1344	1.0828	1.1859
7	85/128	1.0828	1.057	1.1086
8	171/256	1.1086	1.0957	1.1215

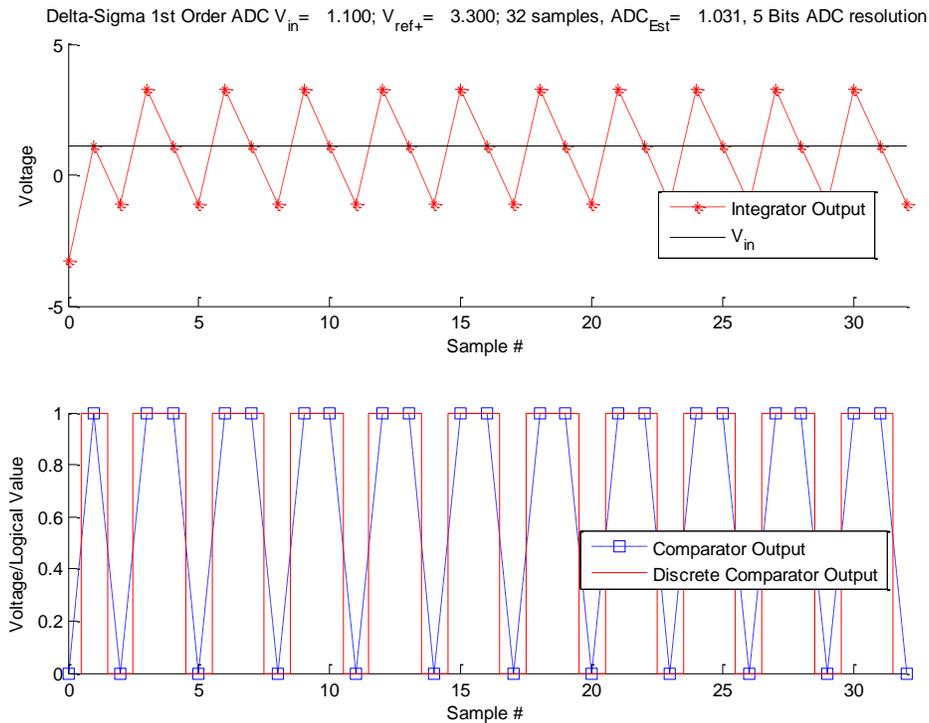


Figure 2-6 Sample Quantization of a Sigma-Delta ADC, outputting 5 Bits

Although in theory near infinite resolution ADC are possible, they would require infinite samples and hence and infinite sampling speeds, in practice delta sigma ADCs are limited by their oversampling speed, which in turn is limited by the integrator speed and its respective comparator. In Figure 2-6 the effect of a rapidly changing signal vs the comparison speed can be observed, this lagging effect can cause incorrect conversion values, to prevent this, delta-sigma ADCs require sharp antialiasing filters near their oversampling speed (usually in MHz) additional to traditional antialiasing filters for ADC's.

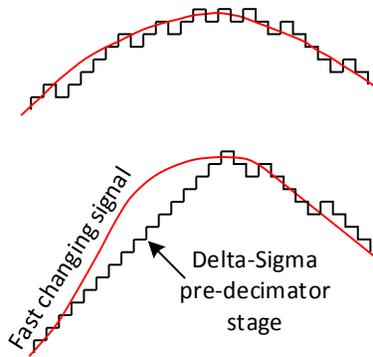


Figure 2-7 Delta-sigma conversion speed limit, adapted from [46].

### 2.3 Filters

Filters are networks of elements that process signals in a frequency-dependent manner, absorbing certain frequencies by grounding those signals, in its basic form they can be thought as a frequency dependent signal divisor, which works on the same principals as the voltage divisor principle. The underlying transformation results on the necessity of using a transfer function to express its frequency response output in terms of the input signal.

A filter will affect the amplitude of a signal, as well as its phase. This is called the phase response and might alter the overall waveform shape depending on the signal frequency components and filter cutoff frequency. These distortions manifest themselves as signal overshoots, in Figure 2-8 the effect of overshooting can be clearly seen for 8<sup>o</sup> order low pass filters given a square input, wherein the Bessel filter offers a constant delay for all pass band frequencies, outputting a smoothed but clear response. The effects of the phase response are of particular interest for designing frequency measurement devices, which will be described in detail in chapter 6 of this work.

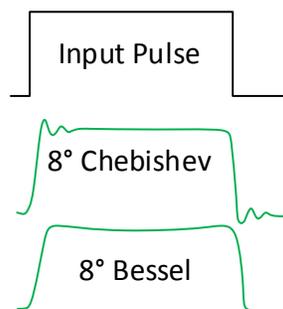


Figure 2-8 Group delay characteristics, for eight order filters given a square input, adapted from [47].

### 2.3.1 *Low pass filters*

Low pass filters (LPF) allow low frequency components to pass through the filter, this is particularly useful for removing system noise and preventing aliasing effects on ADCs capturing circuits. Aliasing effects appear when system frequencies seem to be lower than they are actually are, aliasing depends on the sampling speed of signals and can be eliminated by filtering signals above the sampling rate of the ADC.

### 2.3.2 *Butterworth*

The Butterworth filter is the best compromise between attenuation and phase response; it has no ripple in the pass-band or the stop-band, and as such is considered to have maximally flat response. The Butterworth filter achieves its flatness at the expense of a relatively wide transition region from pass band to stop band, with average transient characteristics.

### 2.3.3 *Bessel*

Butterworth filters have good amplitude and transient behavior. The Chebyshev filters improve on the amplitude response at the expense of transient behavior (band pass ripple). The Bessel filter is optimized to obtain a better transient response due to its linear phase design in the pass band; this allows a constant line delay, useful in audio applications and in places where the delay is constant, such as frequency measurement. This also means that there will be relatively poorer frequency response (less amplitude discrimination).

A Bessel filter can be designed by creating certain limitations, specifically by a linear phase delay that can be modeled from Eq. 2.2 (Phase delay equation) [47], or more generally represented by Eq. 2.3 (Group delay equation) [47].

Phase delay

$$\theta = -\omega D \quad \text{Eq. 2.2}$$

Group delay

$$D = -\frac{d\theta}{d\omega} \quad \text{Eq. 2.3}$$

### 2.3.4 Active filters

Active filters include an additional Operational Amplifier (Op-Amp) to a network of RC components; RC networks alone are susceptible to load impedance, which affects their performance and filtering characteristics. Op-Amps can be ideally thought as an infinite impedance load (input side), zero output impedance, and infinite gain amplifier [48]; this component can be used to interconnect a series of filter stages while isolating characteristics of each one of them, thus preserving the intended response. Due to their non-ideal nature, physical Op-Amps should be chosen according to the designer specifications [49].

Active filters allow the designer to obtain higher than unity gain factors, and simplify filter design by using filter blocks. For example, a third order low pass filter can be made by using a series of well-known structures, such as first order filters, or a second order filter plus a first order filter [49], this is also known as filter chaining [47]. In Figure 2-9a a first order low pass active filter, is shown, while Figure 2-9b. shows a high pass active filter, both are configured in a unity non inverting configuration.

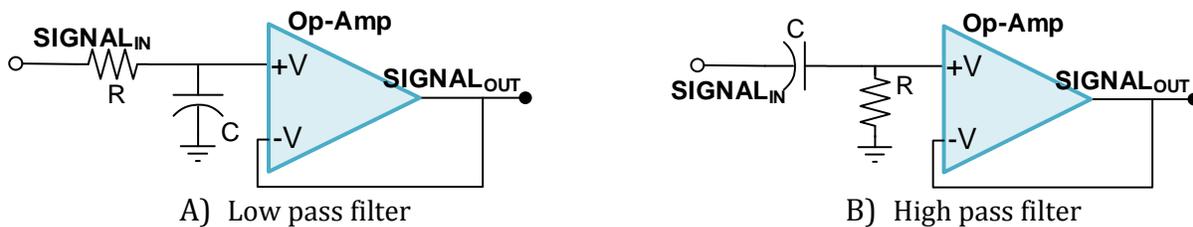


Figure 2-9 Fundamental first order active filter circuits.

### 2.3.5 Key-Sallen topology

As mentioned earlier there exist a set of well-known filter topologies, one example is the Sallen-key topology, it features a reduced number of components for a second and third degree filters, while providing high frequency noise rejection by the inclusion of capacitor in its feedback circuit for its low pass configuration [48]. Sallen-key circuits can be designed to meet characteristic damping factors of common filter responses, Sallen-key filters exhibit high sensibility coefficients for high Q order filters [50], limiting their use to low order filters. R. P. Sallen and E. L. Key developed the classical circuit in the 1950's [51], Figure 2-10 shows the classical second order filter, its transfer function considering an ideal Op-Amp is shown in Eq. 2.4. A third order filter can be accomplished

by cascading a first order filter or by inserting a modification as shown in Figure 2-11 the transfer function of a third order filter is shown in Eq. 2.5 [52].

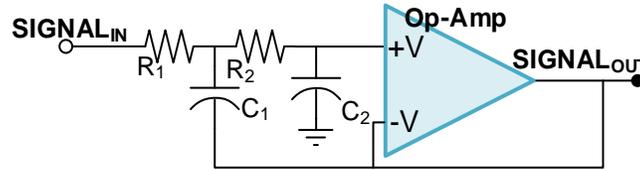


Figure 2-10 Basic Second order Sallen-Key filter topology

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{R_1 C_1 R_2 C_2} \frac{1}{S^2 + S \left( \frac{1}{R_2 C_1} + \frac{1}{R_1 C_1} \right) + \frac{1}{R_1 C_1 R_2 C_2}} \quad \text{Eq. 2.4}$$

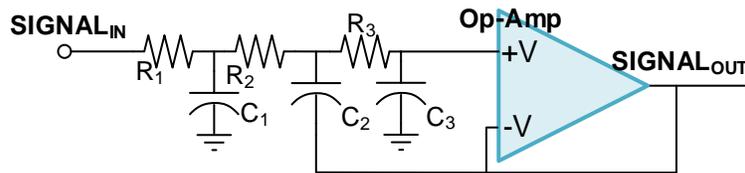


Figure 2-11 Modified third order Sallen-Key filter topology using a single Op-Amp

$$\frac{V_{out}}{V_{in}} = \frac{1}{R_1 C_1 R_2 C_2 R_3 C_3} \frac{1}{S^3 + S^2 \left( \frac{1}{C_1 R_1} + \frac{1}{C_1 R_2} + \frac{1}{C_2 R_3} + \frac{1}{C_2 R_2} \right) + S \left( \frac{1}{C_2 C_3 R_2 R_3} + \frac{1}{C_1 C_2 R_2 R_3} + \frac{1}{C_1 C_2 R_1 R_3} + \frac{1}{C_1 C_2 R_1 R_2} \right) + \frac{1}{R_1 C_1 R_2 C_2 R_3 C_3}} \quad \text{Eq. 2.5}$$

Due to the complexity of designing low sensitivity filters with the desired response output, several approaches have been devised, choosing large capacitor can incur in large Printed Circuit Board (PCB) space, while the use of small capacitors can be adverse due to the existence of parasitic capacitance [53]. Sensitivity analysis have shown that capacitor values should be selected as disperse as possible, while resistor values should be closed by [54], in order to ease design procedures an online third order filter design tool is available at [52].

## 2.4 Fourier Series

The Fourier series allows any periodic signal to be decomposed into a series of sine and cosine waveforms, given the condition that the integral of the signal exists, that is to say that the signal must have a finite power over the period [55]. This definition can be expanded to the electrical power system, since the electrical system is a physics ruled system, then it has a finite power, thus all electrical signals present in a network can be decomposed by the Fourier signals.

The Fourier series allows any electrical periodic signal to be represented by an infinite sum of sinusoidal terms, plus a DC component, as shown in Eq. 2.6 [55]; this allows the Fourier series to transform time domain quantities to the frequency domain.

$$f(t) = A_0 + \sum_{i=1}^{\infty} B_i \text{Sen}(i\omega t) + C_i \text{Cos}(i\omega t) \quad \text{Eq. 2.6}$$

where:

$$\begin{aligned} A_0 &= \text{DC component} \\ \omega &= \text{Fundamental frequency} \\ i &= \text{Frequency multiple} \\ [B_i, C_i] &= \text{Frequency multiple amplitude} \end{aligned}$$

A classic example of the Fourier series is the decomposition of a square wave of frequency  $\omega = 2\pi/s$ , which is composed of an infinite sum of odd harmonic frequencies (a harmonic in this case, is an integer multiple of the fundamental frequency) which exhibits a decaying magnitude ( $Mag_i = \frac{4}{\pi i}$ ). In Figure 2-12, the first four odd harmonics are plotted, each of them using an individual row; each row shows the effect of considering up to the  $n^{\text{th}}$  odd harmonic (in this case up to the 7th), for each harmonic added, a frequency domain plot is generated to show the decaying existence of odd harmonics.

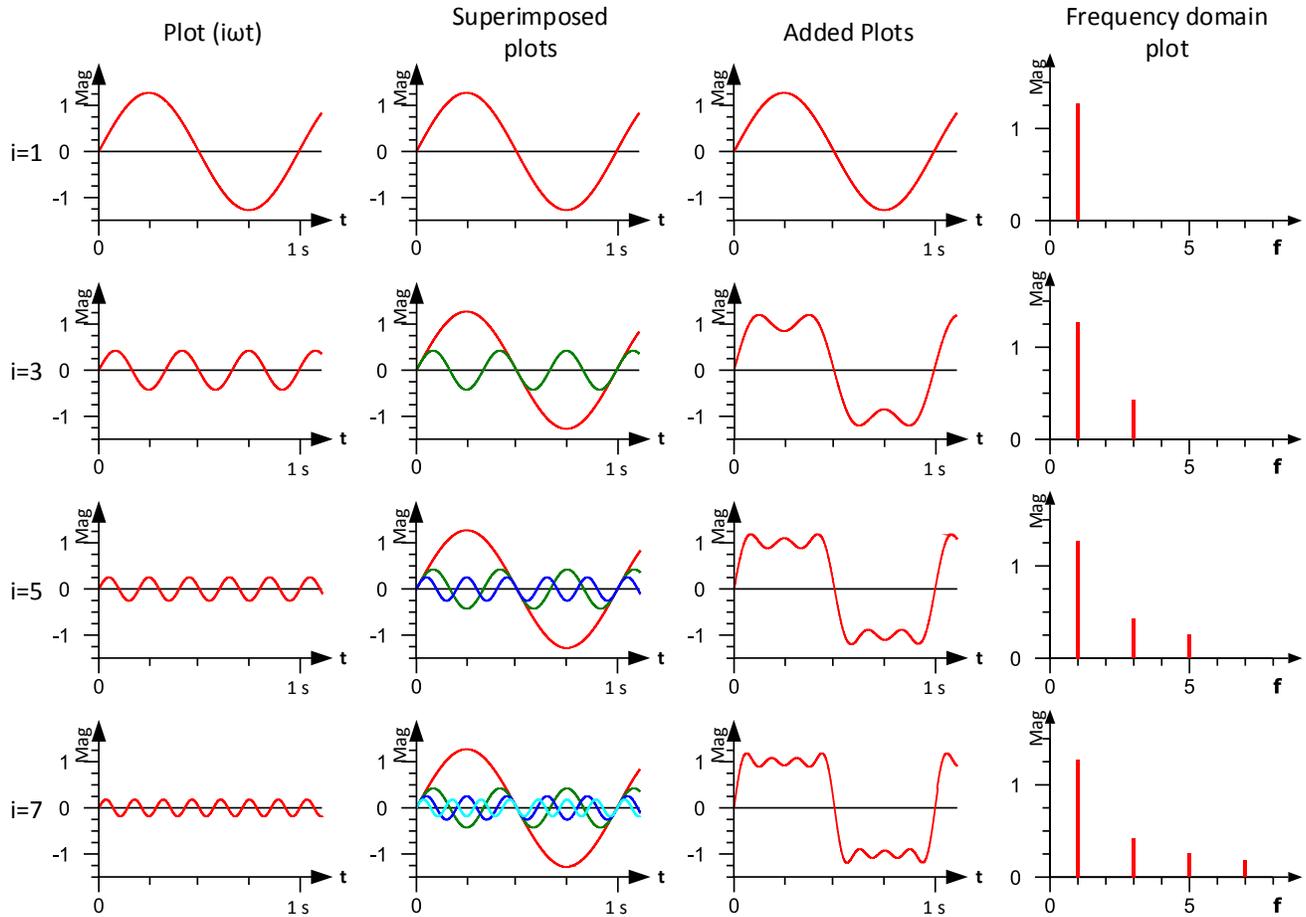


Figure 2-12 Square wave decomposition, first four odd integer frequencies, adapted from [56]

As it can be seen from Figure 2-12 the higher the number of considered harmonics, the closer to a square wave shape is achieved. In measurement devices, a signal should be ideally decomposed to an infinite set to be correctly measured, but in practice, this is limited to a finite number due to the sampling limitations of hardware and available computation resources.

## 2.5 Harmonics

Periodic functions repeat themselves at regular intervals, formally “a function  $f$  is periodic of period  $T$  ( $T > 0$ ) if, only if  $f(t + T) = f(t)$  for all  $t$ ” [57]. In electrical systems the period represents an electrical cycle, and the period  $T$  is equal to  $1/F_0$ , where  $F_0$  represents the fundamental frequency of the system, which for North America is 60 Hz. Although most circuit analysis books deal with an ideal sine wave to represent the periodic function in the real world these waveforms are distorted, mainly because the presence of nonlinear loads.

Distorted waveforms are mainly present in the current waveforms and to a lesser extent in the voltage signal, since this distortion causes adverse effects on the power quality factors, increasing energy losses, and reducing equipment life [58], the level of distortion must be measured so it can be further studied. This distortion is often given by the total harmonic distortion factor, or expressed in terms of harmonics.

Harmonic sources have been studied, and in some cases corrected, on large power systems, meanwhile on the residential systems, the effects were mostly ignored. New power efficiencies requirements, customer-oriented QoS standards and increasing residential nonlinear loads, have raised red flags on harmonics originating at the distribution feeders [59], most harmonics on the distribution power system, are usually due to the following residential loads, and/or field equipment.

- Transformers
- Compact Fluorescent Lamps
- Electronic controls (speed variable controls, washing machines, water pumps)
- PMW based power supplies
- Rectifiers-All DC operating apparatuses (battery changers, printers, routers, audio systems, LCD TVs)

## 2.6 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a mathematical tool that enables to decompose a discrete signal into a finite set of sinusoidal waveforms; these waveforms represent the signal spectral components, which in turn can be used to assess the harmonic contents of an electrical signal (speaking in the electrical field). As seen previously on Eq. 2.6, any signal is composed of an infinite set of waveforms, but the DFT only allows to quantify the existence of a finite set of waveforms (up to the Nyquist frequency) by using correlation methods (thus the finite label). These correlations are done for each of the period waveforms that are suspected to be present in the original signal. The correlation factors are obtained by multiplying a  $\omega$  period input signal by a  $\alpha\omega$  periodic sine

waveform at each sample point, this correlation is repeated by an offset angle waveform (the cosine function) to obtain a scaled magnitude and angle, which represent a harmonic signal in electrical systems.

In Table 2.2 spectral decomposition formulas are shown according to the factors required by Eq. 2.6, these formulas must be applied for each spectral frequency of interest, causing a large computational burden when large window sizes are managed, or the number of interest frequencies are large. In computational terms the complexity of DFT for N samples is defined as  $(O)N^2$ .

Table 2.2. Spectral decomposition formulas.

$B_i = \frac{2\Delta t}{T} + \sum_{i=1}^n S_i \text{Sen}(f\omega t_i)$ $C_i = \frac{2\Delta t}{T} + \sum_{i=1}^n S_i \text{Cos}(f\omega t_i)$ $A_i = \sqrt{B_i^2 + C_i^2}$
<p>where:</p> <ul style="list-style-type: none"> <li><math>i = i^{\text{th}}</math> sample number</li> <li><math>S = \text{Input signal}</math></li> <li><math>f = \text{frequency multiple of input signal}</math></li> <li><math>A = \text{Component Amplitude}</math></li> <li><math>n = \text{Sample number}</math></li> <li><math>\Delta t = \text{Time between samples}</math></li> <li><math>T = \text{Signal period}</math></li> </ul>

## 2.7 Fast Fourier Transform

The Fast Fourier Transform is an optimized version of the DFT, by eliminating redundant operations, using divide and conquer algorithms and exploiting symmetry, it enables computer efficient signal decomposition. It introduces the requirement of power of two ( $2^n$ ) sample numbers and outputs all the integer spectral decomposition frequencies up to the Nyquist limit for a given window size, meaning it can be used for a variety of applications requiring fast signal decomposition. Fast Fourier implementations can be done in floating point math, and integer math, with its respective benefits and drawbacks, but it's ideally suited for math libraries that handle complex numbers (due to twiddle factors), and those compilers that enable recursion.

## 2.8 Energy Metering

Energy metering is used to obtain a monetary value for the energy consumed in a given period of time, at the most fundamental level; registered energy consumption is multiplied by a monetary factor to obtain a monthly statement. Energy consumption is usually recorded in kWh and/or kVArh depending on the type of client, utility policies and load characteristics. Utilities companies generate revenue by charging consumers their consumption while deducing generation and transmission costs. Current generation and transmission networks are highly optimized areas of the energy market, leaving energy metering with the potential of improving revenues.

A kWh is a unit of energy used to represent a 1000 watt consumption over a period of time (hours), similarly kVArh is used to represent reactive power consumption over a period of time (hours), these units (along with the power factor) have been traditionally used as measuring parameters. Power is normally measured by using voltage and current signals; Eq. 2.7 shows the general equation for obtaining real power, which is often employed to obtain a monetized value of the consumed energy.

$$P = |\mathbf{V}||\mathbf{I}| \cos(\theta_v - \theta_i) \quad \text{Eq. 2.7}$$

Similarly active, reactive power and the power factor can be calculated by equations [60], as it can be seen all equations consider a phase difference between phasor  $\mathbf{V}$  and  $\mathbf{I}$  and both signals are sinusoidal in shape.

$$\begin{aligned} Q &= |\mathbf{V}||\mathbf{I}| \sin(\theta_v - \theta_i) \\ S &= \mathbf{V} * \mathbf{I} = \sqrt{P^2 + Q^2} \\ PF &= \cos(\theta_v - \theta_i) \end{aligned} \quad \text{Eq. 2.8}$$

Each utility assigns weights, relations, as well as time of use coefficients to determine a monetary amount due over a determined time. On the last 50 years, the use of nonlinear loads has growth [61]; requiring utilities to measure nontraditional parameters, such as harmonics, distortion factors and to store them in detailed logs, these parameters require a more detailed modeling of the signal characteristics that can be done using traditional energy metering algorithms.

## 2.9 IEEE 1459

As mentioned earlier, modern energy measurement requires to consider non-sinusoidal waves (due to nonlinear loads) or unbalanced power conditions, IEEE has published standard 1459-2000 titled “IEEE Trial-Use Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, balanced, or Unbalanced Conditions” [61] to address those issues. IEEE 1459-2000 considers fundamental and non-fundamental (due to harmonics) energy quantities, creating in some cases new definitions to describe quantities according to their source. Table 2.3 enlists fundamental energy quantities as described in the standard, most additions are in the harmonics components. Similarly, in Table 2.4 some non-fundamental quality indicators are given.

Table 2.3 Fundamental energy quantities described on IEEE 1459-2000

Symbol	Name	Units
$V$	Voltage	$V$
$I$	Current	$A$
$V_1$	Fundamental voltage	$V$
$I_1$	Fundamental current	$A$
$V_H$	Harmonic voltage	$V$
$I_H$	Harmonic current	$A$
$S$	Complex power	$VA$
$P$	Active power	$W$
$N$	Nonactive power	$var$
$S_1$	Fundamental apparent power	$VA$
$P_1$	Fundamental active power	$W$
$Q_1$	Fundamental reactive power	$var$
$P_{F1}$	Fundamental power factor	---
$S_N$	Nonfundamental apparent power	$VA$
$S_H$	Harmonic apparent power	$VA$
$P_H$	Harmonic active power	$W$
$D_I$	Current distortion power	$var$
$D_V$	Voltage distortion power	$var$
$D_H$	Harmonic distortion power	$var$
$S_N/S_1$	Harmonic pollution	---

Table 2.4 Non-fundamental quality indicators described on IEEE 1459-2000

Symbol	Name	Units
$THD_V$	Total harmonic distortion (Voltage)	--
$THD_I$	Total harmonic distortion (Current)	--

In Table 2.5 the fundamental energy quantities are grouped together according to the traditional nomenclature, where each traditional component is composed of a fundamental quantity plus additional harmonics.

Table 2.5 Summary of energy quantities grouped by source, adapted from [61].

Quantity	Combined	Fundamental powers	Non Fundamental powers
Apparent	$S$	$S_1$	$S_N S_H$
Active	$P$	$P_1$	$P_H$
Nonactive	$N$	$Q_1$	$D_I D_V D_H$

### 2.9.1 RMS Voltage- Discrete Time domain

Voltage ( $V$ ) is traditionally expressed on terms of the Root Mean Square (RMS), which is equivalent to integrating the signal overtime; according to IEEE 1459 its value depends on the fundamental components in addition to harmonics, the RMS value of a complete cycle on the discrete time domain can be computed according to Eq. 2.9.

$$V = V_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N [x(i)]^2} = \sqrt{V_1^2 + V_H^2} \quad \text{Eq. 2.9}$$

where

$x(\dots)$  = Vector containing the discrete time signal readings

$N$  = number of obtained readings per cycle.

$V_1$  = Voltage due to the fundamental frequency signal

$V_H$  = Voltage due to nonfundamental frequency signals

### 2.9.1 RMS Current- Discrete Time domain

Current is similarly expressed on terms of the RMS value, on Eq. 2.9 the discrete time-domain computation formula is given, with its respective components according to IEEE 1459.

$$I = I_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N [x(i)]^2} = \sqrt{I_1^2 + I_H^2} \quad \text{Eq. 2.10}$$

where

$I_1$  = Current due to the fundamental frequency signal

$I_H$  = Current due to nonfundamental frequency signals

### 2.9.2 *Harmonic current and voltage components*

As mentioned in the introduction of IEEE 1459 a great emphasis is given to quantities generated by harmonic sources, on the standard certain variables are often used to define other quantities, one of these cases is the harmonic voltage and current quantities, which are shown on Eq. 2.11. According to the standard, all non-integer and integer harmonics must be considered, including the DC component, and in order to calculate each of their values a frequency domain representation is suggested.

$$\begin{aligned} I_H &= \sqrt{\sum_{h \neq 1} I_h^2} \\ V_H &= \sqrt{\sum_{h \neq 1} V_h^2} \end{aligned} \quad \text{Eq. 2.11}$$

### 2.9.3 *Total Harmonic Distortion (THD)*

A traditional measure of waveform distortion is the THD, which is represented by Eq. 2.12 where the current and voltage THD formulas are given. According to IEEE 1459, two paths for obtaining the THD are possible but they must be chosen according to the error propagation characteristics of the selected hardware architecture.

$$\begin{aligned} THD_V &= \frac{V_H}{V_1} = \sqrt{\left(\frac{V}{V_1}\right)^2 - 1} \\ THD_I &= \frac{I_H}{I_1} = \sqrt{\left(\frac{I}{I_1}\right)^2 - 1} \end{aligned} \quad \text{Eq. 2.12}$$

### 2.9.4 *Active Power*

According to IEEE 1459, the active power (Eq. 2.13) is composed of the useful power  $P_1$  (Eq. 2.14) and harmonic active power  $P_H$  (Eq. 2.15). This definition is used to differentiate the active power that generates useful work vs the one that only generates heat on rotating machines.

$$P = P_1 + P_H \quad \text{Eq. 2.13}$$

$$P_1 = V_1 I_1 \cos(\theta_1) \quad \text{Eq. 2.14}$$

$$P_H = \sum_{i \neq 1} V_h I_h \cos(\theta_h) \quad \text{Eq. 2.15}$$

Where

$$\begin{aligned} \theta_1 &= \text{Phase angle difference between } V_1 \text{ and } I_1 \\ \theta_h &= \text{Phase angle difference between } V_h \text{ and } I_h \end{aligned}$$

### 2.9.5 **Reactive power**

According to IEEE 1459, the reactive power is only composed of fundamental components, and can be computed according to Eq. 2.16.

$$Q_1 = V_1 I_1 \cos(\theta_1) \quad \text{Eq. 2.16}$$

### 2.9.6 **Fundamental apparent power ( $S_1$ )**

The fundamental apparent power is a core energy quantity that is used by utilities to measure energy consumed by its customers, and as such, for historical reasons only considers a purely sinusoidal waveform, to assure backward compatibility IEEE 1459 defines the fundamental apparent power as the product the fundamental V and I components (Eq. 2.17)

$$S_1 = V_1 I_1 = \sqrt{P_1^2 + jQ_1^2} \quad \text{Eq. 2.17}$$

### 2.9.7 **Non-fundamental apparent power ( $S_N$ )**

The non-fundamental apparent power is used to compensate the deviation between the calculated apparent power ( $S_1$ ) and measured apparent power under non-sinusoidal conditions, its components can be separated by Eq. 2.18

$$S_N = \sqrt{S^2 - S_1^2} = \sqrt{D_I^2 + D_V^2 + S_H^2} \quad \text{Eq. 2.18}$$

### 2.9.8 **Other quantities described by IEEE 1459**

On IEEE 1459, there other auxiliary components that are required to describe energy consumption under non-sinusoidal conditions, and these are given by Eq. 2.19.

$$\begin{aligned} D_I &= V_1 I_H = S_1 (THD_I) \\ D_V &= V_H I_1 = S_1 (THD_V) \\ S_H &= V_H I_H = S_1 (THD_I) (THD_V) \end{aligned} \quad \text{Eq. 2.19}$$

$$D_H = \sqrt{S_H^2 - P_H^2}$$

$$N = \sqrt{S^2 - P^2}$$

### 2.9.9 Apparent power

On the IEEE 1459 standard, the apparent power maintains the traditionally definitions of  $VI$  used on most electrical engineering courses, but introduces the 3D space representation to account for the harmonic components; the apparent power is thus composed of traditional P, Q components plus an additional axis known as the distortion power (D). On Eq. 2.20 the apparent power is fully decomposed into a three-axis system, which enables it to construct the 3D representation given in Figure 2-13.

$$S = VI = \sqrt{S_1^2 + S_N^2} = \sqrt{P_1^2 + jQ_1^2 + S_N^2} = \sqrt{P_1^2 + jQ_1^2 + D_i^2 + D_v^2 + S_H^2} \quad \text{Eq. 2.20}$$

$$= \sqrt{P_1^2 + jQ_1^2 + D_i^2 + D_v^2 + D_H^2 + P_H^2} = \sqrt{\hat{i}(P_1^2 + P_H^2) + \hat{j}Q_1^2 + \hat{k}(D_i^2 + D_v^2 + D_H^2)}$$

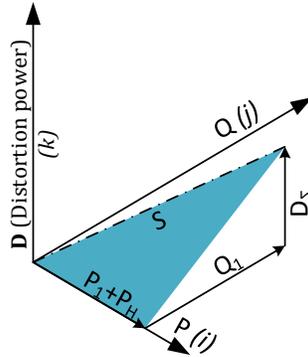


Figure 2-13. The 3D approach to explain S, according to IEEE 1459.

### 2.9.10 Fundamental power factor

On the IEEE 1459 standard, the fundamental power factor only depends on the values of the fundamental real and apparent power quantities; its relationship is defined by Eq. 2.21.

$$P_{F1} = \cos\theta_1 = \frac{P_1}{S_1} \quad \text{Eq. 2.21}$$

### 2.9.11 Power factor

An additional quantity is listed on the IEEE 1459 standard to enable backward compatibility with the widely used power factor (P.F) term, and can be calculated according to Eq. 2.22.

$$P_F = \frac{P}{S} = \frac{P_1 + P_H}{\sqrt{S_1^2 + S_N^2}} \quad \text{Eq. 2.22}$$

### 2.9.12 Vector apparent power (3-phase systems)

Lastly, an apparent power quantity is introduced on the IEEE 1459 standard, designed to measure the total amount of supplied apparent power to three-phase loads. This quantity is named the “Vector apparent power”  $S_V$  and it differs from the “complex power” term by considering the distortion power (D). This process can be described mathematically by Eq. 2.23 and graphically by Figure 2-14, in this case, the “complex power” is defined as the Arithmetic apparent power ( $S_A$ )

$$S_{V(l,j,\hat{k})} = [\hat{i}(P_A) + \hat{j}(Q_A) + \hat{k}(D_A)] + [\hat{i}(P_B) + \hat{j}(Q_B) + \hat{k}(D_B)] + [\hat{i}(P_C) + \hat{j}(Q_C) + \hat{k}(D_C)] \quad \text{Eq. 2.23}$$

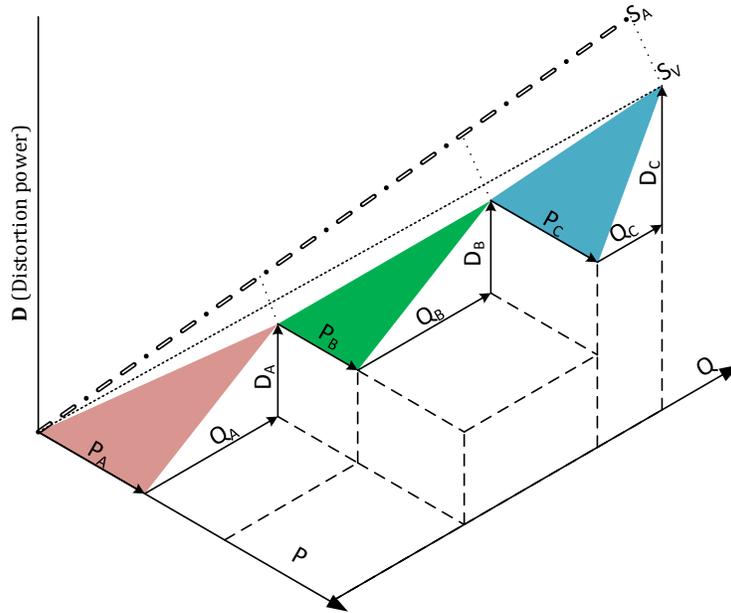


Figure 2-14 Arithmetic apparent power ( $S_A$ ) and Vector Apparent power ( $S_V$ ) under unbalanced non-sinusoidal conditions, adapted from [61].

### 2.10 Phasor Measurements Units

Phasor Measurement Units (PMU) enable network operators to observe the dynamic state of the power system in near real time [62]. These units measure the electrical signals (voltage and current) in a phasor manner, adding a time stamp that enables subsequent comparisons to be performed with other time-synchronized units. In order for these measurements to be useful, certain precision levels, as well as communication protocols must be met; these are dictated by IEEE C37.118.

IEEE C37.118 “IEEE Standard for Synchrophasors for Power Systems” dictates the accuracy and data reporting requirements for PMU units, providing a testing framework to assess overall signal registration capabilities under certain transient events, these events include rapid frequency variations, signal swells and harmonic signal contamination.

### 2.10.1 *Hardware components*

Phasor measurement units consist of several core hardware components that are illustrated by Figure 2-15. The first part of the unit transforms analog signals into a frequency domain representation, after this, a time stamp usually provided by a Global Positioning Unit (GPS) receiver is appended to the conversion. Once a conversion is time-stamped, it’s usually queue for subsequent data transmission to a central office, where network operators often visualize the readings of multiple units.

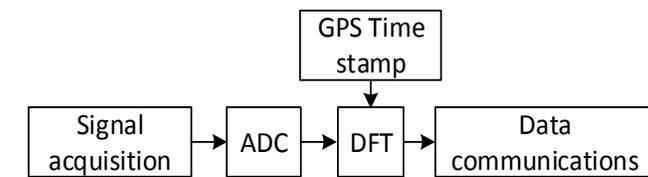


Figure 2-15. Basic hardware components of PMU units.

As it can be seen from Figure 2-15 many of the hardware components of a PMU are shared with the smart meter architecture proposed on chapter 1, and thus a hybrid design that contains both functionalities is possible by appending the GPS receiver unit. The GPS receiver works by listening to the data transmissions coming out of a network of satellite units that are constantly rotating around the earth, these data transmissions include precise time references that are decoded by the receiver unit archiving in most cases better than 1  $\mu S$  time precision.

### 2.10.2 *Phasor signal representation*

As mentioned by the introduction, PMUs report sinusoidal signals in a phasor form; phasors can be described mathematically by Eq. 2.24. This representation is the preferred method for determining wide area stability in power systems, and some relative new techniques employ this information to detect system instability before it actually occurs.

$$x(t) = X_m \cos(\omega t + \phi) \quad \text{Eq. 2.24}$$

where:

$$\begin{aligned} X_m &= \text{Waveform peak value} \\ \omega t &= \text{System angular frequency} \\ \phi &= \text{Phase angle} \end{aligned}$$

Recently PMUs have started to appear on distribution networks due to the inclusion of distributed generation; in these cases, a simplified phasor representation can be used, since it is likely that the system frequency remains the same for the same electrical area. This simplified representation is given by Eq. 2.25.

$$x(t) = \frac{X_m}{\sqrt{2}} (\cos(\phi) + j \sin(\phi)) \quad \text{Eq. 2.25}$$

### 2.10.3 IEEE C37.118

PMU units are standardized by IEEE C37.118, on the 2011 revision two PMU classes are specified, these are the M (Measurement) class and the P (Protection) class. Their operation is classified into two operating states; these are called the stationary and dynamic states, for the stationary state Table 3.6 enlists the measurement requirements for PMU units.

Table 3.6 IEEE C37.118 standard, class requirements for the stationary operation of PMU

	Class M	P class
Data reports per second	15-120	15-100
Error tolerance.	Total Vector Error <1%	
% de harmonic components, up to the 50 <sup>th</sup>	1%	10%
%Voltage signal	10%-120%	80%-120%
%Current signal	10%-200%	10%-200%

The standard uses the term “Total Vector Error” (TVE) to describe the relative error between the accepted (true) value of the signal and the measured signal value, it can be obtained by using Eq. 2.26, while a graphical description of the TVE can be seen on Eq. 2.26

$$TVE = \frac{\sqrt{(\widehat{X}_r - X_r)^2 + (\widehat{X}_i - X_i)^2}}{(X_r^2 + X_i^2)} * 100 \quad (\%) \quad \text{Eq. 2.26}$$

where

$$\begin{aligned} \widehat{X}_r &= \text{Measured value on the real axis} \\ \widehat{X}_i &= \text{Measured value on the imaginary axis} \end{aligned}$$

$X_r =$  Accepted (true) value on the real axis  
 $X_i =$  Accepted (true) value on the imaginary axis

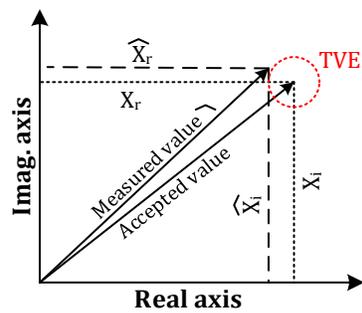


Figure 2-16 The TVE concept, graphical representation.



## CHAPTER 3

### 3. DATA SECURITY IN COMMUNICATIONS.

#### 3.1 Introduction

The need for transmitting information in a secretive manner has puzzled humankind for several centuries. Early examples of cryptography were substitution and transposition ciphers. Substitution ciphers such as the Caesar cipher used in ancient Rome, relied on substituting each letter of a message by another preset letter; transposition ciphers relied on the reversing of words or letter scrambling [63]. These types of early cryptography (colloquially known as classical ciphers) can be easily broken if an attacker recognizes patterns or by mere brute force attacks.

In the XIX century Kerckhoffs's principle laid an important concept of cryptography: "The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience" [64]. Before Kerckhoffs's laid its principle, people relied on the secrecy of the method, rather than the use of a secret key. On modern days, cryptography functions (also known as primitives) are standardized and their inner workings are available to the public, except for countries where export restrictions exist [65].

According to [66] and [67] there are set of objectives that cryptography must address which, grouped together according to their similarity, define a set of basic goals that security services must fulfill to ensure its purpose. These goals are enlisted below.

- 1. Confidentiality and secrecy:** to provide means that keep data contents restricted to those authorized to have it. It refers to the basic property of encoding data according to certain algorithms, which render data meaningless to other parties, since most data most travel by an insecure medium (such as the internet) it prevents eavesdropping.
- 2. Data Integrity:** to implement means to detect manipulation of the encrypted message on its travel path. This manipulation could be related to the addition, removal or replacement of the intended data, also known as message forging.

3. **Authentication:** to procure means to identify the subjects involved in the data communication. Authentication provides at the same time data integrity if the data origin can be verified or trusted.
4. **Non-Repudiation:** to arrange means of verification of previous actions. This can be in a form of a log by a third entity or digital signature.

In addition to the before mentioned goals of a security service provider, there are other optional features that are desirable for certain applications [67]. These features although not universal enable certain entities to communicate under particular constraints or security issues, such as wireless internet networks, enterprise VPNs, and monitoring equipment (e.g. smart meters) or car-to-car communications for collision avoidance. Some additional goals that a security service might fulfill are enlisted below:

5. **Identification/entity authentication:** Provide means to establish, verify, and revoke credentials to participating parties.
6. **Access control:** Provide means to restrict information to authorized entities.
7. **Availability:** Provide means to assure reliability of an entity, even under attacks.
8. **Physical security:** Provide means to obstruct physical tampering of the device, by recording intrusions, or deleting data.
9. **Anonymity:** Provide means to protect the identity of the subject, usage patterns, location or access times.

### 3.2 Crypto Elements Implemented for This Thesis

Although it is often discouraged to implement crypto primitives on production hardware, due to possible pitfalls, in this thesis the author develops a time-optimized *Advance Encryption Standard* (AES) encryption/decryption function, and proposes an algorithm for extracting unique ID's from Static Random Memory (SRAM) memory by using Physical Unclonable Functions (PUF) in order to provide a secure credential storage. The developed crypto elements are essentially intended for use in a smart meter architecture, but could be adapted to other wireless sensor networks.

AES is a cryptographic primitive that needs an ample introduction to be properly implemented, at least from the security point of view, and thus requires a two-part process for its development. Firstly, the mathematical and algorithm background is mostly given on Annex E, with possible attacks described through this chapter. Chapter 7 details most of software implementation characteristics, with an emphasis on code optimization based on the employed microcontroller architecture.

A PUF function extraction algorithm is proposed in section 3.7.4, it is based on the procedure described by authors in [68], but it differs itself by storing byte positions instead of the error correction codes. The proposed method obtains similar results (from the randomness point of view) to the ones presented by [68], at the cost of more memory use, but with an improved response to temperature variations on the SRAM unit.

### **3.3 Cryptographic Terms**

After Kerckhoffs principle established that the security of a system depended on the key and not the actual algorithm inner workings, efforts were done to create standardized cryptographic primitives. Certain properties such as data blocks, substitution, and permutations were explored during the following years, by the 1970's the first standardized cryptographic primitives were published, giving birth to cryptography, as we know it today. In annex D an introduction to cryptographic terms are given.

### **3.4 Cryptography Modes**

There are two basic types of ciphering modes, symmetric or shared key (known password), and asymmetric or public key cryptography, each has certain advantages over the other, and in practice both sets are used to develop a security suite.

#### **3.4.1 *Symmetric-key cryptography***

These families of cryptography algorithms rely on the same cryptographic keys for both encryption and decryption of the plaintext. The keys represent a shared secret that is used to protect a private data link, and it requires each party to have a copy of the key, introducing the difficult task of key distribution and key storage.

Some of the main principles of symmetric cryptography are substitution and transposition. Substitution refers to scheduled-variable replacement of symbols according to the cipher algorithm, while transposition seeks to scramble the input data in an orderable and repeatable manner. Symmetric key algorithms can be further divided in two major types: stream ciphers encrypt data one symbol at a time, while block ciphers encrypt a set of symbols (usually bytes) at the same time. some well-known symmetric algorithms are DES (Data Encryption Standard), 3DES (Triple Data Encryption Standard), and AES (Advanced Encryption Standard)

Symmetric-key cryptography often offers high speed processing [69], short key storages, and it is generally used for bulk encryption.

#### 3.4.1.1 *Crypto Elements of symmetric-key cryptography*

Crypto processes can be represented by a set of equations, for example equation (Eq. 3.1a) can be read as encryption function (E) of message (m) given key (k) outputs an encrypted message(c). The same holds true for equation (Eq. 3.1b) where a decryption function (D) of message (c) given key (k) outputs a decrypted message(c), by joining (a) and (b) one can prove that symmetric key cryptography outputs the original message if only the encryption key is equal to deciphering key (Eq. 3.1c).

$$\begin{aligned}
 E_k(m) &= c & \text{(a)} & \text{Eq. 3.1} \\
 D_k(c) &= m & \text{(b)} & \\
 D_k(E_k(m)) &= m & \text{(c)} &
 \end{aligned}$$

#### 3.4.1.2 *Advanced encryption standard*

The Advanced Encryption Standard (AES) is a symmetrical block encryption specification, published by NIST in 2001 [70]. It works by using a series of substitution-permutations (known as rounds) that provide the required confusion and diffusion properties. AES works in tree key sizes 128, 192 and 256 bits; each key length provides an increasing level of security, while at the same time keeping the procedure simple to implement in hardware or software. The details of the AES rounds are given in detail on annex E.

In Chapter 6 a time optimized AES implementation is presented, this implementation is resistant to timing attacks (see 3.5.1). The AES implementation is useful in implementing the Transport Layer

Security protocol, which allows secure data transmission over TCP/IP connections employed by the proposed smart metering unit.

### 3.4.2 *Asymmetric-Key cryptography*

Although symmetric key cryptography enables secure data communications it imposes, certain requirements that make it impractical to use it as the only cryptographic primitive. Some of the main disadvantages are:

**Difficult key interchange:** In order to enable data communications between two entities, the key must be previously agreed among the parties, which introduces the problem of a secure key interchange channel.

**Large key storage:** If a large secure network based communication is required, then a key pair must distributed for each possible connection, requiring a large key database storage. This can be modeled by Eq. G.7, where  $n$  denotes the number of participating parties [67].

**Non-Repudiation capabilities:** Since the key is shared by at least two entities, it is impossible to recognize the message origin, since either end can create an encrypted message that given the same plaintext is untraceable from a third party point of view.

$$\frac{n \circ (n - 1)}{2} \quad \text{Eq. 3.2}$$

Public key cryptography algorithms depend on the use of a pair of private and public keys, these keys, although different, are mathematically related. The private key allows the encryption of a given message that can only be decrypted by using the public key, providing a simple identity verification system. This also works the other way around, enabling encryption by using the public key and allowing decryption only by the private key holder. Public key cryptography relies on hard to solve mathematical problems (at the present time) as a mean to provide confidentiality. They have the disadvantage of being slow compared to symmetric cryptography. Some well-known public key cryptography algorithms are RSA (Acronym for the inventors) and DH (Diffie-Hellman key exchange).

Public key cryptography introduces another advantage over symmetric-key cryptography, known as digital signatures that allow non-repudiation implementation and certificates for identity verification if a trusted third party exists [71].

### 3.5 Attacks on Cryptographic Security Implementations

Attacks on the cryptographic primitive algorithms is well studied, and their associated risk is considered low for standardized suites, the real risks usually appear during implementation stage and day to day use, for example keys used on AES have a limited life, that depends on the amount of data encrypted [72]. Some attacks related to crypto security that receive media attention are due to unhandled code, unconsidered buffer overflows, memory holes, and programming errors. In the following sections some forms attacks/pitfalls are described.

#### 3.5.1 *Incorrect cipher mode of operation*

Block cipher modes although secure, should be analyzed before use in a particular application, for example, ECB mode should only be used for data containing random data, or pattern-less data. An example of using ECB mode on data set with patterns can be observed in Figure 3-1, in this dataset, a two color image is used, since ECB always outputs the same value for a given input, some information is leaked into the final ECB encrypted image, this leaked information can give out information to an eavesdropper. To prevent these types of attacks, block cipher modes should always be chosen according to their characteristic and real-life use scenario.

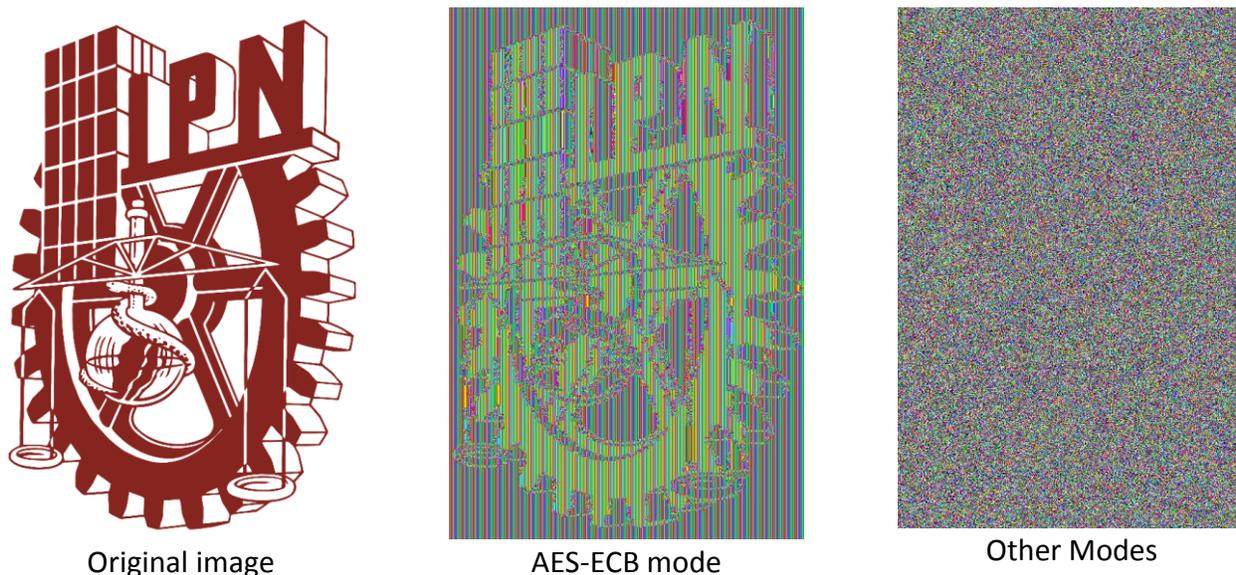


Figure 3-1 Uses of encryption modes, and possible pitfalls.

### 3.5.2 Not using MAC or authentication cipher modes.

The use of encryption cipher block modes does not guarantee data integrity, for example in Figure 3-2, an attacker might insert, delete, or modify an encrypted message without the receiving end noticing the change. These types of attack are easy when plaintext-ciphertext pairs are known, or when non-chaining cipher modes are used, such as ECB with fixed Initialization Vector (IV), in some cases even data can also be regrouped.

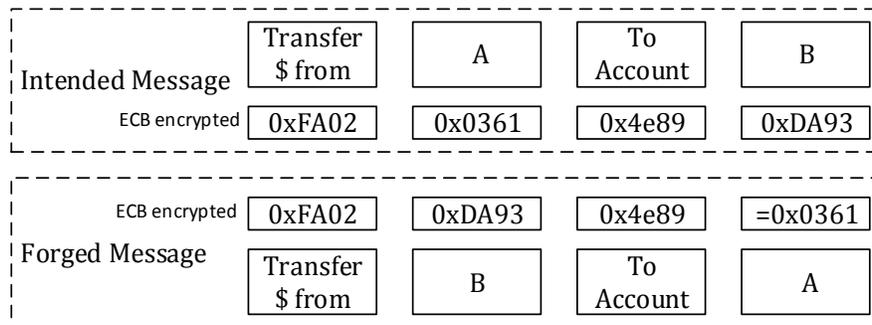


Figure 3-2 Attack Based on the ECB mode, or static/weak IV on CTR mode

Other types of attacks are possible for different ciphers, or cipher modes of encryption, an interesting case is presented for the CBC, although one might initially think that changing data of blocks in the stream will corrupt other near blocks, an invisible attack is possible by changing only the transmitted IV. In this attack, the adversary only has to know the plaintext, and the transmitted IV, he can forge a message by just generating a new IV that masks the original message (see Figure 3-3), since the IV is only used to decrypt the first block, the attack cannot be noticed in most cases. The new IV can be computed by XORing the old IV as shown on Eq. 3.3.

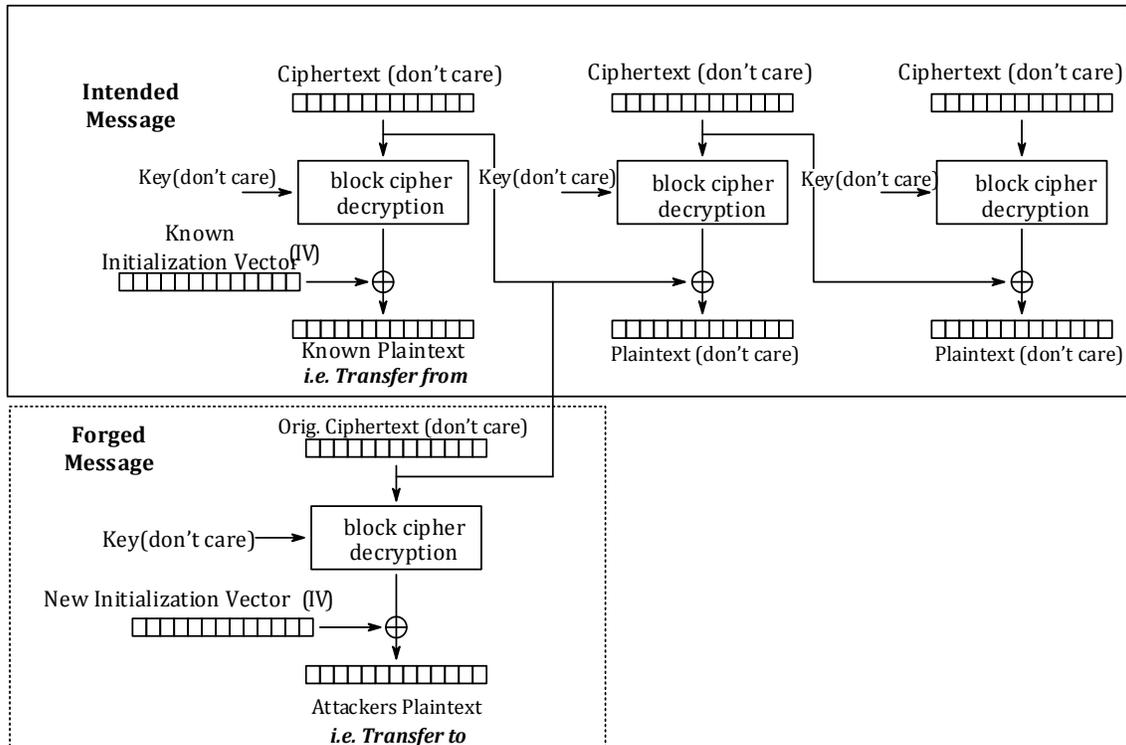


Figure 3-3 CBC attack based on IV masking, for reference the internal CBC decryption mode of operation is shown.

$$IV' = IV \oplus m_{intended} \oplus m_{forged} \quad \text{Eq. 3.3}$$

On the other hand, hash functions should only be used to detect transmission errors, or unintentional data corruption, they should not be used to validate the message origin, unless that information is embedded on the message structure such as the X.509 certificates.

### 3.5.1 Timing attacks

Timing attacks are a form of side channel attack, which attackers can exploit to find additional information about the plaintext, ciphertext or authentication information, these attacks are due to timing variations during execution or processing, in the next sections some of these attacks are described. This is a key aspect of any crypto primitive implementation that must be considered during software development, and it serves as a preamble to section 6.3.3.1 of this work.

#### 3.5.1.1 MAC verification timing attack

Suppose a function 'fc' checks if a sent tag<sub>s</sub> is valid or not for a given message 'm', most algorithms will first compute the tag = f(m, k), and then check if the tag = tag<sub>s</sub>. The timing attack exploits

deficiencies on tag comparison scheme. For example, if the check is done byte per byte and is stopped at the first mismatch; an attacker can make time comparisons to determine up to the  $n^{\text{th}}$  correct byte and work its way up until all the bytes of the  $tag_s$  are valid (see Figure 3-4) thus generating a valid tag for message 'm'.

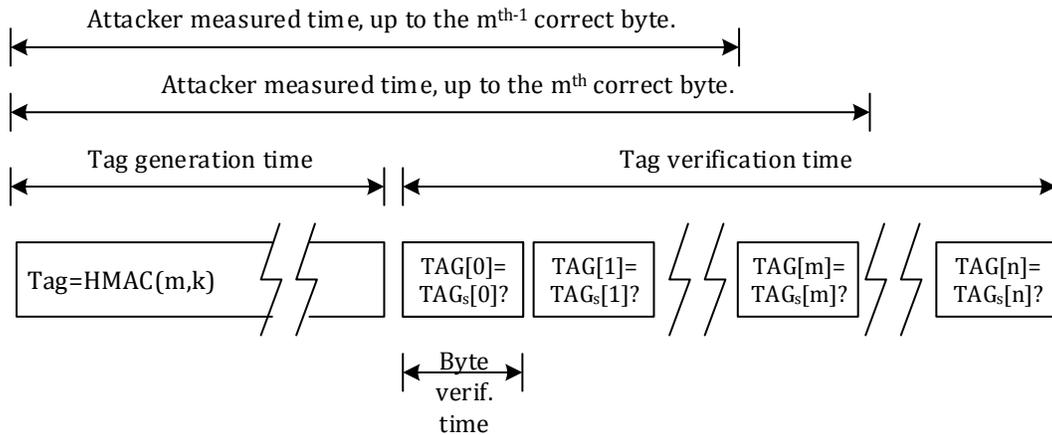


Figure 3-4 MAC verification timing attacks, an attacker successively test bytes from the MSByte up to LSByte, if the byte is correct the response time increases.

To eliminate these types' of attacks, all MAC validations should take the same time, i.e. implementations should compare all bytes before outputting the comparison value, software implementations should also consider compiler optimizations effects.

### 3.5.1.2 Cache memory attacks

The cache unit can be viewed as fast memory unit that optimizes speed execution (see section 6.3.3.1), although beneficial in most cases, its use in cryptography is controversial, since it can be the entry point to several attacks. Cache units work as high speed memory buffers, that contain previously accessed data, if a program queries a RAM location, it will try first to locate the address at the cache (cache hit), and if fails, it will access the RAM space (cache miss). Cache hits and misses cause power and time variations that can be analyzed to obtain information about the key, or original message, these attacks are denominated "time-driven attacks", and they are based on gathering multiple timing information on repeated encryption processes and using correlation techniques to retrieve the key [73].

There are other types of cache attacks, known as "access-driven attacks", these rely on the data remains of a crypto process; for example, when large LUTs are used to lower the execution time

during diffusion and confusion operations (e.g. AES mixcolumns) plaintext can be recovered from the cache. This is due to the fact that cache units have low data capacity, and only a small parts of the LUTs are used, effectively caching only the used entries, an allowing an attacker to construct an attack based on the LUT values used. Access-driven attacks are usually possible on shared resources devices, such as most multitasking operating systems on PC and mobile devices.

Other attacks can exploit the key scheduling operation, by interrupting the AES operation during the first rounds [74], or by gathering cache access patterns through techniques that fall under the “trace-driven cache attacks” [73]. In [74] some countermeasures are given, mostly targeted at restricting cache access, optimizing program flow to avoid memory access, clearing data, or total cache deactivation.

### 3.5.2 *Reverse engineering*

Reverse engineering (RE) is the process of assessing a technology to discover their inner workings and to determine how it operates [75], although reverse engineering is possible in all devices, it is easier to do on consumer products, or field devices due to wide availability and lack of control. RE can be done to improve a design (based on a competitor’s product) or in some cases to steal proprietary information, such as algorithms or databases; in other cases RE can be used to circumvent security features such as copy protections or crypto elements.

#### 3.5.2.1 *Software attacks*

RE can be done via hardware or software access, software in the embedded world represents the firmware and/or special hardware drivers used to bring functionality to a product. Although code is stored in machine code, source code can be extracted from the firmware via decompilation techniques; decompilation allows transforming machine language into a higher-level code such as C. The resulting code is often readable but difficult to comprehend, and it requires plenty of analysis to extract useful information. In cryptographic-embedded solutions, these analyses can retrieve key storage locations, used credentials, and even determine security vulnerabilities, if this vulnerabilities are exploited, the security of a system can be put in jeopardy.

The security risk for a hijacked device depends on its connectivity, economic impact and possible human risks. For example, an altered toaster can cause localized fires only on the compromised

devices endangering human lives, whereas a computer virus can reproduce itself through networks, possibly affecting millions of computer users, on the other hand, specialized viruses can attack bank accounts or even destroy tangible assets, such as the *Stuxnet* worm [76].

There are several ways to access the program code of embedded devices, even when there are copy-protect mechanisms enabled, for example:

- Exploiting open debugging directives
- Dumping external flash memory, if code is stored off the chip.
- Exploiting weak permission JTAG ports (device programming access ports)
- Removing copy-protect fuses (via voltage patterns, chemical etching, etc.)
- Partial firmware replacement.
- Using over the air captured firmware upgrades [77].

Once an attacker retrieves the code, decompilers could be the next stage, but under certain cases weak crypto implementations can be exploited by raw program data, for example if keys are stored on the program code (hardcoded), randomness tests can be executed to find areas with high entropy, and keys can be recovered (see Figure 3-5) [77]. This attack relies on the fact that machine code often uses structured data patterns vs a typical random pattern found on keys, particularly those of asymmetric nature, to prevent this types of attacks keys should be stored in a obfuscated manner (see section), or stored in secure areas of the chip if available.

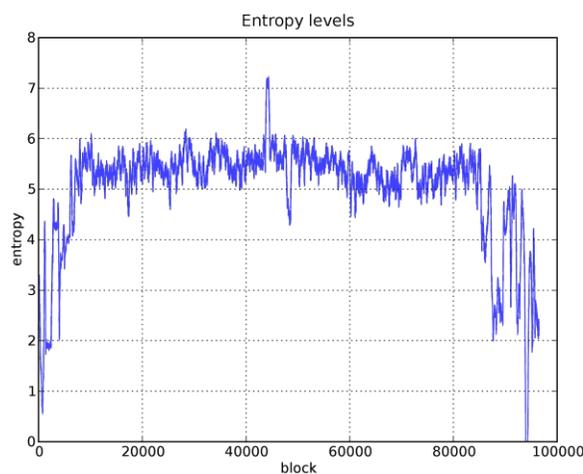


Figure 3-5 Entropy of a sample program, containing an asymmetric key located at the peak entropy point, taken from [77].

Software attacks can compromise a large part of the network if access keys are shared among a large number of devices, thus ideally each device should have its own key to prevent wide area attacks (such as in smart meters) [77]. Over the air (OTA) firmware updates are particularly vulnerable to serve as an attack access point since they can be used to extract and alter the program code, firmware updates can be obtained by direct server access or by hardware sniffing techniques (see section 3.5.2.2), to prevent this types of attacks firmware updates should be encrypted and validated before acceptance.

### 3.5.2.2 *Hardware attacks*

Embedded devices often consist of a set of computing units and slave devices that allow data visualization, storage and communication. These devices often communicate through standardized protocols that operate without security (e.g. UART, SPI, and I2C) which makes them vulnerable to eavesdropping, eavesdropping occurs in hardware through hardware sniffers, such as the “bus pirate” open-hardware project.

On certain devices that use of external communication radio modules (Wi-Fi, ZigBee, etc.) data can be monitored by installing jumper cables in the communication PCB traces as illustrated in Figure 3-6, these tools can be used to recover network credentials, or to inject data into the network through the use of sniffer computer terminal tools (see Figure 3-7). To mitigate these types of attacks, the use of Systems on Chip (SoC) solutions are recommended, in cases where this is not possible network attacks and security should be handled entirely by the microcontroller, avoiding off the chip hardware crypto accelerators. In some cases, it is also wise to implement data wipes upon tamper detection, as well as epoxy encapsulation of PCBs [77].

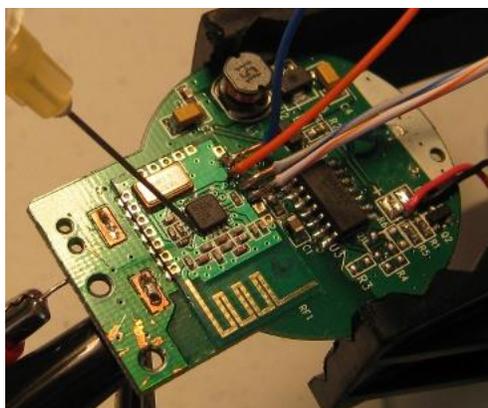


Figure 3-6 Communications sniffing, taken from [77]

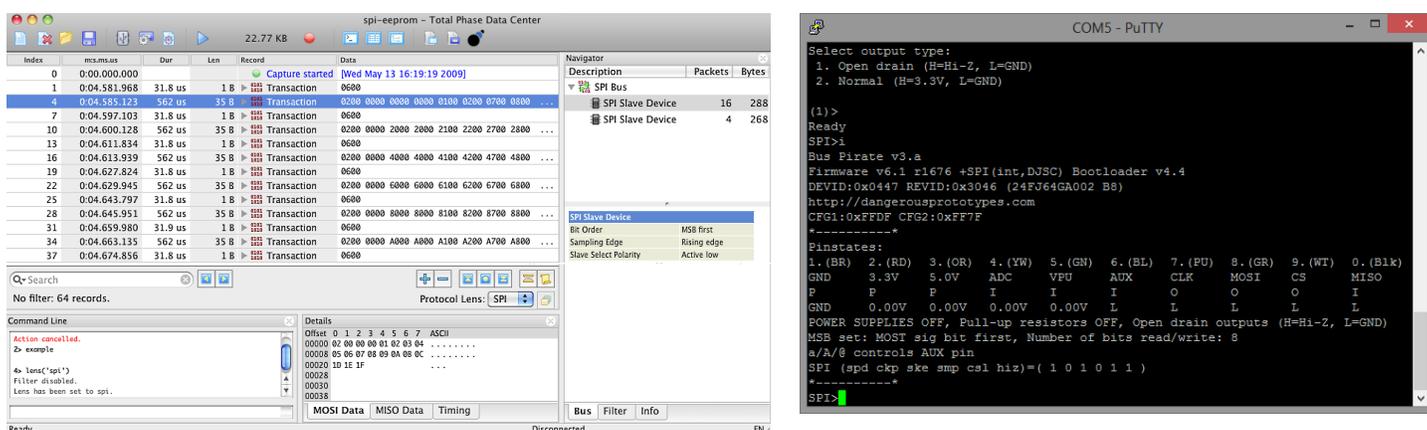


Figure 3-7 Professional vs open source sniffer visualization tools

Hardware attacks can be further exploited thru other techniques that are explained below:

**Component Impersonation:** Creating fake devices to record communications, or to instruct a component to execute certain task.

**Glitching attacks:** Creating timing/voltage variations to cause firmware errors that can cause false recordings, or give too much information about the undelaying processes.

**Connection flooding:** Creating high connection requests to create data overflows, hang states or denial of service.

**Chemical etching:** On some cases, chemical etching is required to access the inner protection mechanism to enable code extraction.

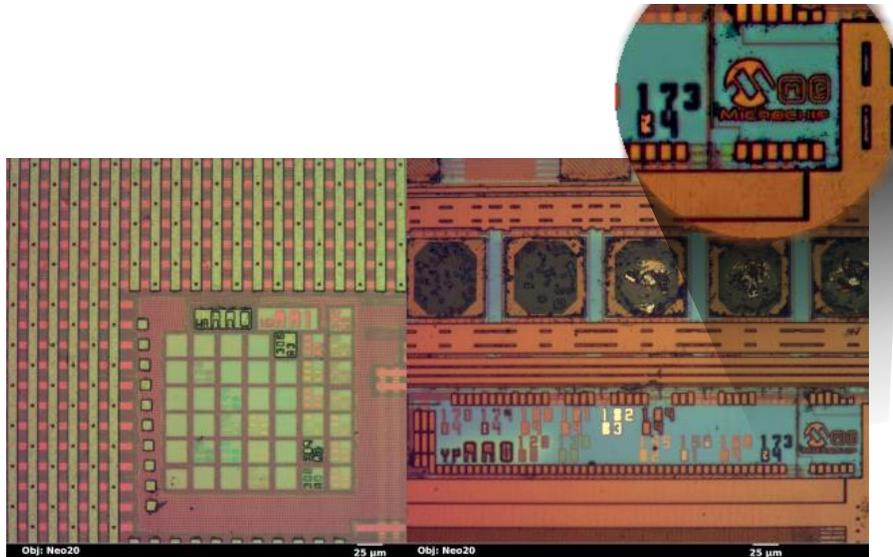


Figure 3-8 Chemical etching to expose silicon die, on a PIC32MZ (left) and PIC32MX (right) microcontroller, adapted from [78]

### 3.6 Smart Meter Security

Most people associate network attacks to computer viruses that hijack computing resources, for a third party benefit. Most attacks only compromise digital data and most damages are located on the virtual side. Although financial consequences are common, according to the US, a network attack can be defined as “Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself” [79].

In 2010 “*Stuxnet*” worm became widely known to the Information Technology (IT) world. This computer malware targeted specific hardware devices, specifically Siemens™ controllers. It relied on Windows™ as the carrier system, using external data storage devices and Local Area Networks as its spreading technique. This worm executed a fingerprint identification algorithm designed to detect an Iranian nuclear plant controller, when the target system was found it injected a custom firmware into the device controller. This cyber-attack caused damages to a physical system, in this case a uranium enrichment plant [76]. “*Stuxnet*” based attacks can be seen as the worst case scenario of a cyber-attack, these types of attacks can result in physical consequences. Similar attacks can be possible for the smart grid due to their widespread use and multiple points of entry that can create vulnerabilities that could be exploited by third parties [9].

Third parties can launch attacks for personal interests (reducing the energy bill), small area pranks, and large-scale attacks as an act of terrorism, or warfare technology. In Table 3.1 a list of possible outcomes is represented regarding security vulnerabilities.

Table 3.1 Cyber-Physical attacks scenario for smart grids, adapted from [9] .

<i>Attack \ Consequence</i>	<i>Cyber</i>	<i>Physical</i>
<i>Cyber</i>	Eavesdropping of private information	Stuxnet attack types
<i>Physical</i>	meter bypassing	Instability due to physical damage

Smart grid security has been a subject of interest for various market players and regulators. NIST has suggested the use of security measures, which should be resistant against data interception, information tampering, and prevent unauthorized access [15] [80]. NIST has also assumed that devices will be compromised at some point in time (network attacks), and a method for attacked meters containment and eventual recovery must be devised [80].

In [9] authors describe the requirements for a secure smart grid deployment, expose some associated risks, and provide ways to prevent the most basic attacks, giving an specific emphasis on cyber-physical security, cyber security encompasses three main axes [15].

**Confidentiality:** Smart meters allow detailed recording of energy consumption, using the concept of energy fingerprints. Analysts can determine the presence of cyclic loads, define types of loads installed and under certain cases, determine specific device operating states [81]. This information can be ideally used to assess power quality indicators or related studies by the utility personnel. Unauthorized parties with access to this information could use consumption data to infer financial capabilities, living patterns, and compromise consumer privacy. In figure 3-9, a minute resolution demand curve is plotted. The plotted curve can be used to visually identify various apparatus operating in the consumer premises, as well as determine specific operating statuses. This type of information can be valuable for the utility to determine user mean consumption, peak consumptions, or perhaps a daily load factor, but it can give too much information to a third party.

According to NIST, data must be only accessible by the user and the required entities (required business parts). Information gathered by the smart meters, should be treated as confidential and

appropriate policies should be laid and enforced in order to prevent unauthorized use of data. Also only truly required data should be collected or stored, even if monitoring systems are able to provide more services. Above these requirements, data requests should not be anonymous and must not be traceable [82].

It is important to note that the utility must previously define what type of data is confidential and what information can be publicly available, such as tariffs, audits, commands sent to the meter or firmware updates. Market players also need to understand that confidentiality is based on the use of access credentials and not in the obscurity of the infrastructure [64].

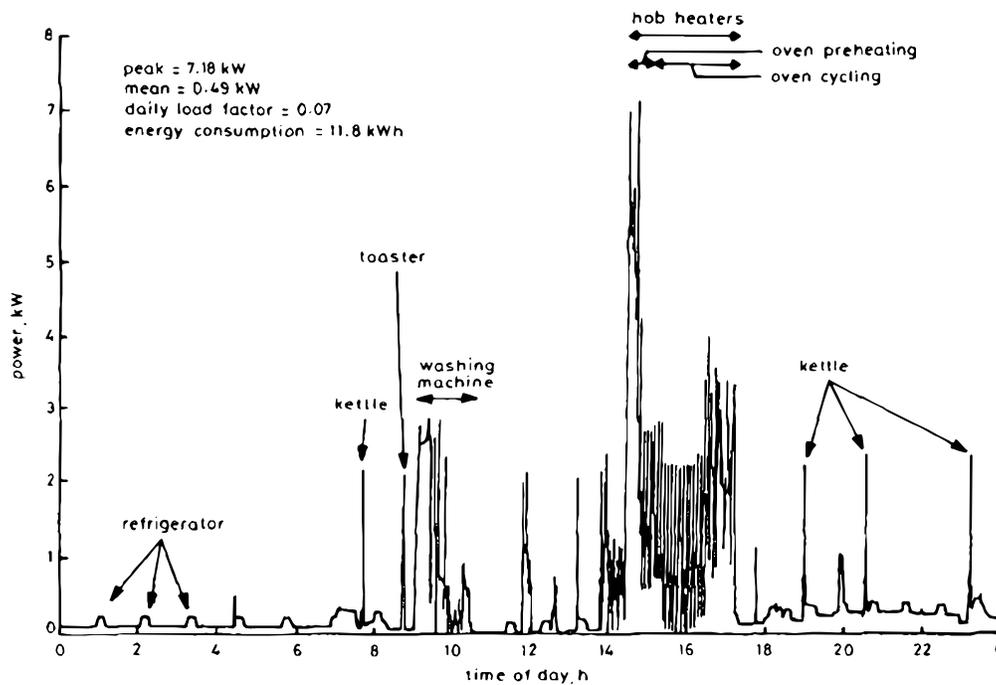


Figure 3-9 Example of load identification, from a detailed demand profile for a single user, taken from [82], [83]

**Data Integrity:** There must be a +54scheme to detect data modification that has occurred during the data traveling path; reported commands and measurements must be validated before storing or being executed, data must be validated for authenticity (Message came from where it says it comes), integrity (Message contains the original intended data).

Integrity can prevent attackers from issuing fake commands, such as remote disconnections, fake time of use pricing, but more importantly it can prevent custom firmware injection into field

devices. Custom firmware deployments could contain malware or could allow complete grid control by a third party.

**Distributed Denial of Service (DDoS) attack resistance:** a Distributed Denial-of-service attack seeks to congest a network with fake requests by using dummy devices, commonly known as distributed attacks. An over congested network causes targeted devices to reject some portion of overall requests due to finite computing capacities. Rejected requests can include legitimate ones, causing the so call Denial of Service. A DoS attack pursues to temporally suspend or render unreachable a service to its intended users. This service can be provided in terms of the smart grid by data server (utility data management system), data concentrator (data aggregator) or an end user device (smart meter).

Some vulnerable points of “smart grid” enabled networks, in order of criticalness are: dynamic pricing servers, remote command servers responsible for load management, service reconnections/suspension servers, and auditing services. Individual smart meters are also vulnerable but represent non-critical components, since consumption data can be read manually through an optical port, or in most cases, a manual inspection can resolve unreachability [9].

### 3.7 Proposal for securing key storage in microcontrollers.

In cryptography, there is always the risk of an attack, in most cases the security breach will occur at weakest point, bypassing other security checkpoints. Several authors try to warn the risks associated with cryptography, one of this quotes says, “the system security is only as strong as its weakest link” [84], meaning security must be viewed as a system, and all parts must implement equal security measures to prevent attacks. A common example of security breaches is on the corporate world, where is common to see state of the art firewalls, password protected servers, etc. and a *post-it note* with the user credentials on the desktop monitor, thereby lowering the entire security level.

#### 3.7.1 Key management

An important aspect of cryptography is the key lifecycle, composed of generation, transfer, storage and final disposal, proper key lifecycle steps guarantee that data systems remains safe(from the key access point), the problem is that keys are hard to store, mainly because a determined attacker could gain access via software holes, physical tampering, or reverse engineering. In recent years there has

been an interest in Physical Unclonable Functions (PUF), which use a physical pattern to generate a unique binary sequence, this pattern should be hard to predict, and impossible to duplicate [68], PUF's can be seen as static TRNG.

### 3.7.2 *PUF in Microcontrollers*

Physical Unclonable Functions in electronic circuits (i.e. microcontrollers) are due to variations in their manufacturing process, mainly in the transistor layout, which is subject to variations in geometries, deposition height, and silicon doping. These variations cause different power characteristics that can be exploited to obtain a random pattern, this random pattern could be used as a key, or can be further processed to obtain a set of Physical Obfuscated Keys (POK) [85].

In this section, the author describes a procedure for obtaining unique IDs from the SRAM unit found on most microcontrollers, mainly targeted to the PIC32MZ architecture fully described on chapter 6. In the next sections an overall introduction to PUF function is given, while in section 3.7.4 the experimental development of the proposed PUF extraction function is described.

Although several approaches to generate PUFs from microcontrollers have been proposed, there is a tendency to use Static Random Access Memory (SRAM) units to generate PUF data, SRAM units serve as main memory units for storing data during program execution, such as variables and program states. SRAM units are physically implemented by grouping a set transistors into a latch system, called the cell memory structure (see Figure 3-10), these memory cells tend to initialize themselves during the power-up sequence to random data, which depends, on temperature, previously stored data, ambient noise, and other phenomena. But these cells exhibit certain tendency to create patterns that depend on the manufacturing variability, thus the goal for a SRAM PUF extraction function is to extract the useful information (unique ID), in a repeatable manner from a SRAM unit.

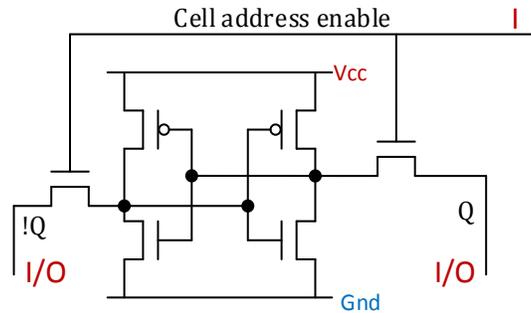


Figure 3-10 Typical T6 SRAM layout, that uses balanced pair communications to improve noise characteristics.

### 3.7.3 PUF extraction function

These SRAM PUF extraction functions must have filtering mechanisms that enable a program to generate the same ID under different conditions such as temperature variations, and previous operating states, while at the same time being capable of generating unique ID's between similar silicon batches, or cases where theoretically two devices should behave equally.

The extraction function must be designed according to the chip technology and should be robust against temperature and voltage variations as well as device aging, and thus must implement mechanisms that minimize ID recovery errors; some techniques involved in the design of PUF extraction functions involve use of repetition codes, error-correcting codes and checksums. The generated ID must also satisfy the properties of a TRNG to some extent, and must generate the unique ID, with the least possible amount of hardware resources [67].

Some properties that are likeable on a PUF extracting function are:

- ID's extracted by the PUF between different devices should be unpredictable (*hamming distance*  $\approx$  50%) [67], [68]
- Each PUF chip has to provide always the same output [68]
- The source of entropy should be easy to decode (digital values) [67]
- Without physical access to source of entropy, it should be computationally impossible for an attacker to generate the value [86].

Among the most interesting properties of a PUF function are those related to TRNG, specifically the hamming distance, which is measure of the distance between two strings. It indicates what percent

of information of the first string must be changed to achieve the value of the second string [87]. For binary sequences, Eq. 3.4 gives the hamming distance for two equal length streams

$$HD_{AB} = \frac{\text{number of \{ "1" | (A \oplus B) \}}}{\text{size (A \oplus B)}} * 100 \quad \text{Eq. 3.4}$$

Ideally, the hamming distance of the generated PUF value between two chips should be 50%, but also a set of TRNG properties should be fulfilled, although depending on the size of the bit sequence the result might vary from the ideal distributions. These tests are described on detail on [88], a short description of two of these tests are given in the next paragraphs:

**The Chi-Square test:** The Chi-Square test is used by many randomness tests to determine how much the observed data varies from the theoretical value, and is often expressed in degrees of freedom (*df*), and can be calculated according to Eq. 3.5 [89].

$$D = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i} \quad \text{Eq. 3.5}$$

where:

$$\begin{aligned} k &= \text{number of groups/bins} \\ o_i &= \text{Observed frequency of the } i^{\text{th}} \text{ group} \\ e_i &= \text{Expected frequency of the } i^{\text{th}} \text{ group} \end{aligned}$$

**Equidistribution test:** The bit stream is split into n-bit fields, and put into  $2^n$  bins according to their value, calculating its probability at the end of the process, the test determines that elements are uniformly distributed in each category according to the Chi-Square distribution.

### 3.7.3.1 Error repetition codes

One of the simplest error checking mechanisms are the error repetition codes; this type of error checking mechanism is often used on communications, and is based on the data redundancy principle. In the data redundancy principle, each single bit/byte value is repeated a number of times during transmission (.i.e. repeated) improving single burst error resistance, some advanced

techniques like FEC (see section I.1.1) distribute the bit contents over a large data space that improves multiple/long burst error resistance.

In [68] the authors propose to use repetition codes to extract unique ID values through a PUF function based on the SRAM unit. In this case, the repetition codes are created to mask the bit values contained on a byte, by considering the first bit value of a byte as the overall byte value, and generating a unique error correction code per byte that creates a majority decision, via XORing processes, this can be better understood by the example described in Table 3.2.

Table 3.2. Error correcting code applied to a PUF extracting function, as described by [68]

Error repetition generation process	SRAM startup value 1 (random)	SRAM startup value 2 (random)
Byte(x) power up value: 0b1001011	0b1001011 $\oplus$ 0b0110100=0b1111111	0b1101111 $\oplus$ 0b0110100=0b1011011
Mask based on the first bit: 0b1111111	Majority decision: 1 (7 set bits)	Majority decision: 1 (5 set bits)
Error correction bits (stored on SRAM for future use): 0b0110100		

### 3.7.4 Experimental development of the PUF function.

In this section, an experimental development of the proposed PUF generation function is described. In order to develop a PUF algorithm the target device must be tested for feasibility, i.e. finding areas of the chip that are able to provide a program with random data, in this case the test subject is the PIC32MZ device. This microcontroller uses a T6 SRAM layout that has been previously identified as a good source of entropy, according to [86], an internal image of the SRAM layout is given in Figure 3-11 (where the typical T6 layout can be observed).

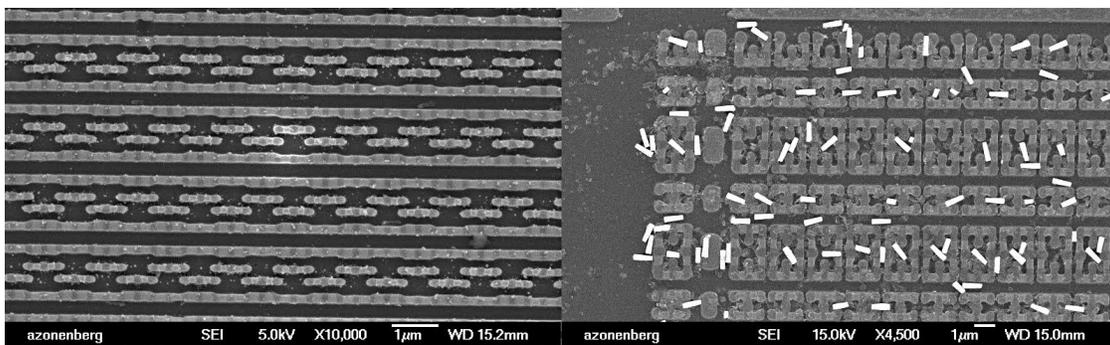


Figure 3-11 Internal silicon layout for the PIC32MZ series device (left) and PIC32MX (right), showing the SRAM layout, known as T6-doughnut, taken from [78]

In order to test the extracting function and its TRNG characteristics, under critical conditions two identical devices were chosen, these devices were manufactured at the same time (i.e. same batch), and thus only differ on a single digit that identifies them (see Table 3.3). Creating an interesting circumstance for the PUF extraction algorithm, since in theory, the devices were created under the same environment and they should behave similarly.

Table 3.3. Physical microcontroller identifiers for the devices used during the experimental phase of the PUF extraction function development.

	<b>DATE CODE</b>	<b>Serial Number</b>
Device 0	13413TD	EAS1A3STDL
Device 1	13413TC	EAS1A3STDL

The methodology used to capture the RAM state is the one described by Figure 3-12, where readings were taken at different temperatures to create signal fluctuations on the silicon components, these variability are due to the temperature effect on the forward voltage characteristics of diodes, used to form the transistors employed on the memory cell. The obtained readings for each device are partially shown in Table 3.4 (device 0) and Table 3.5 (device 1); these partial readings only contain 32 bytes of information, while the actual buffer size employed to generate a unique 128-bit key uses a total of 1 Kb and it is regarded as the large window on the following sections.

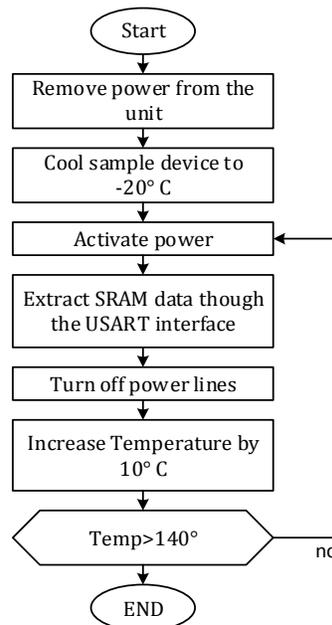


Figure 3-12 Methodology used to extract data from the PIC32MZ unit

The microcontrollers were loaded with a modified initialization code that disables automatic initialization of the memory segment (zeroing). Furthermore, to prevent ghost memory-effects on the ram unit a segment of this memory was reserved for system use, and no process is allowed to store data into the address range used to extract the PUF data, these simplifies the PUF function testing procedure.

Table 3.4. Sample bit sequence extracted from a PIC32MZ device using a custom firmware, with temperature variations. Device #0

Temperature	Memory contents (HEX)
-20	02EEFC0E62E9B0D8D52F1C44A36193E6F1676350CF17F3DB510E1447D514D38F
-10	02EEFC2660E9B0D8D52F1E45A36193EEE12367C0CE17F39B510E1447D114D38F
0	02EEFC0E62E9B0DBD52F1C45A34113EEE12763D1CF15F39B530E1447D114D38F
10	02FCFCB662F9B0D8D52F1D44A34193EEF12761C0C417F39B510E1447D114D38F
20	02CAFCE0E62E9B4DFD52F1C45A36193E6E12761C1C717F39B510E1447F514D38F
30	02DCFC2E62F9B0D8D52F1C45A34193EEF12741C0C715F39B511C1447D114D38F
40	02EEFC2E60F9B0DCD52F1C44A34193EEE123E1C0C515F31B511E1447D114D38F
50	02CCFC0660E9B0DBD52F1E45A34193EEE12363C0CF17F3DB511E1447D114D38F
60	22ECFC1662E9B0D8D52F1E44A34193EEF12761C0C717F3DB500E1447D014D38F
70	02ECFC1E60E9B0D9D56F1E45A36193E6E1676354CE15F39B510E1447F514D38F
80	02EFFC0662EDB0DBD52F1C45A34193EEE12763C0CD15F3DB511E1407D114938F
90	02CEFC8660E9B0D8D52F1C44A34193CEF12361E0C515F3DB510E1447D114938F
100	02CEFC0E62E9B0D8D56F1C44A36193E6E16763C0C515F35B530E1447D114D38F
110	02FEFC0E60E9B0DAD56F1D55A34193E6E12763F4C715F39B510E1447F114D38F
120	02CFFC1E60F9B0DAD56F1E45A36193EEE12763C0C417F39B510E1447D014D38F
130	02CEFC0662F9B0DED52F1C45A36193EEE12361C0C615F39B510E5447D114D38F

Table 3.5. Sample bit sequence extracted from a PIC32MZ device using a custom firmware, with temperature variations. Device #1

Temperature	Memory contents (HEX)
-20	E02E8CBF4A755CF8CED1F390B64AF8D8E9A32073510B60C644550D255AA344CC
-10	E0268CBF4A65DCF8CFD1F190B64AF8D8EBA32073510B60C244550D2558A344C8
0	D02E8CBF4A655CF8CED1F190B65BF8D8EBA22052500B60C24C450D2558A346CC
10	E02E8CBF4A655CF8CED1F390B64AF8D8FBA32052500B60C446551D255AA356C8
20	E02E8CBF4A755CF8CED1F390B65AF8D8FBA32072510B60C244551D2558A346C8
30	F02E8CBF4A655CF8CED9F390364AF8D8F9A22052500B60C04C551D2558A354C8
40	F06E8CBF4A655CF8CED9F390364AF8D8FBA32072500960C440451D255AA354DC
50	C02E8CBF4A655CD8CED1F390B65AF8D8FBA32072500B60C044551D2558A344C8
60	F02E8CBF4A655CF8CED1F390364BF8D8F9A32052510960C044551D2558A344C8
70	F02E8CBF4A655CF8CED1F390B65AF8D8F9A32052500B60C444550D255AA344C8
80	E02E8CBF4A65DCF8CED9F390B65AF8D8EBA32052500960C44C551D255AA356C8
90	F02E8CBF4A655C78CED9F390364AF8D8F9A32052510B60C444551D2558A344C8
100	D0269CBF4A655CF8CED9F390B65AF8D8EBA22072500960C0425D1D2558A354C8
110	90268CBF4A655CF8CED9F390B65BF8D8FBA32072500B60C0085D1DA558A354C8
120	D02E8CAF4A655CF8C6D9F390B65BF8D8F9A32052500B60C404559DA55CA354D8
130	D02E8CBF4A655CF8CED9F290965AF8D8F9A22072500B60C442551DA55CA354DC

Once the devices bit stream were captured, a preliminary test was done to assess the randomness of the bit streams; the results are shown in Table 3.6, where the presence of set bits ('1') is close to 50% on both devices.

Table 3.6. Set bits (0b1) distribution for each sample device

	Small window (32 bytes)	Large window (1024 bytes)
Device 0	50.3906	50.6836
Device 1	47.2656	48.9380

Simultaneously the Hamiltonian distance was computed, for both devices, using the stream readings obtained at -20° C, obtaining a Hamiltonian distance of 48.43%, the obtained XORed stream value was 0xE2C070B1289CEC201BFEEFD4152B6B3E18C443239E1C931D155B19628FB79743 for the small window sample, and was further used to evaluate the equidistribution test, with the results shown in Table 3.7

Table 3.7. Equidistribution test for the XORed string of the small window sample.

Value	% of occurrences	Value	% of occurrences	Value	% of occurrences
0b0	51.76	0b00	29.13	0b000	15.29
0b1	48.23	0b01	22.05	0b001	14.12
		0b10	23.62	0b010	11.76
		0b11	25.20	0b011	9.41
				0b100	10.59
				0b101	15.29
				0b110	11.76
				0b111	11.76

#### 3.7.4.1 Proposed PUF extraction algorithm

With the readings obtained in Table 3.4, a “set bit” count was obtained for the first 7 bits of each byte. Once the bit count was determined for each memory sample (temperature dependent) these were compared, by their byte positions to other samples, creating a comparison vector that contains the minimum, average and maximum “set bit” count for each byte, these are resumed in Figure 3-13 for device #0. Afterwards a repetition code was established using the most stable bytes, which were selected according to their minimum and maximum values with the help of a boundary condition that establishes the byte value, the results can be seen in Figure 3-13, where the usable bytes are highlighted in red to indicate a ‘1’ and blue to indicate a ‘0’.

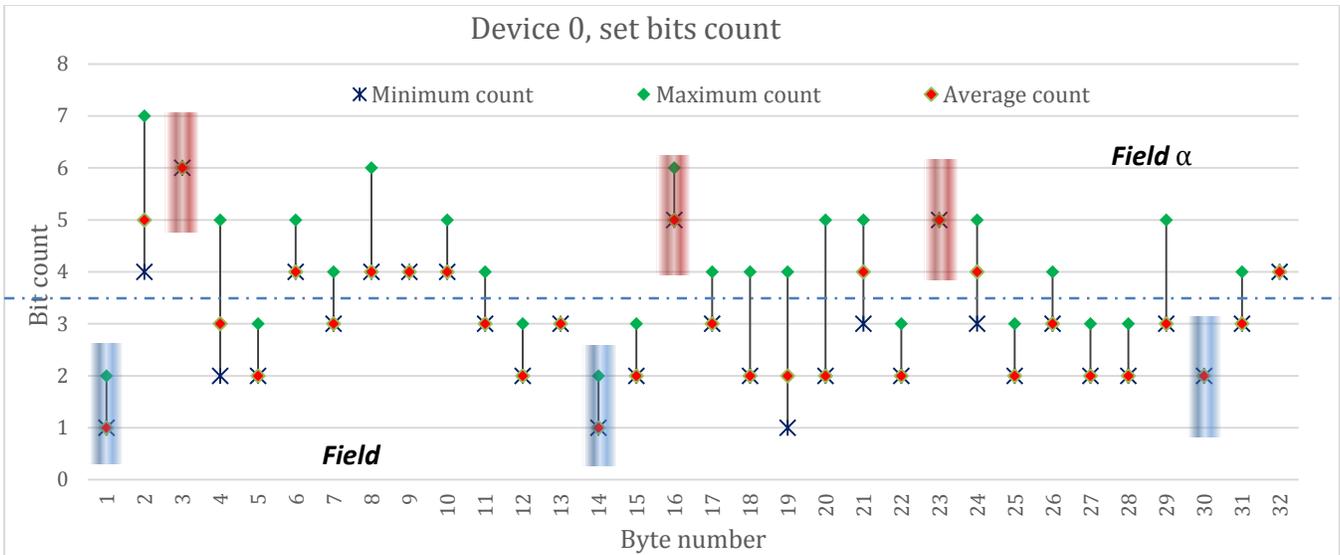


Figure 3-13 Bit set count for each byte, in device 0

The usability of each bit was determined by the equation shown on Eq. 3.6, where a tolerance error  $\epsilon$  is inserted to consider possible bit toggles not seen on the bit stream capture, due to the limited sample size. This equation enables the creation of a unique table containing suitable byte positions, that can be stored on the microcontroller, so that the device can compute its ID.

$$\begin{aligned}
 & \text{if } \{[Max(B_i) + \epsilon] \in \alpha\} \wedge \{[Min(B_i) - \epsilon] \in \alpha\} \text{ then value} = 1 \\
 & \text{if } \{[Max(B_i) + \epsilon] \in \beta\} \wedge \{[Min(B_i) - \epsilon] \in \beta\} \text{ then value} = 0
 \end{aligned}
 \tag{Eq. 3.6}$$

where:

$$\begin{aligned}
 & Max(B_i) = \text{maximum set bits in byte } i \\
 & Min(B_i) = \text{mimum set bits in byte } i \\
 & \alpha = \text{Field denoting a value of 1; its lower bound is set to } 7/2 \\
 & \beta = \text{Field denoting a value of 0; its upper bound is set to } 7/2 \\
 & \epsilon = \text{Added tolerance error, set to 1 in this case}
 \end{aligned}$$

The extracted bit values and positions, for this particular device are given in Table 3.8. An identical procedure was done for device #1, generating the bit pattern shown in Figure 3-14, with its corresponding bit position and values shown in Table 3.9 (from the small window set).

Table 3.8. Extracted bit values for device 0

Bit	Byte position	Value
0	1	0
1	3	1
2	14	0
3	16	1
4	23	1
5	30	0

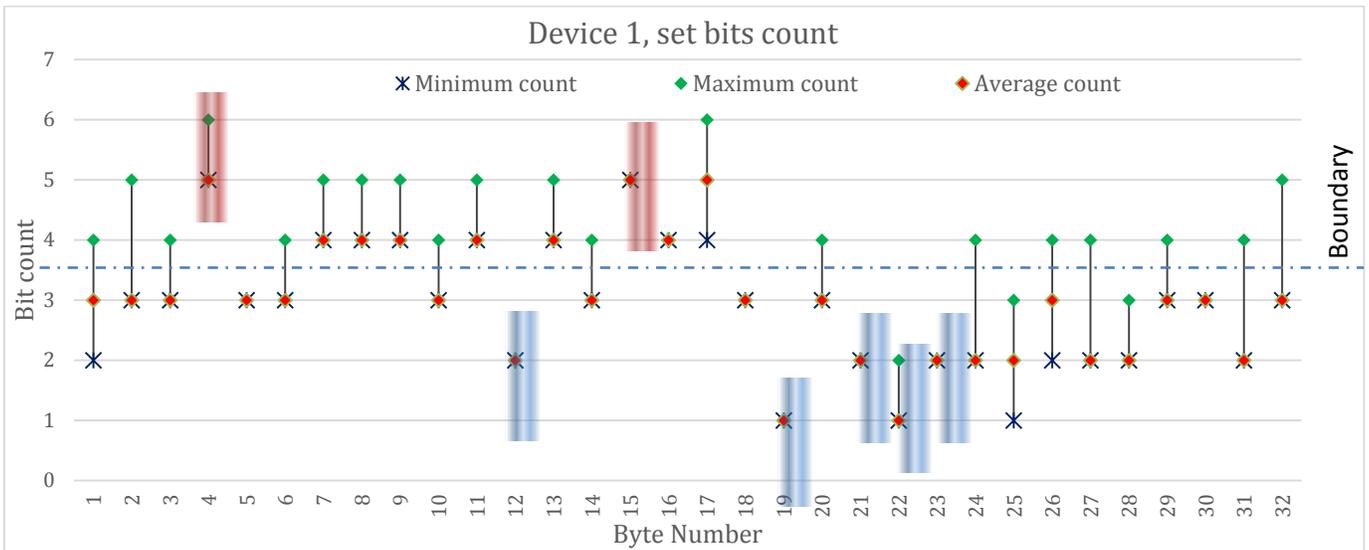


Figure 3-14 Bit set count for each byte, in device 1

Table 3.9. Extracted bit values for device 1

Bit	Byte position	Value
0	4	1
1	12	0
2	15	1
3	19	0
4	21	0
5	22	0
6	23	0

### 3.7.4.2 Results evaluation

The previously described methodology was expanded to process the entire sample window, consisting of 1024 bytes. In Table 3.10 the byte position and computed value is shown for the first 128 usable bytes of device #0, while in Table 3.11 the results are shown for device #1.

Table 3.10. Complete bit sequence extracted from device #0 (0x58FAF7E6881EDEAFFAC2A7CF0259941)

Bit	Byte position	Value									
0	1	0	32	129	1	64	292	1	96	495	0
1	3	1	33	130	0	65	296	1	97	503	0
2	14	0	34	133	0	66	299	1	98	509	0
3	16	1	35	141	0	67	308	1	99	510	0
4	23	1	36	148	1	68	311	1	100	511	0
5	30	0	37	149	0	69	317	0	101	519	0
6	34	0	38	151	0	70	329	1	102	520	1
7	50	0	39	165	0	71	340	0	103	521	0
8	55	1	40	167	0	72	341	1	104	537	0
9	58	1	41	169	0	73	342	1	105	538	1
10	59	1	42	171	0	74	347	0	106	539	0

Table 3.10 Complete bit sequence extracted from device #0 (Continued)

11	61	1		43	175	1	75	352	0	107	540	1
12	63	1		44	177	1	76	359	0	108	543	1
13	68	0		45	182	1	77	366	0	109	545	0
14	70	1		46	185	1	78	376	1	110	558	0
15	71	0		47	186	0	79	394	0	111	559	1
16	84	1		48	192	1	80	397	1	112	573	1
17	86	1		49	193	1	81	402	0	113	578	0
18	91	1		50	196	0	82	416	1	114	579	0
19	94	1		51	198	1	83	431	0	115	582	1
20	96	0		52	214	1	84	443	0	116	583	0
21	102	1		53	224	1	85	460	1	117	584	1
22	103	1		54	232	1	86	464	1	118	596	0
23	105	1		55	235	0	87	466	1	119	598	0
24	106	1		56	236	1	88	470	1	120	606	0
25	112	1		57	240	0	89	476	1	121	613	0
26	114	1		58	244	1	90	478	0	122	615	0
27	116	0		59	255	0	91	480	0	123	617	1
28	117	0		60	256	1	92	484	1	124	619	1
29	119	1		61	266	1	93	485	1	125	623	0
30	122	1		62	288	1	94	486	1	126	624	1
31	126	0		63	289	1	95	492	1	127	631	1

Table 3.11. Complete bit sequence extracted from device #1 (0xA054CBEF8AE074510716DEDAD41C543)

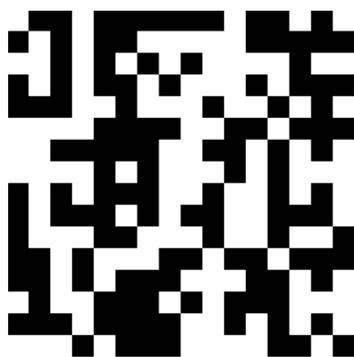
Bit	Byte position	Value									
0	4	1	32	130	1	64	316	0	96	465	1
1	12	0	33	138	0	65	319	0	97	466	1
2	15	1	34	139	0	66	349	0	98	468	0
3	19	0	35	145	0	67	355	0	99	473	1
4	21	0	36	160	1	68	359	0	100	476	0
5	22	0	37	168	0	69	366	1	101	485	1
6	23	0	38	180	1	70	367	1	102	486	0
7	34	0	39	188	0	71	374	1	103	490	0
8	35	0	40	189	1	72	375	0	104	494	0
9	41	1	41	191	1	73	379	0	105	495	0
10	44	0	42	193	1	74	388	0	106	500	0
11	45	1	43	200	0	75	393	1	107	503	1
12	52	0	44	204	0	76	396	0	108	512	1
13	53	1	45	211	0	77	397	1	109	513	1
14	55	0	46	213	0	78	398	1	110	534	0
15	57	0	47	216	0	79	407	0	111	553	0
16	59	1	48	219	0	80	409	1	112	557	0
17	62	1	49	225	1	81	413	1	113	575	1
18	65	0	50	228	1	82	421	0	114	580	0
19	72	0	51	256	1	83	431	1	115	594	1
20	77	1	52	263	0	84	432	1	116	604	0
21	82	0	53	268	1	85	433	1	117	609	1
22	92	1	54	271	0	86	439	1	118	628	0
23	96	1	55	278	0	87	441	0	119	631	0
24	97	1	56	279	0	88	445	1	120	632	0
25	102	1	57	280	1	89	448	1	121	638	0
26	114	1	58	285	0	90	449	0	122	641	1
27	116	0	59	291	1	91	450	1	123	643	1
28	122	1	60	311	0	92	451	1	124	644	0
29	123	1	61	312	0	93	456	0	125	647	0
30	127	1	62	314	0	94	462	1	126	665	0
31	128	1	63	315	1	95	464	0	127	672	1

Once the IDs were obtained from the sample devices additional tests were done to evaluate their randomness. In Table 3.12 the results of the equidistribution tests are presented, with  $df$  according to the Chi Square test. As it can be seen, the results are good for small sequences but can grow rapidly when the grouping bins have a low count, and thus should be ignored when analyzing short sequences.

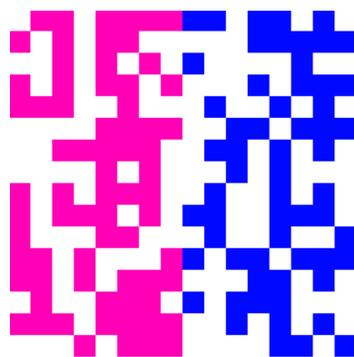
Table 3.12. Extracted IDs random properties, (Equidistribution test)

Device #0 ID						Device #1 ID					
Value	%	Value	%	Value	%	Value	%	Value	%	Value	%
0b0	44.19	0b00	17.76	0b000	9.86	0b0	52.40	0b00	28.85	0b000	18.84
0b1	55.81	0b01	28.97	0b001	11.27	0b1	47.60	0b01	24.04	0b001	11.59
D	1.531	0b10	24.30	0b010	14.08	D	.281	0b10	23.08	0b010	7.25
		0b11	28.97	0b011	14.08			0b11	24.04	0b011	5.80
		D	4.125	0b100	11.27			D	.968	0b100	11.59
				0b101	11.27					0b101	15.94
				0b110	5.63					0b110	18.84
				0b111	22.54					0b111	10.14
				D	17.437					D	17.81

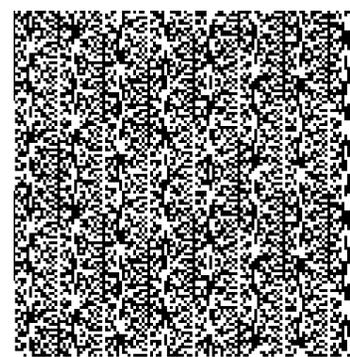
The human brain has great pattern recognition capabilities, often identifying patterns that are not discoverable using mathematical tests [90]. In Figure 3-15 the reader can evaluate by itself the quality of the random data by comparing the pattern generated by joining the two ID's vs the one produced by a PRNG.



Joint ID's represented by monochromatic bit representation



Joint ID's highlighting bits coming from device #0 (Magenta) and device #1 (blue)



A similar approach using a PRNG, with visible patterns, adapted from [90]

Figure 3-15 A visual representation of the generated pattern vs the one coming out of a PRNG.

Another property of random data is the low autocorrelation values generated by analyzing the sequence, in Figure 3-16 the autocorrelation plot for the ID obtained by the device #0 is shown, similarly the results for device #2 are shown in Figure 3-17.

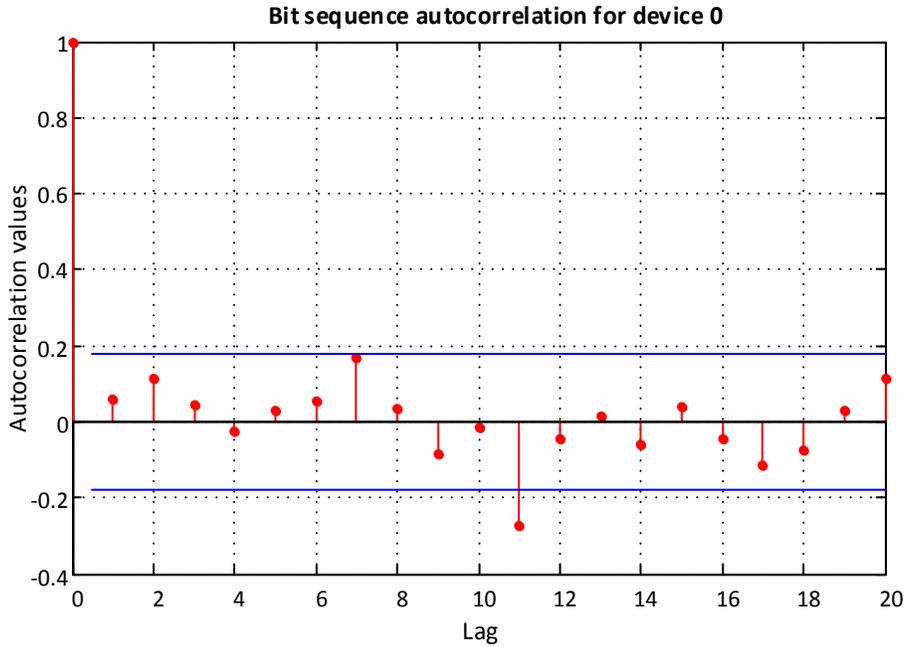


Figure 3-16 Autocorrelation function for device #0

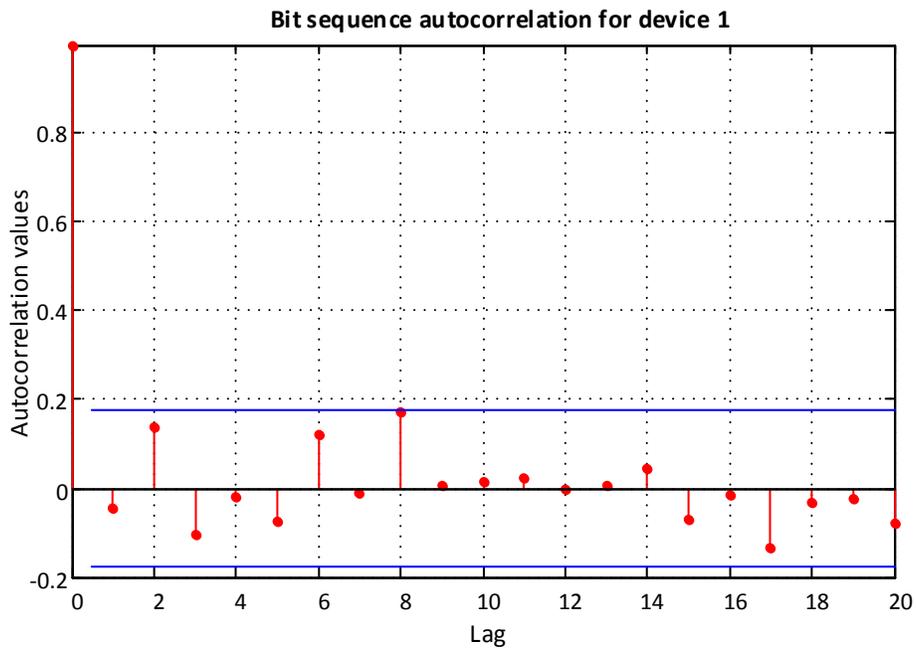


Figure 3-17 Autocorrelation function for device #1

To complete the PUF quality test the Hamiltonian string is generated by XORing the bit sequence obtained from both devices, an a equidistribution test is performed, obtaining good results (D is within acceptable range)

Table 3.13. Hamiltonian bit string equidistribution properties.

Value	% of occurrences	Value	% of occurrences	Value	% of occurrences
0b0	49.52	0b00	24.04	0b000	15.94
0b1*	50.48	0b01	27.88	0b001	13.04
D	.031	0b10	23.08	0b010	11.59
		0b11	25.00	0b011	7.25
		D	.656	0b100	10.14
				0b101	15.94
				0b110	7.25
				0b111	18.84
				D	12.815

\*this value is also known as the Hamilton distance

### 3.7.4.3 Further improvements on the PUF function

Although the preceding algorithm produces high quality identifiers (from the TRNG point) it does not offer a checking mechanism that enables the device to verify that generated ID is correct, and could be an important pitfall on production hardware if an unverified ID is used. To improve this, a known plaintext-cipher pair is used to verify the ID, in this case the ID is used as the key, and the checking mechanism works in a similar way to a tag verification mechanism of hash values (see Figure 3-18).

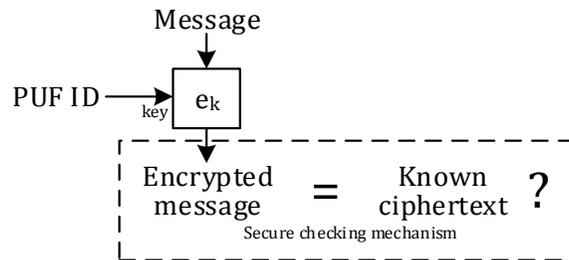


Figure 3-18 Proposed PUF verification method.

The proposed PUF validation algorithm forms part of the firmware implementation that will be described in section 6.5.9 of this thesis.

## CHAPTER 4

### 4. Digital Communications in Smart Metering Networks

#### 4.1 Introduction

Until now this work has mentioned the importance of security in telecommunications, without discussing the communication means. In a general sense, communications refer to the data exchange between a sender and a receiver through a channel medium by using a common protocol. In the digital world, this communication is commonly based on the use of network communications. Network communication refers to the telecommunications network that enables data exchange between a set of networking devices, called nodes. These nodes initiate, relay and receive data through a set of network links, these links are the equivalent to traditional channel mediums, and can be wire-based (e.g. Cat cable, Fiber Optics) or wireless (e.g. Wi-Fi, ZigBee, GSM). Above this physical links, protocols handle data, enabling data interchange between the participating members; some examples of commonly used protocols are TCP and IP [91].

In this chapter a bottom-up approach is given, firstly discussing low-level communications, and working its way up into secure network communications, in the first part wireless network interfaces will be discussed (Physical layer), moving to low level protocols (connection handling), and finally discussing the internet based communication networks used in a smart meter networks(TCP, IP, TLS).

In the last few years several smart grid networks were laid, using cellular networks, Wi-Max, and local area networks, and in some cases even fiber optics; although initially thought as a low rate, high latency network, applications relying on smart meter data soon required higher throughput that prior deployments were unable to handle, rendering them obsolete. Current acceptable ranges are in between 256-512kbps, but are being constantly being revised by standardization organizations to prevent obsolescence; organizations suggest using software upgradeable radio-modules or field upgradeable hardware .

At the end of the chapter a radio frequency (RF) transceiver unit is presented, this RF unit uses a custom-build circuitry exclusively designed for its use on the proposed smart meter. Although this chip offers many features like automatic hardware parsing and crypto accelerators, these features

remain unused and rather operations are handled via an external microcontroller to improve overall security and facilitate radio communications upgradeability via software. These types of radio handling characteristics are recommended by NIST on [9], and can be further read in section 6.1.

Although the used transceiver is only certified for IEEE 802.15.4-2006 compliance, a modification of physical layer was created that allows near IEEE 802.15.4g-2012 interoperability. On top of the physical layer, a proprietary mesh network was designed based on the requirements of the TCP/IP protocol, which enables smart meters units to communicate by using the TLS protocol.

## 4.2 **Wireless Communications**

Communications based on radiofrequency (RF) are possible due to the wave propagation mechanism described by Maxwell, which states that “a time-varying magnetic field acts as a source of electric field and a time-varying electric field acts as a source of a magnetic field” [92]. Maxwell equations propose that the Electric ( $\mathbb{E}$ ) and Magnetic ( $\mathbb{B}$ ) are auto sustainable, meaning that a signal can travel long distances until both of them are absorbed by the medium.

The wave propagation mechanism enables to use electromagnetic waves as a communication channel, by controlling the sent signals at the transmitter side, and decoding received signals at other end. Although RF would be theoretically possible at low frequencies, huge antennas would be needed, and thus practical RF applications operate in between the 30 kHz and 100 MHz range. There are several fundamentals related to RF, but perhaps the most basic rule is that the higher the frequency used, a higher channel capacity can be achieved at the cost of a lower transmission range. In order to separate the electromagnetic spectrum according to its applications (required data rates) the RF spectrum has been split onto several sub-spectrums, which are highly standardized across the world [93]

### 4.2.1 ***Fundamentals of RF***

The term “carrier frequency” refers to the center frequency used for transmitting data, this frequency has an associated wavelength that is usually given in meters, it is used to estimate the transmission range and establish appropriate antenna sizes. The wavelength of a signal is given by

Eq. 4.1 where  $v$  denotes the phase velocity of the medium and  $f$  the carrier frequency, for RF applications  $v$  can be considered equal to the speed of light  $c$ .

$$\lambda = v/f \quad \text{Eq. 4.1}$$

Since a carrier frequency only contains a sine wave, it would be difficult to transfer large quantities of information (by using on and off states). To enable data transmission some control over the RF channel is required, this control is done by means of modulation. Modulation can be defined as the process of embedding a data signal into a carrier signal, by alternating the carrier frequency characteristics, such as frequency, amplitude and phase according to the stream of data to be transmitted.

Once modulated the transmitted signal must then travel through the air (the channel medium) until it reaches a receiving station to be demodulated and be transformed back to the original signal. This process is illustrated by the communications diagram shown in Figure 4-1. In the next section a more on depth explanation of the RF communications components is given.



Figure 4-1 Basic communication diagram showing the main stages of wireless communications.

#### 4.2.1.1 Power measuring in RF

Electrical engineers often use multiples of watts to refer to the generated, transmitted and received power, because most of their transmitted power travels across lossless environments, but in the RF field, these losses can be huge, therefore communication engineers often work with logarithmic scales to ease power losses calculations. In the general sense, every time a signal is transmitted its amplitude (and thus its power) quickly diminishes as the receiver moves further away. These losses are better represented in a logarithmic scale, and most transmitter power ratings, receiver sensibilities, as well as antenna gains will be expressed in dB, in RF communications dB are often tied to a particular reference value, in this case to 1 mW (see Table 4.1).

Table 4.1. Sample dB scale tied to 1mW of power.

dB power	mW
-40dB	.0001
-20dB	.01
0dB	1
20dB	100
40dB	10000

There are other advantages of using the dB scale, in a general sense for every 6dB of power increase the transmitted distance doubles, and for every 3 dB increase in power the amplitude doubles [94].

#### 4.2.1.2 The channel

The channel is medium through which the electromagnetic waves travel, often considered as the empty space, and having adequate transmission responses. In practice, this channel introduces noise from other RF sources (other transmitters and reflections), causing signal losses as well as delays [95]. The channel can be considered as a hostile environment where most of the signal degradation occurs, and as such, the demodulation scheme should be designed to withstand the channel noise.

Another aspect of the channel is the loss of power (i.e. amplitude) of the transmitted signal; the most basic models consider the transmission distance and wavelength parameters to help determine the lost power, one of these models is given for the empty space by the Friis equation [93], shown on Eq. 4.2

$$P_r = P_t G_t G_r \left( \frac{\lambda}{4\pi r} \right)^2 \quad \text{Eq. 4.2}$$

where:

$$P_r = \text{Received power}$$

$$P_t = \text{Transmitted power}$$

$$G_t \text{ and } G_r = \text{Transmitter and receiver gains}$$

$$r = \text{distance between transmitter and receiver.}$$

#### 4.2.1.3 *The modulation*

Analog telegraph transmission was the first practical application of RF, demonstrated by Marconi in 1895, these analog transmissions were based on the spark-gap transmitter (impulse radio) that created two different length pulses with equal frequency to transmit telegraphic messages [93], and although it fulfilled its purpose, it did not allow signal transmission, as we know it today. Later on Reginald Fessenden created the first theories of modulation by using sinusoidal waveforms, and gave birth to the first modulation technique called Amplitude Modulation (AM).

### 4.3 **Issues with Respect Wireless Communications**

In the beginning communications were either designed to be point-to-point (telegraph) or used in broadcasting services (one transmitter, multiple clients) such as radio stations. However, in the modern digital era, every device can transmit and receive data through the same-shared channel medium, meaning that some form of medium control must be implemented to avoid data collisions or signal interference.

Traditionally RF medium control was done thru frequency allocation, i.e. allocating certain frequencies to a particular service (such as a TV channel or radio station), this type of regulatory control was possible thanks to relatively uncrowded space, and the existence of wide area services. This contrasts with modern digital communications that are often limited to operate on the unlicensed bands, and as such must compete for the medium access; since this medium is often bandwidth limited, several communication protocols use channel allocation to split the available frequency spectrum and enable simultaneous data communications.

Although the idea of channel allocation helps to organize traffic, these channels are often overcrowded, and some form of control is needed. Since a single channel-master supervisory control would be impractical, many protocols rely on self-managed access controls, including busy channel detection and collision avoidance. Some of these techniques are Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), and Clear Channel Assessment (CCA), and are further discussed below.

### 4.3.1 *Clear Channel Assessment*

The CCA determines the presence of an ongoing transmission by listening to the RF channel and evaluating the presence of a valid modulated signal by analyzing the airwaves for a determined time [94]. If the channel is empty during the entire assessment time, the channel is considered clear and data transmission can begin, this form of control is considered local, since only the transmitting node is used to assess if the channel is clear.

### 4.3.2 *The hidden node problem*

The CCA test fails to detect transmissions that are occurring at the receiver end, creating the hidden node problem, which means that a unit that wishes to transmit can only sense its immediate area, but cannot determine the receiver end status. This can be exemplified by Figure 4-2 where the node A cannot see the existence of node C and thus can falsely determine the channel status, whereas B can be affected by interference if A and C wrongly evaluate the channel and transmit at the same time.

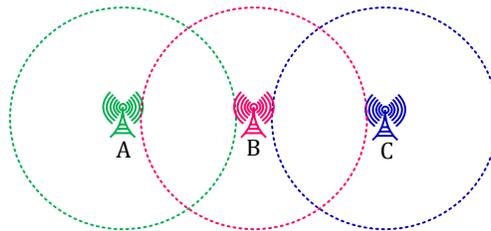


Figure 4-2. The hidden node problem, node B can see both A and C; but C and A units cannot see each other.

### 4.3.3 *Carrier Sense Multiple Access with Collision Avoidance*

Since CCA is vulnerable to the hidden node problem, CSMA/CA employs additional steps to ensure that both communication units are clear from interference, it does this by sending an additional clear to send request to the other node, which silences both sides of the medium during the transmission time. The CSMA/CA algorithm is better explained by using the UML diagram shown in Figure 4-3.

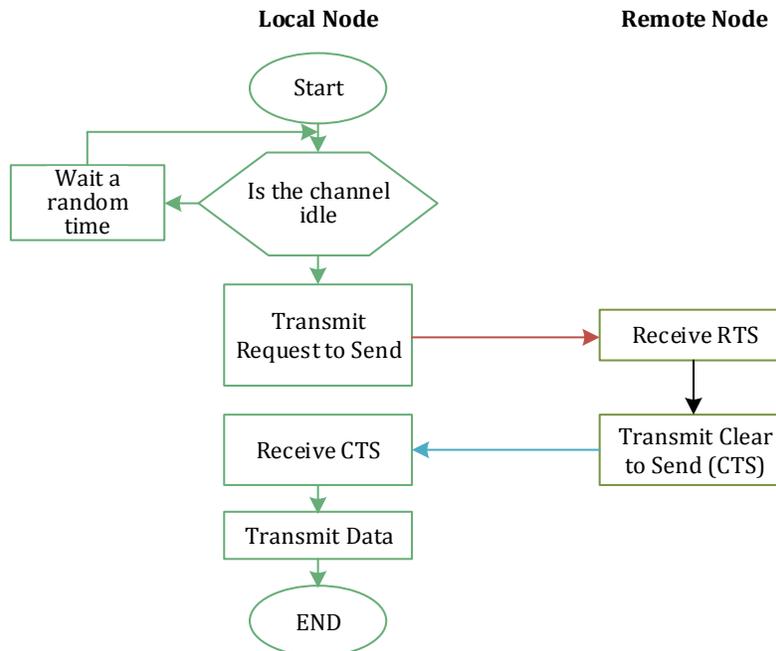


Figure 4-3 The CSMA/CA algorithm, Based on [10], [94].

#### 4.4 The Open Systems Interconnection model

In the early 1970's the International Organization for Standardization (ISO) proposed the Open Systems Interconnection (OSI) model to organize data communications, this model splits network communications into a set of seven layers according to specific activities [94]. On each layer a set of limited data processing is performed, creating data packages that are validated within the layer and that enable modularity, this type of modularity enables to support complex protocols such as HTTP thru the use of a protocol stack.

As mentioned earlier the OSI model consists of seven layers, thru which data moving from a local application must pass before reaching a remote client application, these layers although not always necessary, are part of the most protocol stacks such as HTTP (web surfing), FTP (file transferring), SMTP (email). In the next section an in-depth description of each layer will be given.

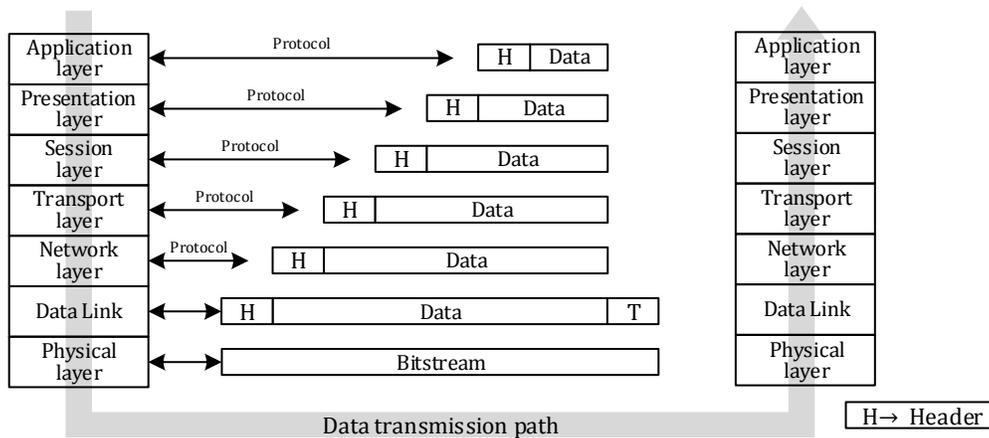


Figure 4-4 Data transmission path in the OSI reference model, adapted from [93].

### Physical layer

The physical layer (often shorted as PHY) is responsible for handling low-level data transmission thru a medium, creating and managing electrical signals in cable networks, and decoding RF communications on wireless networks [93]. An important aspect to note about layers is that each is responsible for validating the transmitted data by means of headers and tags, but in this case the bit level transmission is only checked for demodulation errors.

### Data link layer

The data link layer is responsible for validating raw bit transmissions, which could have been corrupted by the transmission medium; it also manages the data stream coming from higher levels (buffering data according to the channel capacity). This layer is often split in two parts, the Medium Access Control (MAC) and Logical Link Control (LLC) layer; the MAC is responsible the point to point addressing and channel sharing protocols (i.e. transmitting only when the channel is idle). The LLC is responsible for allowing full duplex communication and provides the unicast, multicast or broadcast capabilities. From this point on all layers, implement data checking mechanisms [93].

### Network layer

This layer is responsible for routing the data packages from end-to-end, i.e. finding the travel path that enables the shortest (and possibly fastest) data path.

## Transport layer

The transport layer is responsible for enabling reliable data streams between end-to-end devices, it does this by supporting package scheduling, flow control and packet recovery.

## Session Layer

This layer is responsible for managing device-to-device connections at the session level, i.e. permitting long communications between parties that relay in successive request and response messages, providing in some cases authentication, authorization and session recovery.

## Presentation Layer

The presentation layer is responsible for application data delivery and formatting of data, e.g. it allows decoding of data streams into correct image, audio and video files.

## Application Layer

It is responsible for the final data presentation to the user, e.g. web browsers, video streaming services and data monitoring applications.

### 4.5 The IEEE 802.1x Family of Standards

Almost all wireless network communications are based on the IEEE 802.1x family of standards, with perhaps the 802.11 being the most widely known, 802.11 is often identified by its commercial name “Wi-Fi”, it enables fast Wireless Local Area Network (WLAN) communications for computing devices. The 802.1x family of standards deals with the Physical and Link layers of network communications, some of these standards can be seen in Table 4.2.

Table 4.2. Part of the 802.1x family of standards, adapted from [96]

Standards Committee	IEEE 802: Local Area Networks (LAN)/Metropolitan Area Networks Standards (MAN) Committee				
Working group	IEEE 802.11 WLAN	IEEE 802.15 Wireless Personal Area Networks (WPAN)			
Task Group	802.11a/b/g/n	802.15.1 WPAN/Bluetooth	802.15.3a WPAN High Rate	802.15.4 Low Rate WPAN (LR-PAN)	802.15.4g LR-WPAN-SMUN
Promoter	Wi-Fi alliance	Bluetooth SIG	Wi-media	ZigBee Alliance	Smart Grid Projects

#### 4.5.1 *The IEEE 802.15 family of standards*

IEEE 802.15 regulates communications at the Wireless Personal Area Network (WPAN) level, this type of wireless communications enable data transfer between a local set of devices, with occasional higher network uplinks (i.e. internet). WPAN networks are divided between high speed (HR) and low speed (LR) networks, some examples of high-speed networks are Bluetooth and wireless USB, which require high data throughputs. On the contrary, low speed networks are intended for monitoring applications, such as home automation devices and industrial sensor networks.

Low speed devices are often required to have low power requirements and a low overall cost, thus require simple communication protocols that can be handled by power and cost restricted hardware [96], for this purpose the IEEE has established the IEEE 802.15.4 standard. This standard regulates LR-WPAN communications in a variety of scenarios and devices, including smart metering networks.

##### 4.5.1.1 *The IEEE 802.15.4 Standard*

The IEEE 802.15.4 standard encompasses a wide set of aspects, including frequency spectrum allocation and specific modulation schemes that are applicable to wide areas of the world, to summarize this information Table 4.3 enlists the wireless medium characteristics.

The 802.15.4 standard works as the core reference model for LR-WPANs but due to its restrictive protocol nature, many amendments have been developed to suit a variety of real life scenarios. One of these cases is IEEE 802.15.4g which widens the availability of modulation schemes raising overall transmission speed, and increasing the data payload size. The “IEEE 802.15.4g-Standard for local and metropolitan area networks: Low-Rate Wireless Personal Area Networks (LR-WPANs) for Low-Data-Rate, Wireless, Smart Metering Networks (SMUN)” published in 2012 is an amendment of 802.15.4 that establishes new PHY and MAC layer requirements for smart meter communications [10]. Due to its low involvement in the OSI model, it leaves the user with possibility of using upper layers to encapsulate complex protocols such as TCP/IP or the IEEE C22 family of smart metering standards.

The IEEE 802.15.4g amendment indirectly improves the network layer properties by increasing the data payload, creating transmission characteristics that are ideally suited for the smart meter industry. Enabling wider data management capabilities, such as the encapsulation of complete TCP packets, higher transmission speeds, and facilitating the incorporation of internet based protocols such as 6LoWPAN, without requiring proprietary protocols such as ZigBee, MiWi, or WirelessHART

Table 4.3. IEEE 802.15.4 Physical Layer characteristics, summary taken from [10], [96] [97]

Frequency Bands	Channel Number	Center Frequency (Hz)	Used Modulation schemes	Bit Rates (kb/s)	Availability
868 MHz	0	868.3	BPSK, ASK, O-QPSK	20, 100, ,250	Europe
915 MHz	1	906	BPSK, ASK, O-QPSK with Frequency-Hopping Spread Spectrum (FHSS) (Same concept as DSSS but chips alter frequency) [94]	40,250	United States
	2	908			
	3	910			
	4	912			
	5	914			
	6	916			
	7	918			
	8	920			
	9	922			
	10	924			
2.4 GHz	11	2405	O-QPSK with DSSS, FSK*, OFDM*	250, 500*	World wide
	12	2410			
	13	2415			
	14	2420			
	15	2425			
	16	2430			
	17	2435			
	18	2440			
	19	2445			
	20	2450			
	21	2455			
	22	2460			
	23	2465			
	24	2470			
25	2475				
26	2480				

\* Only available on 802.15.4g amendment.

Since the 2.4 GHz frequency band, also known as the 2.4 GHz “Industrial, Scientific and Medical” (ISM) band is the most prevalent unregulated electromagnetic spectrum in the world, the following sections will only consider it for the purposes of explaining the IEEE 802.15.4g.

### 4.5.2 The 2.4 GHz ISM band

The 2.4 GHz RF spectrum is an open, general use band that lacks licensing fees and as such attracts many equipment manufacturers. This semi-unregulated band makes it easier for them to bring wireless products to the consumer market; these products include wireless telephones, computers, and hands free devices. Each of these devices often communicate through different protocols simultaneously causing this band to be always busy, mostly by data-intensive traffic originated by 802.11, leaving 802.15.4 with the task to coexist by only using the available spectrum left over.

IEEE 802.11 uses a set of 14 (one of them reserved for military use) overlapping channels that are separated at 5 MHz intervals; where each of these channels has 22 MHz bandwidth (see Figure 4-5). Although all of these channels are specified on the standard, most Wi-Fi equipment manufacturers prefer to use the so-called non-overlapping channels, which are located at channels 1, 6 and 11 respectively, creating some spectrum holes that can be exploited by other devices.

On the other hand, IEEE 802.15.4 defines the availability of 16 non-overlapping channels that are separated at 5 MHz intervals, each of them having a 2 MHz bandwidth (see Figure 4-5), creating up to 16 simultaneous channels, although in practice only four are usable. This is because the standard operates at 2.4 GHz and in real life it must share resources with other data intensive applications running on 802.11, the four usable channels are located on the spectrum holes created by IEEE 802.11 thus making the channels 15, 20, 25 and 26 the preferred communication paths (see them highlighted in Figure 4-5).

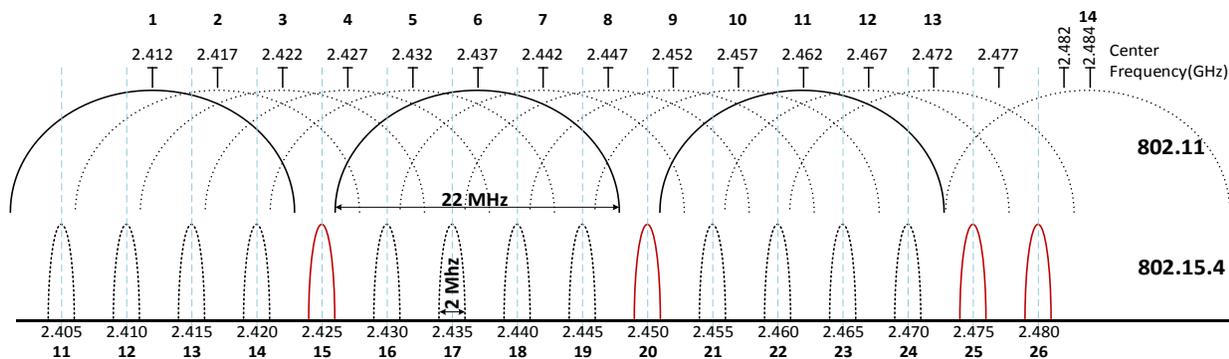


Figure 4-5 The IEEE 802.11 and 802.15.4 RF spectrum adapted from [98].

### 4.5.3 Physical layer properties

As mentioned in the introduction of IEEE 802.15.4g, the standard regulates the physical and link layers of the wireless channel; in a broad sense, the physical layer handles the raw bit stream, but does not possess robust error-checking capabilities, and consequently must rely on an encoding method that minimizes random errors due to the external noise. To do this the standard uses DSSS-based techniques to reduce the like hood of decoding a false bit, plus custom-made algorithms that add data redundancy.

At the 2.4 GHz ISM band the standard specifies a set of modulation schemes that enable various transmission speeds, among these modulations O-QPSK offers the highest transmission speed due to its high speed settling time. In Table 4.4 the requirements for transceivers (radio transmitters/receivers) that operate at 2.4 GHz is given, this table uses terms that will be described in detail in the following sections.

Table 4.4. IEEE 802.15.4g Physical Layer requirements for O-QPSK transceivers operating on 2.4 GHz, adapted from [10]

Frequency band	Chip rate (kchips/s)	Rate Mode	Requires BDE?	Spreading technique.	Requires FEC + Interleaver?	Data Rate (kb/s)
2.4 GHz	2000	0	Yes	(32,1) DSSS	yes	31.25
		1	No	(32,4) DSSS	yes	125
		2	No	(16,4) DSSS	yes	250
		3	No	(8,4) DSSS	yes	500

#### 4.5.3.1 Physical layer terms

The IEEE 802.15.4g uses special definitions to characterize the physical layer properties, these definitions are the same as their parent standard, on the following sections a brief summary of those definitions are given

**Energy Detection (ED):** It gives an approximate value of the received signal power within a particular channel; it does not perform any decoding or demodulation process

**Link Quality Indication (LQI):** Is a measurement of the signal quality, by analyzing the difference between the received signal and a cleanly modulated signal (via I-Q constellation comparisons)

**Receive Signal Strength Indicator (RSSI):** Is a measurement of the signal strength, by means of SNR estimation at the receiver end.

#### 4.5.4 *The data link layer properties*

As mentioned earlier the data link layer enables limited network communications by enabling data transfer capabilities between nodes, this layer is split into two sub layers, the MAC and LLC. The Media Access Control enables point-to-point data communications by managing source and destination packages, the transferring of those packages are the most basic forms of networking communications and as such are studied in detail in the next sections.

##### 4.5.4.1 *The MAC Frame on IEEE 802.15.4g*

In a general sense, a MAC frame enables the transmission of data payloads among peer nodes, the frame is composed of a MAC header (MHR), its payload and a data checking footer (MFR), it is carried over the PSDU field of the physical layer, as it can be seen in Figure 4-6.

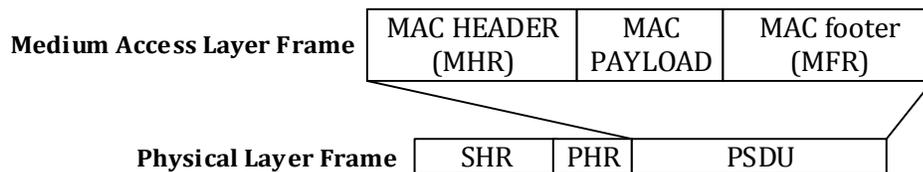


Figure 4-6 The MAC Frame structure [99].

The MHR structure is shown in Figure 4-7, a particular frame headers can vary on size depending on the bit configuration set through the “*Frame Control*” field; these bits establish among other things the frame type (data, command, or acknowledgment), the number of address bits, and security properties. In the next section a brief summary of the MAC header fields is given.

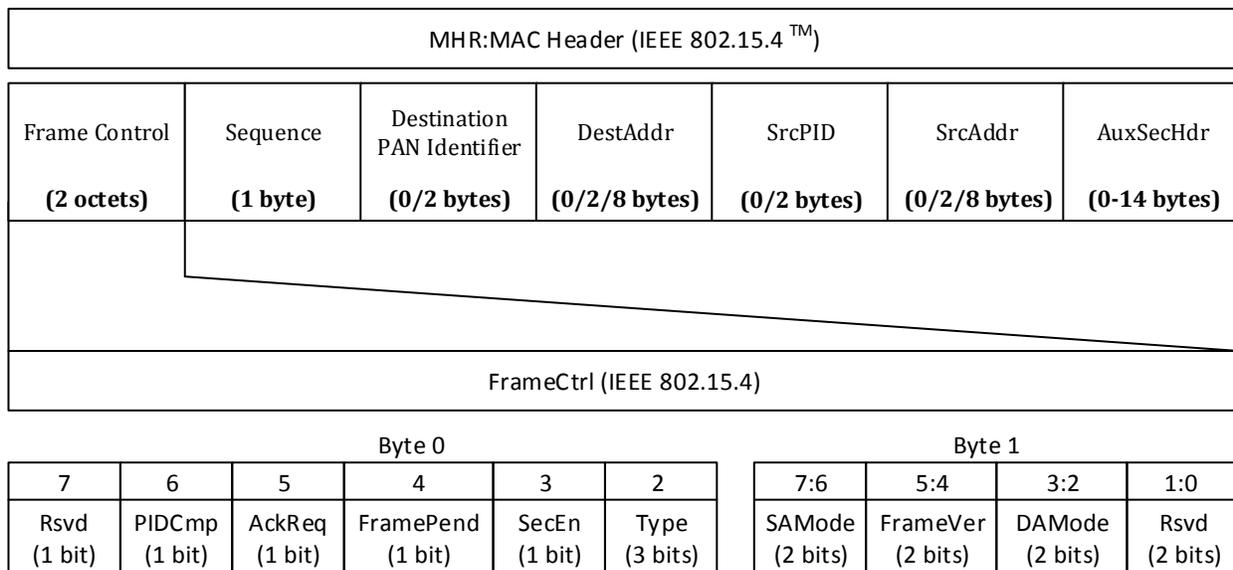


Figure 4-7 The MAC header component of the frame structure [99].

**Frame Control:** The first two bytes of MHR contain information about the frame configuration and intended use, helping the receiver determine the expected frame fields. This enables fast hardware parsing on some transceivers easing off software-based decoding [100], the first field is subdivided into the next bit configurations.

- **Frame Type (Type):** The frame type enables the receiver to perform operations based on the payload contents, for example, a “Data frame” usually contains data intended for upper layers (most transferred frames are of this type), while a “Beacon frame” enables devices to announce their presence and is the basis of the LLC layer. “Acknowledgment frames” are in some cases auto-generated by a remote end upon successfully decoding the frame contents, enabling transmitters to know if the sent message was received or not, “MAC command” frames enable specific operations of the LLC layer, such as network association. In Table 4.5 the field encoding for each of the previously mentioned frame types is given.

Table 4.5. “Frame type” field bit encoding, taken from [101]

Frame type bit encoding	Description (Frame type)
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100-111	Reserved

- Security Enabled (SecEN): The frame bit determines if the presence of an encrypted MAC layer data payload, in case the bit is set, the Auxiliary Security Header (AuxSecHdr) must be present.
- Frame Pending (FramePend): This field should be set if the transmitter has pending frames for the receiver (i.e. on queue), additionally this field should only be used on “Beacon frames” or under special cases ordered by LLC layer control unit.
- Acknowledgment Request (AckReq): This field tells the receiver unit if it must issue an “Acknowledgment frame” once the frame has been validated
- PAN ID Compression (PIDCmp): This field sets if the intended Destination Address (DestAddr) is located on the Personal Area Network (PAN).
- Destination addressing mode (DA Mode): This field is used to select the size of the destination address (DestAddr) field. The destination address can be present, non-present (beacon frames) or implied (acknowledgment frames). In cases where the destination address does exist, it can be short (using the field “Destination PAN Identifier”) or long (8 bytes), in Table 4.6 the possible options are shown according to its bit encoding.

Table 4.6. Destination and Source Address fields bit encoding, taken from [101]

Addressing mode bit value	Description (Frame type)
00	PAN identifier and address fields are not present
01	Reserved
10	Address field contains a short address
11	Address field contains an extended address

- Frame Version (FrameVer): This field ensures future frame compatibility by stamping the frame fields support, for example in 802.15.4-2003 compliant devices this field is set to 0x00, while newer frame compliant hardware’s use 0x01.
- Source addressing mode (SA Mode): This field is used to select the size of the source address (SrcAddr) field. The source address is used to identify the frame origin, and it uses the same principles as the Destination Addressing mode field.

**Sequence Number (Sequence):** This field is used as a frame counter when successive frames are used during transmission between peer nodes, in case this is a Beacon Frame. It specifies the Beacon Sequence Number (BSN).

**Destination PAN Identifier:** This optional field is used only when the “DA Mode” field is set to 16 bits, and it is intended to provide addressing to local PAN coordinators.

**Destination Address (DestAddr):** This optional field must be present when a valid destination exists (“DA Mode”), in case this value is set to 0xFFFF in the 16-bit addressing mode. All devices should decode the package (basic broadcasting).

**Source PAN Identifier (SrcPID):** This field is an optional requirement and it is used to identify a Frame coming from a PAN coordinator, its presence is dictated by the value of “PIDCmp” Field

**Source Address Field (ScrAddr):** This field optionally encodes the address of the frame originator, the presence of this field is dictated by the “SA Mode” field.

**Auxiliary Security Header (AuxSecHdr):** this field is used to store the security attributes of an encrypted frame, its presence is determined by the “SecEN” field.

There are two important MHR data containers, the first one receives the name of **MAC payload**, and contains data from higher layers, it is the actual data being transferred in data communications; data can be encrypted between peer nodes and must be interpreted according to the “AuxSecHdr” field. The other important field is the **MFR**, which contains a 16-bit CRC checksum, this checksum is calculated over the MHR and MAC payload parts of the frame.

#### 4.5.4.2 *The LLC layer*

As mentioned earlier the MAC layer enables basic data transmission over two nearby nodes (i.e. they are visible to each other), this type of communication receives the name of point-to-point links and is the basis of networked communications. These point-to-point links must be handled somehow to enable data transfer by means of routing mechanisms, the LLC layer handles these cases by using a Personal Area Network (PAN) coordinator (i.e. passing thru immediate points).

IEEE 802.15.4g is designed to operate in three different network architectures; these are point-to-point topology, the star topology, and mesh topology. These can be viewed graphically in Figure 4-10, and are explained in the next paragraphs.

**The point-to-point topology:** In this network connectivity model, communications can only occur when both nodes are able to see each other, under the IEEE 802.15.4g standard, the destination

field is used as the filtering method so only a single device decodes and acknowledges the frame; although other units can monitor the data traffic, these most remain passive [101]

**The star topology:** In this mode, communications occur through the use of a central dispatcher unit, known as the Personal Area Network coordinator. This network coordinator works as a data bridge that enables non-visible units to communicate. This type of network introduces two additional concepts, the Full Functionality Device (FFD), and the Reduced Functionality Device (RFD); FFD units are those units that can work as PAN coordinators and offer advanced routing capabilities, as well as message queues. RFD units only work as originator/receiver nodes and can only handle traffic intended to them.

**The mesh topology:** This topology is a generalization of the star topology, where most of the data is transferred through a network backbone consisting of PAN coordinator devices. Since multiple hops are needed to deliver data from the originator node to the destination node, a hop counter and advanced routing capabilities are needed, IEEE 802.15.4g does not provide these services and thus they are often implemented by using proprietary solutions such as ZigBee and MiWi.

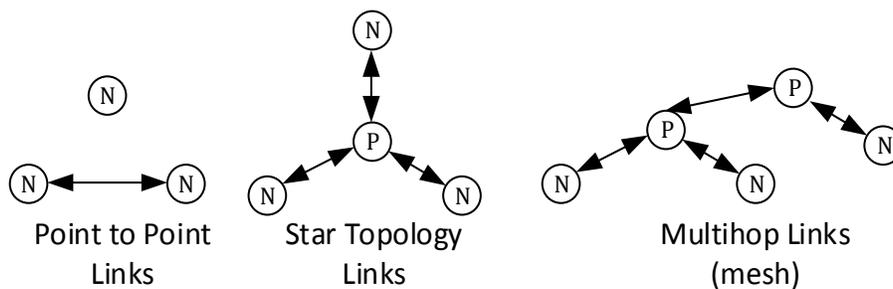


Figure 4-8 The IEEE 802.15.4g link configurations [99].

In order to support the previously described network topologies, the LLC layer uses four frame types to enable network communications among peer nodes. These are summarized in the next sections.

**The data frame:** This container is intended to transmit data coming from higher network layers (e.g. TCP/IP), it contains a destination and source address fields, it can be configured to generate acknowledge frames upon reception by the destination node.

**The Acknowledgment frame:** The acknowledgment frame serves as a “frame received” confirmation message created by the recipient node. This frame lets the originator device know if the previous frame was received or was lost in transit, thus enabling frame rebroadcasting only if necessary.

**The Beacon frame:** When a device is intended to work as a PAN coordinator, it must emit a broadcast frame that lets other nearby nodes know its presence; this frame often contains a unique identifier that can be used to generate an association request. Simultaneously the beacon frame can also be used to denote the existence of a superframe (see section 4.5.4.3)

**The Control frame:** When a RFD/FFD needs to join a PAN coordinator, an association request must be generated, this association request enables to establish security credentials, as well as the creation of routing tables for the FFD device, these types of peer data interchanges are done through the control frame.

#### 4.5.4.3 The superframe format

Although IEEE 802.15.4 does not describe the routing mechanism employed for data delivery, it does introduce a concept known as the superframe that enables slotted communications to occur. Time slotted communications can be used to organize data communications occurring on mesh topologies, by assigning times intended for RFD communications with FFD units, and times where FFD to other FFD communications can occur, this concept can be viewed at Figure 4-9. In this case the contention access period can be used for association requests and network discovery, while the contention free period can be split in to two parts that allow non-backbone and backbone communications to occur.

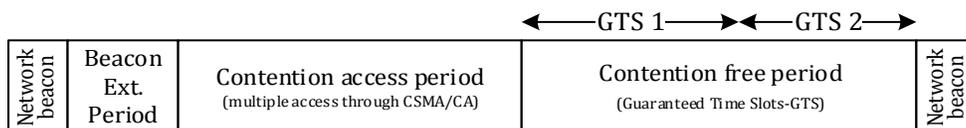


Figure 4-9 The IEEE 802.15.4g superframe format [102].

## 4.6 Wireless Data Communication Interface Development.

As mentioned in Chapter 1 of this thesis, smart meters are characterized for its dependence on two-way data communications; this is often done through dedicated hardware that is controlled by

central microcontroller unit. Although most meter manufacturers use highly integrated units (off the shelf), recent upgradeability requirements have shifted this tendency into using software-based radios that offer better future-proof designs.

Given the aforementioned trend, the author chose to use a software based RF solution by directly handling the physical layer, in this case the MRF24XA unit was chosen as the radio module for the smart metering device, due to its low-level management characteristics, and low overall cost. In this section of the thesis, the author describes the electrical design considerations for the unit, as well as the configurations needed for transforming this device into a partially IEEE 802.15.4g-2012 compliant device.

The MRF24XA transceiver is an IEEE 802.15.4 compliant RF transmitter and receiver unit, which integrates the RF front end (modulator and demodulator hardware), as well as a MAC layer management unit via hardware parsers, enabling fast solution development, while at the same time enabling low-level frame management thru a register-based access.

#### 4.6.1 *Introduction*

The MRF24XA unit implements a proprietary physical layer that enables fast data transmissions, exceeding the speed limits set by IEEE 802.15.4 standard, it offers an almost fully customizable MHR hardware parser that enables to support near IEEE 802.15.4g compliant frames. The MRF24XA includes the following characteristics:

- 3.3 digital logic operation (SPI communications)
- 2.4 GHz Band Operation
- IEEE 208.15.4 PHY and MAC layer compliant
- Proprietary PHY layer that raises overall transmission speed up to 2000kbps with O-QPSK modulation
- Integrated hardware encryption accelerators, supporting 128-bit AES and several block cipher modes
- Automatic CRC checking and generation

- Dual receive/transmit buffers
- Internal LNA and PA units
- Hardware based CSMA-CA transmission management
- Automatic frame acknowledgment mechanism
- ED, RSSI and LQI hardware monitors.

The internal hardware architecture of this unit is shown in Figure 4-10, where all the MAC control process can be controlled via an external microcontroller unit; this enables radio upgradeability and an overall securer communications platform.

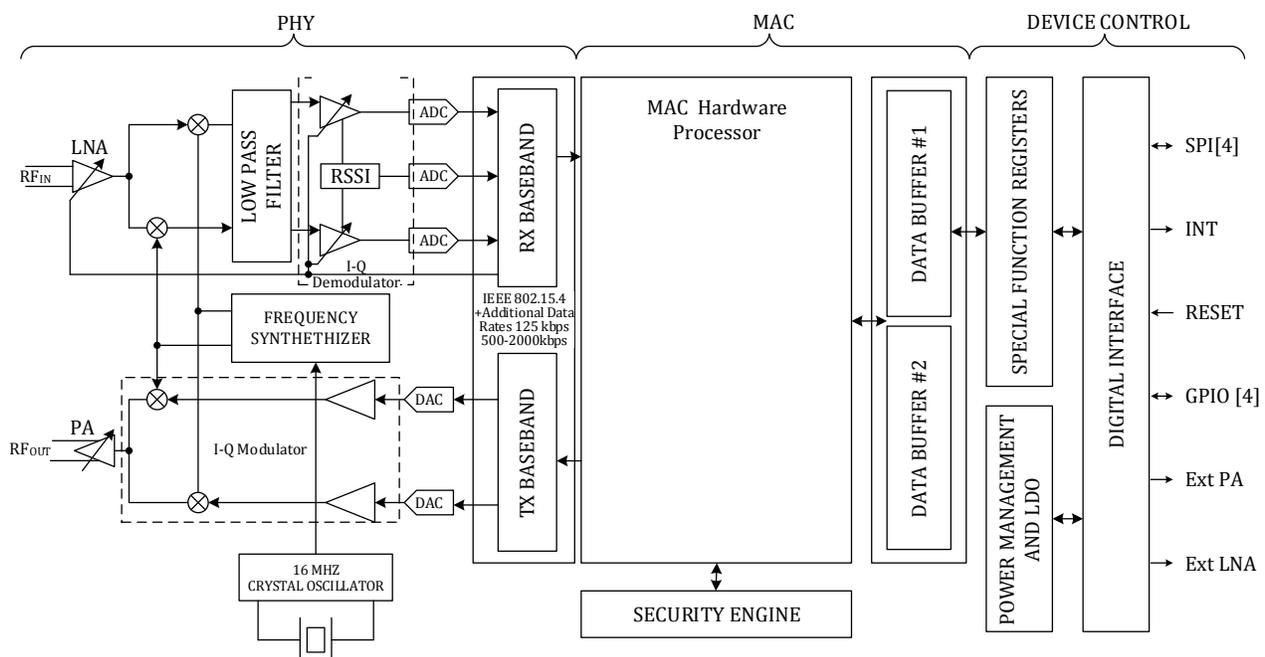


Figure 4-10 The MRF24XA Transceiver internal architecture, based on the architecture described on [100]

The raised security level is achieved by rendering certain hardware attacks useless; especially those that intend to obtain the security credentials by eavesdropping the microcontroller-RF communication channel (see section 3.5.2.2). Also according to [9] the security credentials should never be send out in the clear, even if circuit encapsulation techniques are used.

#### 4.6.2 Electrical design considerations

The MRF24XA unit is encapsulated into a Quad Flat No-leads (QFN) package that is designed to be surface mounted on a Printed Circuit Board (PCB), although it requires a low external component

count, some specific circuit designs techniques must be considered to create a working PCB design. Some of the key areas involved on the development of this unit are:

- Insertion of proper grounding planes
- Insertion of noise grounding capacitors, along all the power traces
- Equal length traces employed for differential signals.
- Impedance matched PCB traces
- Reduction of PCB traces width to reduce capacitance effects on the data lines.

The proposed electrical circuit is given in Figure 4-11, and additional consideration regarding this design is the use of a balance-to-unbalanced transformer (Balun). A Balun works as a high frequency transformer that transforms differential signals (balanced) into single ended signals (unbalanced); this is done so a single ended R-SMA based antenna can be used, these types of antennas are widely available on the market for 2.4 GHz products.

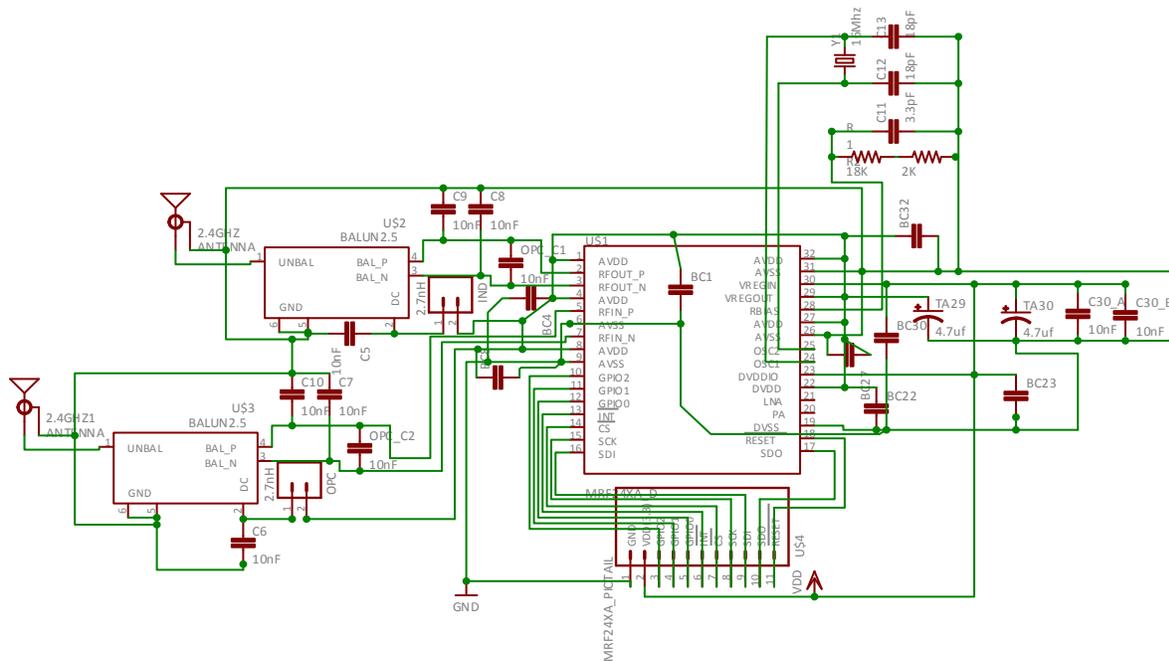


Figure 4-11 The MRF24XA electrical diagram, with external components attached, own design based on [103]

#### 4.6.2.1 PCB Design Guidelines

PCBs provide the mechanical support and electrical connectivity required for electronic designs; there a set of material/configurations that allow circuits to be properly designed, one of the most common PCB material is FR-4, a material composed of glass-reinforced epoxy laminate sheets

The “IPC-Association Connecting Electronics Industries” is an ANSI accredited standards developing association [104], whose goal is to standardize the assembly and production of electronic equipment, and has published a set of guidelines regarding PCB designs. One of these standards is the IPC-2141A, which provides designers with empirical formulas to design PCB traces that transmit high frequency signals, with these consideration and tools available online the PCB layout shown in Figure 4-12 was created.

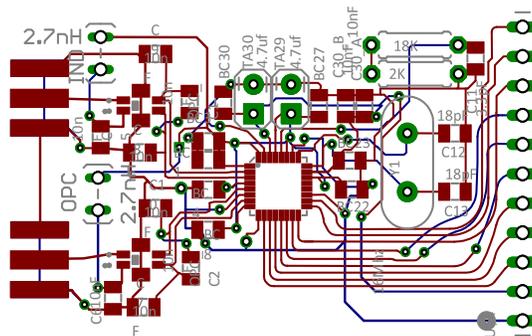


Figure 4-12. Manufactured PCB layout.

Similarly, in Figure 4-13 the actual PCB created is shown, with all the necessary components mounted.



Figure 4-13 The MRF24XA unit mounted on custom made PCB.

#### 4.6.3 Software based radio handling

The MRF24XA unit enables the MPU to outsource low-level tasks like encryption and link management via commands send thru the SPI interface, freeing-up resources for other processes. However, this particular hardware-software layout makes the MPU-MRF24XA communications path susceptible to sniffing attacks. These types of attacks could reveal the security credentials

exchanged during initial hardware configuration, which could be exploited to create network-based attacks like DDoS.

To overcome these security issues and also provide software driven upgradeability the MRF24XA device is mostly handled via software in the proposed meter. This is possible by writing to the configuration registers that can be observed in Figure 4-14.

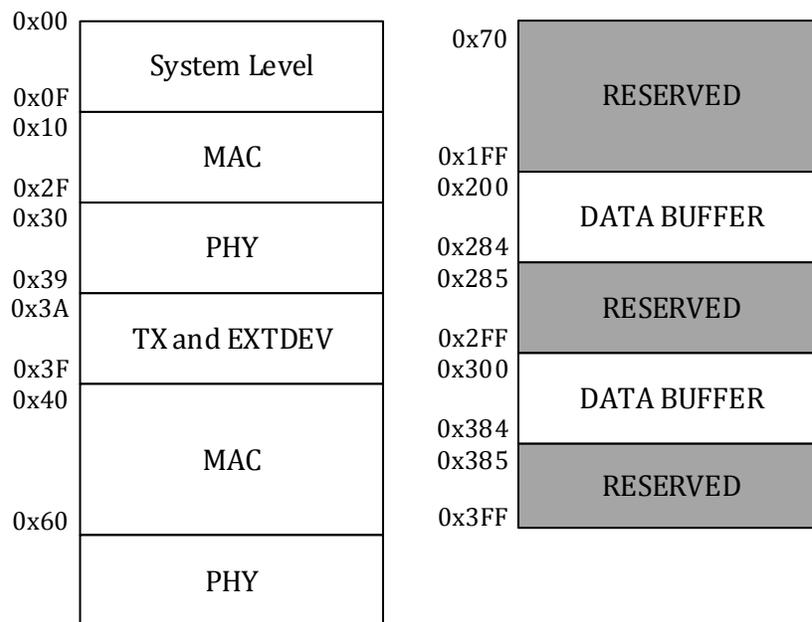


Figure 4-14 The MRF24XA register mapping, physical address space [103]

The configuration registers are divided in three main areas: physical, MAC and control. The physical (PHY) segment contains configurations related to the modulation type, SFD identifiers, CCA timing, TX/RX power and is usually configured once during the booting process (see Figure 4-15 ). The MAC registers contain information regarding the link handling mechanisms (disabled for the proposed unit) and device identifiers, an overview of this configuration process can be observed at Figure 4-15 .

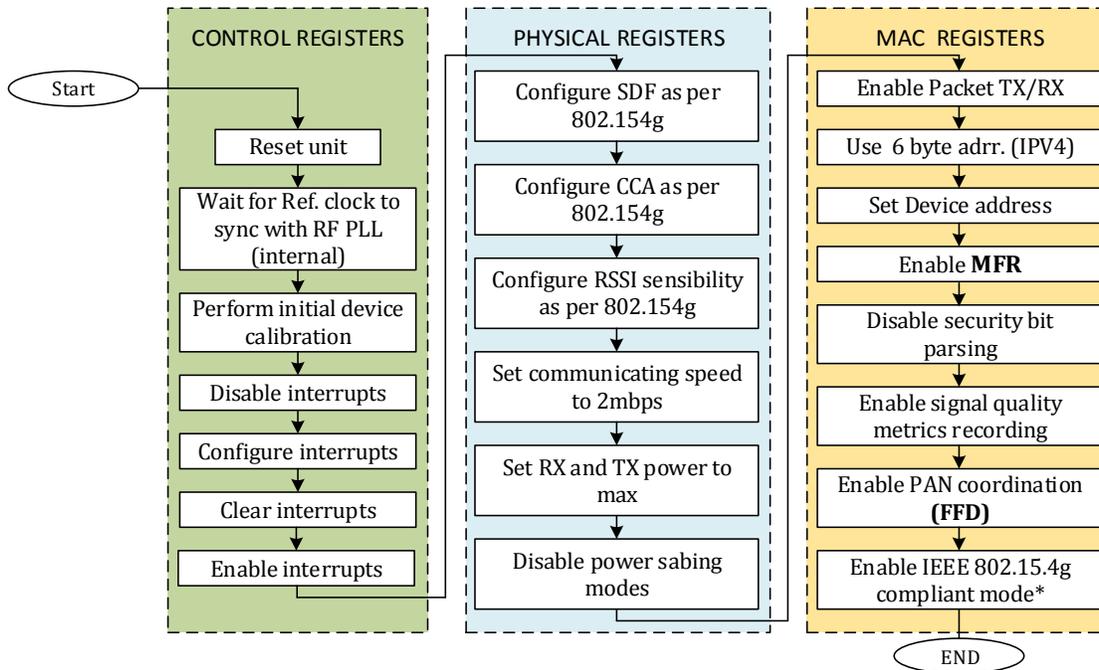


Figure 4-15 Register-based configurations required to handle 802.15.4g thru the MPU

\*The IEEE 802.15.4g mode is set by setting RX/TX channel to 26, setting ADPTCHEN=1 and disabling hardware-based security processing

#### 4.6.3.1 Compatibility of the proprietary MAC header and IEEE MAC header.

Although MHR field in the IEEE 802.15.4-2011 is not compatible with the MHR field in IEEE 802.15.4g a circumvention was made possible by fixing certain frame parameters that are illustrated in Figure 4-16. This is possible due to a proprietary MAC frame-parsing mode available in the MRF24XA that enables to use the MAC fields in an almost unrestricted manner and at the same time enables to hardcode certain values (fixing values).

One of the main circumventions implemented was to enable the use of the second octet from the Frame Control field, by enabling the adaptive channel feature (ADPTCHEN). The ADPTCHEN bit forces the unit to reply an acknowledgment frame on a custom channel, by locking this field to a value compatible with the IEEE 802.15.4g standard the unit is limited to operate in channel 26 at all times but produces a valid MHR structure that will be accepted by fully compliant devices.

Frame Control <b>(2 octets)</b>	Sequence <b>(1 byte)</b>	Destination PAN Identifier <b>(0/2 bytes)</b>	DestAddr <b>(0/2/8 bytes)</b>	SrcPID <b>(0/2 bytes)</b>	SrcAddr <b>(0/2/8 bytes)</b>	AuxSecHdr <b>(0-14 bytes)</b>	MHR:MAC Header (IEEE 802.15.4g™)	
Frame Ctrl (1 octet)	AckInfo (0-1 octets)	Sequence <b>(1 byte)</b>	Destination PAN Identifier <b>(0 bytes)</b>	DestAddr <b>(8 bytes)</b>	SrcPID <b>(0 bytes)</b>	SrcAddr <b>(8 bytes)</b>	AuxSecHdr <b>(14 bytes)</b>	MHR:MAC Header As seen/sent by the MRF24XA <small>(As configured by the first 2 octets)</small>

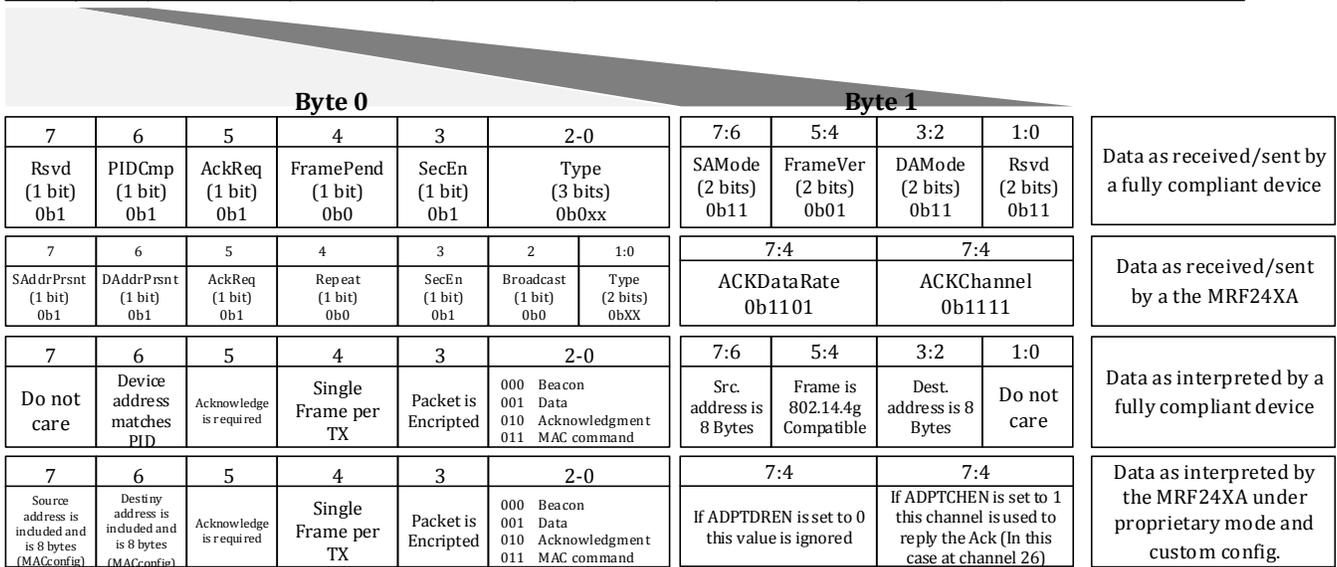


Figure 4-16 The MRF24XA proprietary MAC header in compatibility mode with IEEE 802.15.4g

#### 4.7 Wired Communications Interface Development

Most SMUN networks offer a local/neighborhood area data concentrator that it is often connected to a higher throughput network. Some of these networks are carried by fiber optics, Ethernet or PLC, they are known as the backbone and offer speeds ranging from a few mbps to Gbps. Among Ethernet-based connections certain links can be carried over by Wi-Fi based protocols thus offering high speed and low cost solutions.

Ethernet is formally known as IEEE 802.3 and its specification covers the data link layer and the MAC. For this particular work an off-the shelf PHY chip is used (Figure 4-17), while the MAC is included inside the selected microcontroller. The selected chip is the LAN8740A from Microchip™ which offers 10/100 mbps connections via a Reduced Media Independent Interface (RMII) that greatly reduces the amount of used pins (see Figure 4-18 ).

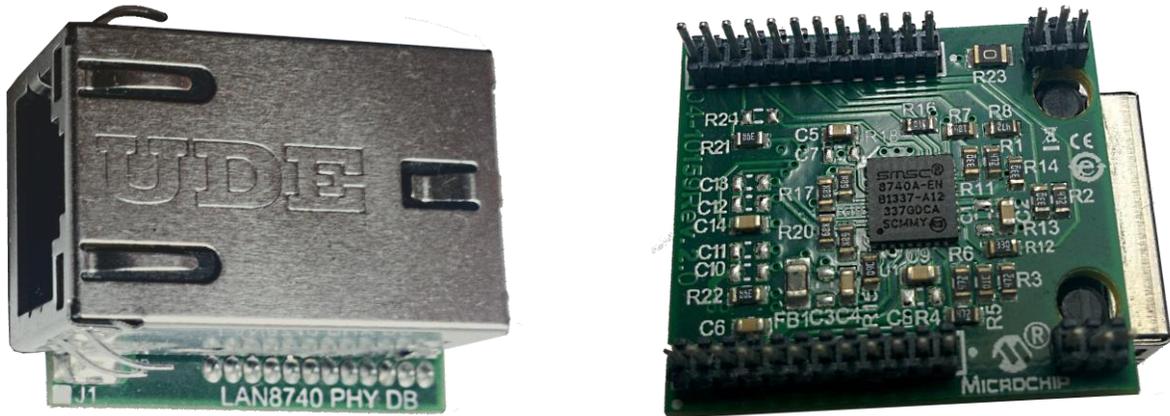


Figure 4-17 The LAN8740A Ethernet PHY mounted in a plug-in module.

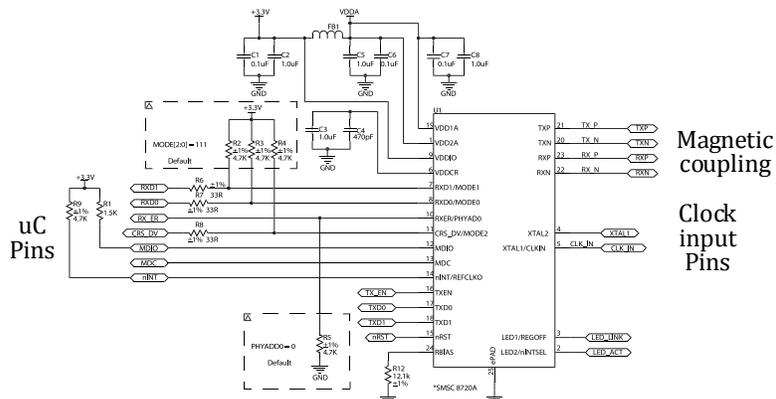


Figure 4-18 The LAN8740A pins used to communicate with the microcontroller.

Since the Ethernet physical layer is carried over a wired medium it uses simplified error handling procedures as well as a simple point-to-point routing mechanisms. These mechanisms contribute to a simpler architecture, details of this architecture can be found at [91].

Initially Ethernet was thought as shared-medium communications channel meaning that communications could be established between two or more entities without an intervening data exchange apparatus, but this has changed due to higher speeds requirements. Most of today traffic occurs through a set of switching equipment that route data according to their destination (in a star-like topology)

## 4.8 TCP/IP IMPLEMENTATION

The Transport Control Protocol (TCP) and Internet protocol (IP) are the basis of modern network communications and carry most of the application data used today (i.e. Word Wide Web). Broadly speaking, TCP is in charge of creating a reliable point-to-point transmission of IP packets by routing them across a set of switching devices and handling network traffic congestions. Meanwhile the IP layer is intended to ensure transmission of packets from host to host, these packets contain the information intended to be transmitted among different applications or devices.

The TCP/IP structure follows the OSI layerization model by successively encapsulating traffic coming from higher layers. An in depth discussion of the frame format and related services can be found at [91]. For the purposes of this project an open source TCP/IP stack was chosen in order to accelerate development, the selected suite receives the name of Cyclone TCP and supports the services illustrated by Figure 4-19.

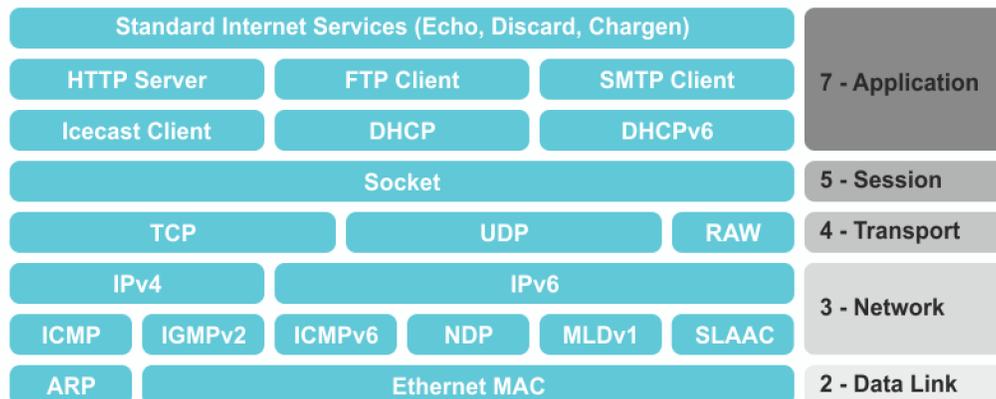


Figure 4-19 The TCP/IP stack available on the cyclone TCP distribution [192]

The services provided by the Cyclone distribution enable any supported device to connect to the internet and act as data clients or data servers over the internet. This characteristic was exploited in order to bring network connectivity to the designed smart metering unit. However, a custom driver for the Ethernet and radio module where needed to be developed, these were accomplished by hooking up the device registers to the MAC layer of the TCP/IP suite, based on the hardware profile available at [193].

## 4.9 TLS

Due to the nature of TCP/IP traffic, data communications occurring between two parties can be easily intercepted along the traffic route. If these communications are not protected (from the information security point of view) data retrieval or even impersonation can easily be mounted. To prevent these types of attacks several protocols have been established, ranging from low-level security suites to higher-level protocols. One of such mechanisms is the SSL/TLS suite, which works above the transport layer (OSI layer 4).

The TLS suite has as its foundation the use of asymmetric key cryptography to establish a session key that will be used to encrypt data during bulk data transfers. The asymmetric key in this case works by creating a chain of trust (i.e. the basis of public key infrastructure) thus providing confidentiality and authenticity (client authentication is optional). In Figure 4-20 the basic handshake mechanism for unauthenticated clients requiring TLS communications is illustrated.

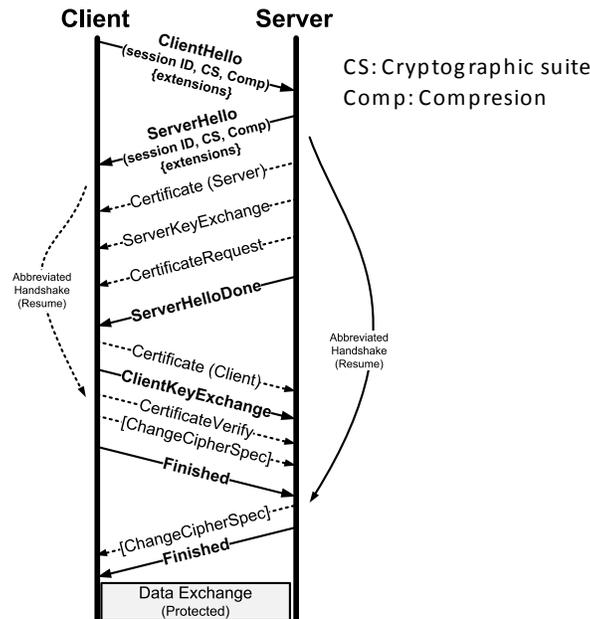


Figure 4-20 The TLS key interchange mechanism, adapted from [91].

SSL/TLS has evolved during the years and currently TLS 1.3 is under approval, each version is generally intended to raise overall security and fix known security holes. Since the TLS 1.1 is supported by the Cyclone TCP library, it was adopted to satisfy the security requirements of the smart meter communications channel. However, due to limited computing resources available on

the MPU only a single TLS mode was enabled. This mode is identified as the TLS\_DH\_RSA\_WITH\_AES\_128\_CBC\_SHA; it uses TLS 1.1 as the data exchange mechanism, with Diffie-Hellman as the key exchange, using AES-128 in chained block mode and SHA as its hashing function (see Annex G).

Since a timing attack resistant AES function was previously developed, the provided Cyclone AES subroutine was replaced with a time-optimized version, the relevant tests are presented in Chapter 6 of this work

#### 4.10 Attacks on Smart Meter Network Infrastructure.

Many devices are currently network enabled, meaning attacks can be mounted on four network levels described in the next section, these levels might vary depending on the specific application or network connectivity capabilities of the smart meter device.

**Client side attacks:** These attacks are based on accessing through desktop clients and accessing user information, if the attack is limited to mere service users only the services associated to user's accounts can be accessed. If the attack is mounted inside the company computers, ample disruptions can be caused depending on the employee permissions. These attacks are often implemented by using malware, viruses and botnets [77].

**Server side attacks:** These attacks are often intended to attack specific computer systems that contain valuable data, including billing services or customer databases, some futuristic visions suggest accessing servers to remotely control smart meters.

**Network attacks:** These types of attacks intend to degrade the network service by means of DDoS attacks, rendering the utilities communications channels useless. This could affect the metering capabilities of the metering network and could cause service disruptions.

**Hardware attacks:** These types of attacks were described in section 3.5.2.2, and are considered to have a low impact risk, unless the network credentials are recovered and no revocation mechanism exists.

## CHAPTER 5

### 5. LOSS ASSESSMENT ON DISTRIBUTION NETWORKS

#### 5.1 Introduction

This chapter pretends to introduce the reader to meter tampering techniques intended to bypass or alter the quantity of metered energy; at the end of the chapter, an energy theft-detection methodology is proposed. This methodology uses a central observer unit similar to one described in [3], but it differs itself by using instantaneous current waveform data vs the accumulated power consumption data which enables to obtain results in real time.

#### 5.2 The Mexican Electricity Market

Energy consumers for the Mexican market can be divided in 3 type of customers based on their consumption type and energy tariffs: industrial, commercial and residential. Some of their main characteristics are described below [105]:

**Industrial:** They consume bulks amount of energy. Their energy consumption pattern is mostly constant during working days, making their energy requirements predictable, they are usually connected directly to HV/MV lines, generating little additional losses. They are mostly located on specific sites and require uninterrupted supplies.

**Commercial:** They usually represent smaller manufacturing plants, shopping centers, office buildings and small specialty shops; they are usually connected to MV/LV lines. In the case of MV connected loads, their energy patterns can be predictable, and energy losses can be below average, but for LV connected clients, their consumption patterns are unpredictable, and as such, they are often considered residential consumers with a higher tariff.

**Residential:** They are connected to LV lines, and have an unpredictable load pattern, with high-energy consumption in the early morning and late evening, their energy tariffs are heavily subsidized. They impose and additional trouble to market growing plans due to poor urban planning.

This chapter is intended to identify energy theft at the low voltage side of distribution systems and as such is tailored to residential consumers. In CFE there are about 37 million contracts, of which 33 million are residential customers, accounting for about 25% of total revenue of CFE. Although the total revenue seems low at first glance, this revenue is much higher due to heavily subsidized tariffs, which are not accounted as revenue, but as federal received funds [106], which can range from 0-70% of the energy cost.

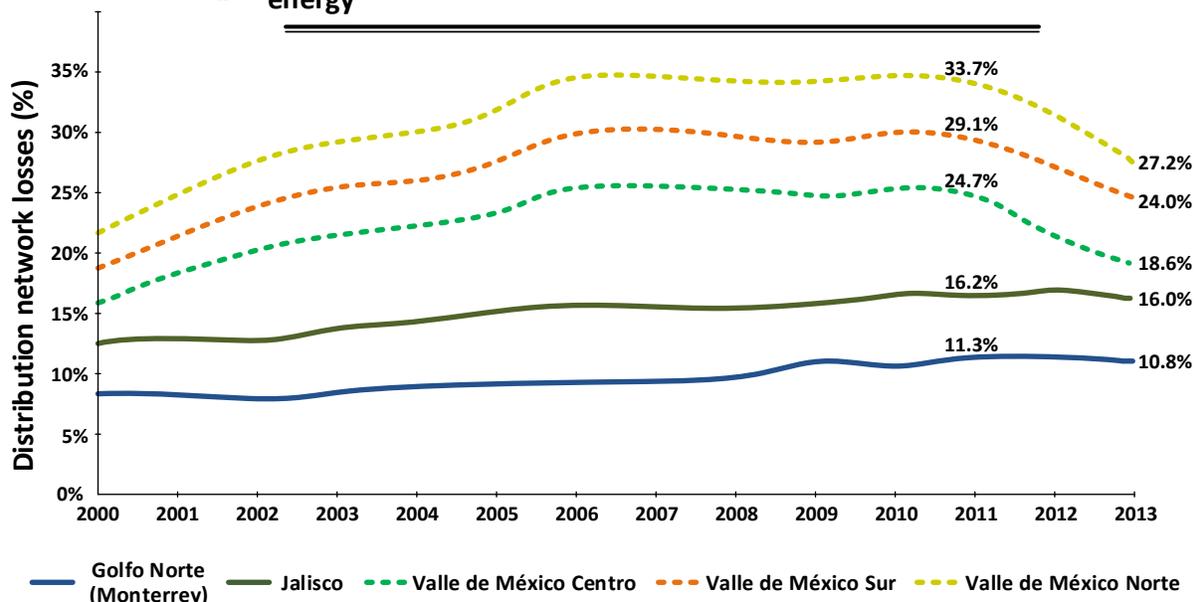
Traditional utilities measure energy thefts in terms of economic losses, which can be minimized by pre-altering the base price of energy in accordance with the expected non-technical losses. However, in CFE, this solution is not available, because the government fixes the unit price solely based on production costs and expected technical losses, which means that CFE in most cases reports negative profits. Due to these negative results, CFE has tried to improve its energy theft indexes by replacing meters and doing energy theft identification based on consumer consumption history. Although these techniques have given positive results, there is work to be done.

### **5.3 Energy Theft in the Mexican Market**

Energy losses in Mexico is a major problem, accounting for almost 35% percent of the supplied energy in certain localities vs the 8% world average (see Figure 5-1 ). This is mostly due to the aging infrastructure and customer energy theft practices. The low overall income and certain society habits makes energy theft a common activity, with some official reports indicating up to 200000 metering thefts are detected each year [107].



**Among the major consumption areas of electricity in Mexico, the Valley of Mexico accounts for the greatest technical and non-technical losses of energy**



**Note :** Prior to 2011 estimates were based on the overall losses of the former Luz y Fuerza del Centro . Data from 2011 onwards is based on recorded measurements taken by CFE, after dividing the Valley of Mexico into 3 Distribution Divisions

**Valle de México Norte:** Gustavo A Madero and 34 Mexico State municipalities

**Valle de México Centro:** Álvaro Obregón Azcapotzalco Benito Juárez Cuauhtémoc Cuajimalpa Gustavo A Madero Iztacalco Iztapalapa Miguel Hidalgo, Venustiano Carranza y 12 Mexico State municipalities

**Valle de México Sur:** Álvaro Obregón, Cuajimalpa, Coyoacán, Iztapalapa, Magdalena Contreras, Milpa Alta, Tláhuac, Tlalpan, Xochimilco and other Mexico State municipalities

Source: Coordinación Comercial, Subdirección de Distribución, Comisión Federal de Electricidad 2014

Figure 5-1 Mexican electric utility reported losses through the years [108].

According to CFE, 98.5% of energy theft occurs at LV lines (i.e. the low side of transformers in distribution networks); with 82% of these being residential customers and 16.5% commercial customers. CFE attributes this high tendency to steal energy from LV lines to the low risk of electrocution and the ease to tap into air wires [109]. As stated by CFE, energy theft is considered to have taken place whenever the next alterations are present:

- Seals tampering
- Seals forgery
- Direct taping to line
- Load connected before the meter
- Pointing needles altered (mechanical)
- Stuck rotating disk (mechanical)
- Meter settings altered

- Current return circuit open
- Altered connections on the meter base
- Open voltage or current coils
- Inverted metering
- Jumper installed on links/blades
- No contract with meter installed
- Software alteration
- Meter substitution (using a malfunctioning meter)
- Altered meter readings

CFE also reports that most energy theft is done through direct wiretapping, which is the easiest method to perform but it is also the simplest to detect [110]. In Table 5.1, the list of most commonly detected metering alterations is given. Although the list only encompasses a single municipality it is likely that the results can be applied to the Mexican utility in general.

Table 5.1. Reported metering alterations on the Mexico state municipality of Chimalhuacan, adapted from [110].

<b>Reported alteration</b>	<b>%</b>
<b>Direct tapping to line</b>	65.1
<b>Load connected before the meter</b>	23.2
<b>Meter substitution</b>	5.2
<b>Jumped links (bypassing)</b>	2.2
<b>No contract with meter installed</b>	1.2
<b>Other cases</b>	1.9
<b>Altered connections on the meter base</b>	0.3
<b>Stuck rotating base</b>	0.2
<b>Meter settings altered</b>	0.1
<b>Inverted meter</b>	0.1
<b>Tampered seals</b>	0.7

According to the CFE strategy on “Infrastructure and Investment framework for the electrical sector”, (POISE) three major activities must be done to improve non-technical losses indexes in the next years. These are listed below.

- Prompt financial and physical resources allocation.

- Gradual introduction of advanced metering infrastructure with illicit activities detection.
- Amendment of the legal framework in order to classify energy theft as a federal felony

Since the inclusion of AMI technology is a core component of this strategy, some deployments have already started to appear. One of this pilot tests is located on the central of Mexico valley in an area known as Polanco, this technology is based on the energy balance concept that was described in section 1.8.4.4 .

#### 5.4 An Overview of Energy Theft Techniques

Energy theft in its basic form involves obtaining electrical energy without having to pay for it, and as such can be performed through several means, although they can be grouped according to their attack domains, which are listed below.

**Direct wire-tapping:** this method is the simplest one and it consists on pulling a direct line from the low voltage transformer end into the customer load. It is commonly found on flea markets throughout Mexico City; it is a common source of fires due to line overloading (see Figure 5-2).

**Modification of the circuits around the metering area:** These types of attacks try to reduce the amount of billed energy by altering the circuits that feed the meter terminals; these can range from flow reversal to phase alteration on three-phase loads. These methods often rely on the tampering of security seals and use of hidden wires to conceal the alteration, and as such are often undetectable to the naked eye.

**Internal meter hardware modification:** These attacks often require deep knowledge of the measuring apparatus, and as such are often performed by dishonest utilities employees or highly skilled individuals. Since modifications are done inside the meter, they are often unnoticeable unless an external calibrated apparatus is used to evaluate the meter performance.

**Meter firmware alteration:** These attacks are applicable to AMR and AMI devices, and rely on software manipulation of a device, mostly through its communication interfaces. Although initially considered a low risk area, extensive network connectivity as well as off-the-shelf attack methodologies could represent a serious risk in the near future.

In this section an overview of external circuit modifications is discussed these types of attacks are often possible due to simple nature of bypassing mechanisms as well as the possibility of concealment of illegal connections on the end user premises.



Figure 5-2. An example of direct wire tapping of air wires in many flea markets around the city of Mexico.

#### 5.4.1 *Attacks on the security seals*

Most people think that tamper-evident mechanisms work to halt attacks, but in practice, they only work as deterrents. Security seals are often found in meter installations, and are used extensively to secure the meter to its fixture or to prevent internal tampering of the meter. Tamper evident seals introduce a psychological flaw, due to the blind confidence they introduce to the utility personnel. This flaw is derived from the fact the utilities assume that their tamper-evident seal will give away any type of attack, but in practice, this is not always true.

The flaw is inserted because most routine checkups only check for the status of the seals, meaning that a properly bypassed seal can trick an inspector, and cause a tampered unit to go unnoticed for several years. These types of attacks can be traced back to the start of metering history, and are first exemplified by the reuse of lead seals, or more modernly by the substitution of broken plastic seals, as seen in Figure 5-3.

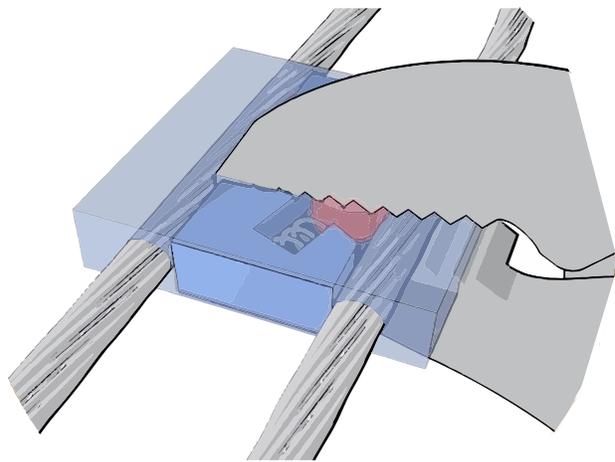
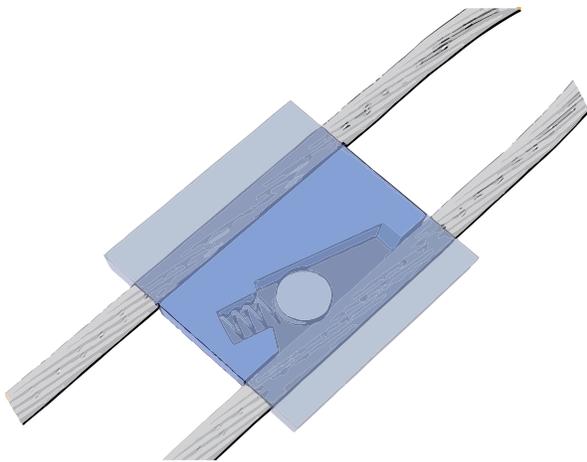


A) A tamper-evident device used commonly by CFE to secure its meters rings.

B) The tamper-evident device is available online through [111], no questions asked.

Figure 5-3 Due to their multipurpose nature, tamper-evident seals are widely available to the general public.

Seals are also susceptible to simple bypassing mechanisms, including use of oil, heating and pressure. An example of these types of attacks can be seen in Figure 5-4 for a tamper-resistant device known as the cable seal.



C) A wire based seal, used by many utilities world wide

D) Bypassing of the seal mechanism by using a pair of pliers

Figure 5-4 Tamper proof mechanism bypassing, method disclosed by a tamper-proof device manufacturer [112].

Utilities also rely on novelty seals to deter tampering, but they might also be susceptible to simple bypassing techniques. One of these novelty seals is known as the padlock seal, which uses a two-

piece device that has clear outer shell that can be used to quickly detect a tampered device, as seen in Figure 5-5.



A) A tampered seal front view

B) A tampered seal back view, with preexisting damage

Figure 5-5 A tamper-resistant padlock with a transparent enclosure.

The padlock mechanism was designed to be self-destructible upon removal, leaving the protected device unsealed and thus tamper evident, but these seals are susceptible to mechanical alteration by inserting external parts, such as the pin illustrated in Figure 5-6.



A) Tamper resistant seal bypassed by the insertion of a lubricated pin



B) Complete separation of the insert; notice a small damage in the legs.

Figure 5-6 A tamper-resistant padlock mechanism bypassed by inserting a pin through an inferior cavity.

The attack proposed in Figure 5-6 enables an attacker to substitute the broken wire that used to secure the meter. Moreover, as it can be seen in Figure 5-7 it leaves very little evidence, which can be unnoticeable in most cases. To illustrate the feasibility of this attack Figure 5-8 shows the final reassembled padlock seal without a noticeable tampering evidence. The created hole can be sealed-off with resins to prevent dust accumulation and prevent future detection.



A) Inner seal insert after polishing the damaged area

B) Outer padlock enclosure, showing the inflicted damage

C) Zoom-in to the damage area, which in most cases will be invisible to the naked eye.

Figure 5-7. The bypassing of tamper-resistant padlock mechanism, by perpetrating minimum damage.



A) A tamper-proof seal with replaced steel wire; front view      B) A tamper-proof seal with replaced steel wire; back view

Figure 5-8 Reuse of tamper-resistant padlock mechanism, with no visible damage.

In cases where the tampering bypass is unsuccessful, padlocks can be forged, even with the presence of unique identifiers, as shown in Figure 5-9. This particular padlock model is used by CFE and features a non-standard QR field used for automated reading and verification (i.e. it prevents generic substitutions). Although the use of a non-standard QR code creates a false sense of security, the seal ID can be forged by using custom programs or modifying generic programs as shown in Figure 5-10. The particular QR code used on this label was identified by the author as being a “type Q” QR code that enables up to a 25% percent data loss, with a non-optimal mask number 7, (the QR code uses certain masks to disperse the data content and minimize error probability).



A) A genuine tamper proof seal with a unique identifier. B) Back of a tamper proof seal. C) A digitally-forged seal using a custom QR generator.

Figure 5-9 Forgery of tamper proof seals, with the presence of valid ID fields.



Figure 5-10 A custom build tool to forge seal IDs, based on an open source QR generator.

Since padlocks are susceptible to bypassing, newer seals have appeared on the market. For example, the one shown in Figure 5-11 is known as “bolt type seal” and is marketed as a single use device, which must be broken, but can be susceptible to forgery attacks (Figure 5-11).



Figure 5-11 A bolt type tamper-proof seal, with printed ID.

As always, there are other types of locking mechanisms. One example of these devices is the barrel lock, although secure, it is susceptible to tool removal counterfeit, as well as utility personnel misuse or mislay. The barrel lock is shown in Figure 5-12.



A) A barrel lock mechanism, side view



B) Removing tool

Figure 5-12 A mechanical barrel lock mechanism for meter locking rings, with its associated tool.

To increase the security of barrel mechanisms, some manufacturers have started to use electronic authorization mechanisms, such as the one shown in Figure 5-13, but due to their simple nature they can be bypassed by lock peaking.



Figure 5-13. An electronic authorization key mechanism for barrel locks.

Also newer generation locks are following the smart grid movement, and have transitioned to a microcontroller driven mechanism, such as the one shown in Figure 5-14 and Figure 5-15. This particular locking mechanism uses a symmetric key transmitted over RFID that enables lock aperture only by authorized key holders. This gets rid of the problem of key management of large utilities and enables traceability. In this case the keys are transmitted over a wireless channel from a central server to personnel key.



Figure 5-14 A next generation boxed meter lock (pre-production sample), with RFID key based mechanism, side view.



Figure 5-15 RFID boxed meter lock (pre-production sample), front view.

#### 5.4.2 *Alteration of feeding circuits*

After the tamper-evident devices have been bypassed the actual energy theft process begins. On this section a small summary of the employed techniques are reviewed. These techniques mostly rely on altering the current passing through the current coil of the metering device, or altering the measured voltage. In either case the main objective is to reduce the amount of measured power. Many of the proposed alterations given in this chapter are the based on the ones described by [113].

##### 5.4.2.1 *Preamble*

Wattmeters are used to measure the amount of consumed energy in a given a moment, but do not accumulate the result. Watt-hour meters in other hand are used to record the total amount of consumed energy, usually in kWh. Watt-hour meters often follow two standardization bodies to measure AC demand: ANSI C12 and IEC 62052, which dictate accuracy limits, connection diagrams, and overall requirements and test parameters [114] [115].

In ANSI regulated countries two meter types are widely available on the market: the bottom-connected meters (type “A”) and detachable meters (type “S”), with the latter being the most common, since their base allows used of concealed wiring that lowers safety risks and enables fast replacement of units. Type “S” units enable multiphase low voltage metering in a common enclosure, and its requirements can be found in section 10 of the ANSI C12 family of standards [114]. In Figure 5-16 the internal wiring of a single-phase wattmeter is given (type “S1”).

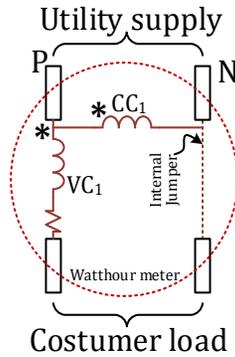


Figure 5-16. Internal wiring of a single-phase watt-hour meter, based on the S1 form.

The connections used on the ANSI C12 family of standards are based on the historic relevance of the electromechanical induction meter that uses a current and voltage coil to produce a rotation proportional to the amount of power consumed ( $V * I$ ). The connection style has been inherited to the electronic meters (although smaller form factors could be possible) enabling seamless upgradeability in most cases. In Figure 5-17 an equivalent wiring circuit of the ANSI “S1” type is given. This circuit will be used in the next sections to describe common alterations on the meter feeding circuits.

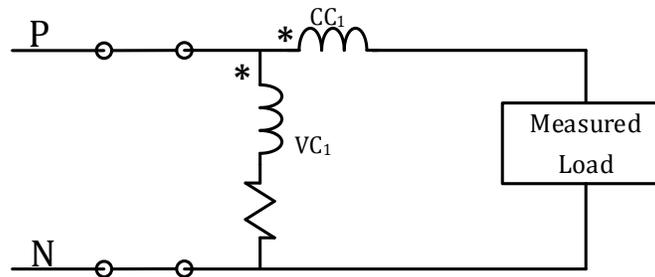


Figure 5-17 A correctly wired metering circuit, where \* denotes the instantaneous polarity.

#### 5.4.2.2 The current inversion method

By a Kirchhoff current analysis it can be shown that inverting the phase (P) and the neutral (N) does not alter the amount of measured energy since the resulting product is always positive. However, if an alternate ground return path is created, then an unmeasured load can be supplied by the circuit shown in Figure 5-18. The circuit can also be expressed mathematically (Eq. 5.1) and essentially the amount of stolen energy is equal to the grounded load. The situation is unnoticeable to most meters but can be detected by technicians by dismounting the meter and testing the terminal block polarity.

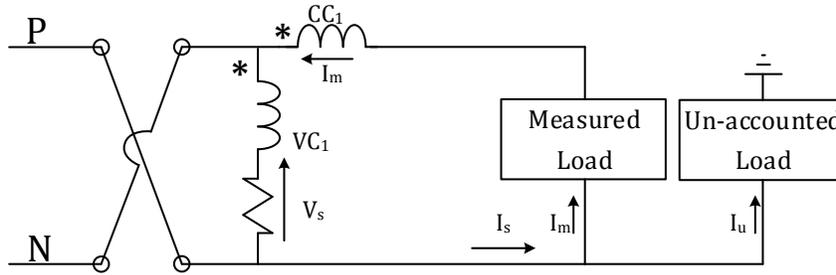


Figure 5-18. An inverted metering circuit with an additional return path.

$$P_s = -I_s * (-V_s) \quad \text{Eq. 5.1}$$

$$P_m = -(I_s - I_u) * -V_s = -I_m * (-V_s) \therefore P_s \gg P_m$$

where

$P_s =$  Supplied power (utility)

$P_m =$  Measured power

$I_m =$  measured current;  $I_s =$  supplied current;  $I_u =$  unaccounted current

$V_s =$  supplied voltage

#### 5.4.2.3 Opened return line

Another common modification is to disconnect the ground return path in the meter base and replace it with an alternate ground path as illustrated by Figure 5-19. This type of theft is unnoticeable in electromechanical meters, but it can be detected by some electronic meters via firmware handling.

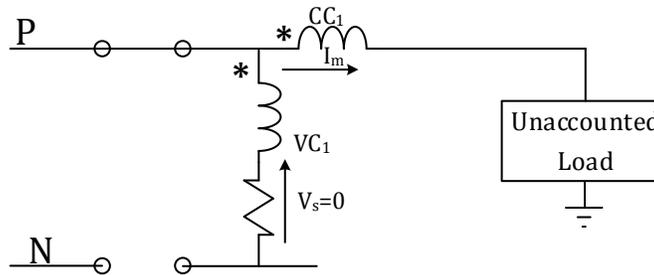


Figure 5-19. An open neutral return circuit causes most-meters to measure a zero consumption

#### 5.4.2.4 Current coil bypass.

This type of intrusion is the most basic one and it consists on installing a jumper across the current coil (see Figure 5-20). The jumper can be adjusted to match the current coil resistance, and thus create a meter that partially measures and does not raise and alarm to the utility billing system. A clever

method of disguising the jumper is install it behind the base terminal blocks, as it can be seen in Figure 5-21.

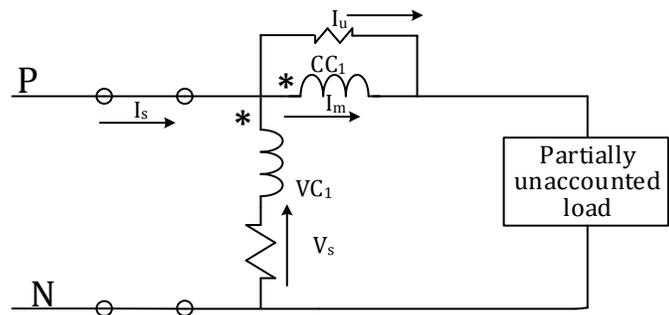


Figure 5-20 Jumper based current coil bypass



A) An ordinary looking meter base



B) A hidden cable under the current-coil terminal block

Figure 5-21 A current bypass method, hidden behind the terminal block.

### 5.4.3 Mechanical meter tampering

Some consumers will go one-step ahead on the theft chain and will target the measuring apparatus, in occasions bypassing tamper-detect mechanisms and forcing security enclosures. These types of attacks are often done by highly skilled persons or dishonest utility personnel, but in some cases, the attacks are so rudimentary that they are done according to “word of mouth”.

Electromechanical meters were the most commonly deployed meters up until the early 2000s. These devices work on the same principals as the induction motor, where the magnetic forces produced by the current and voltage coil are used to rotate a primary disk. This primary disc is

attached to several mechanisms that enable to record the total amount of energy consumed, usually by means of dials. These disks often have low inertia characteristics that enable them to handle fast start and stop conditions.

5.4.3.1 Lowering the rotational speed.

One of the most common methods of slowing down an electromechanical meter is to use a magnet to alter the magnetic field produced by the current coil, or to induce a braking force on the rotating disc, thus reducing the amount of registered energy. Some variants of this method include putting a glass of water on top a meter, with variant degrees of success, although many modern electromechanical meters use aluminum pieces to lower their magnetic properties.

Some devices are also susceptible to magnetic disorientation or mounting position, and stop measuring once their position is altered. This can be implemented by energy thieves by movable mounting panels or detachable fixtures as the one shown in Figure 5-22.



A) An ordinary looking meter



B) Device is easily removed due to deficient fixture

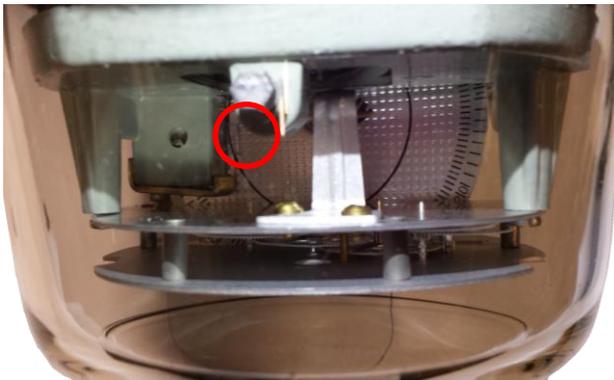


C) The Meter design allows an easy on-easy off removal.

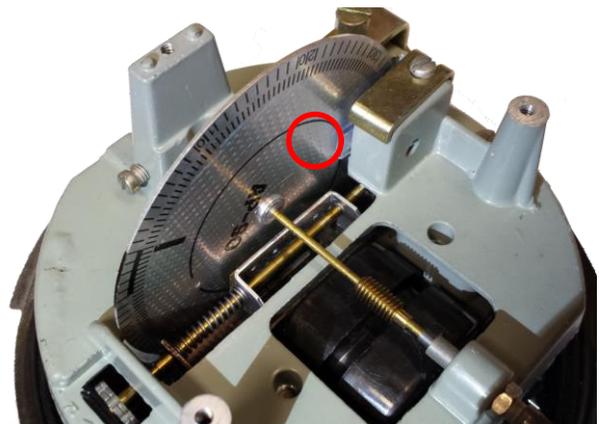
Figure 5-22 Mechanical wattmeter tampering by means of disorientation.

### 5.4.3.2 Disc breaking methods

As mentioned in the introduction some thieves will alter the inner workings of a metering device. In most cases these modifications will seek to reduce the amount of registered energy. These types of modifications are hard to detect on routine inspections and in most cases will require specialized team meter disassembly to locate the intrusion. In Figure 5-23 an example of such an intervention is given. For this particular case, a transparent piece of plastic is inserted between the disk and its supporting structure, producing a breaking force that lowers the amount of measured energy.



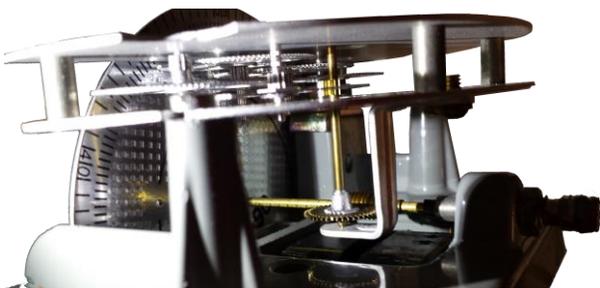
A) An almost invisible breaking mechanism



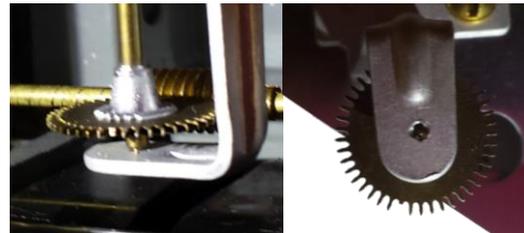
B) The breaking device is in most cases only visible upon complete device disassembly.

Figure 5-23 Mechanical alteration that seeks to break the disk free rotation.

Some severe ways to alter the measured energy consumption is to alter the dial mechanism that stores the amount of recorded energy, as it can be seen in Figure 5-24. These types of intrusions can range from gear damage to complete substitution of the gear ratios, and clever modifications can in some cases pass field calibration tests.



A) A hard to see damage.



B) Close up into the damaged area

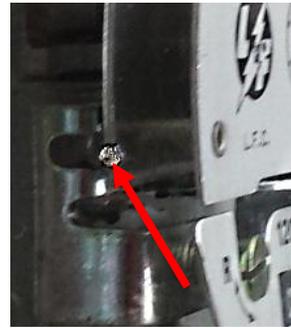
Figure 5-24 A destructive mechanical alteration that seeks to break the disk free rotation, by modifying the dial mechanism.

### 5.4.3.3 Dynamic disc breaking

Some clever approaches of stealing energy are the dynamic disk breaking methods; these are done by inserting foreign objects into the device that can be quickly removed before an inspection. In Figure 5-25 a hole is present on the meter protective case, and can be used to insert a small wire that can alter the metering mechanism.



A) Dynamic alteration of measurement based on the insertion of an object through a hole



B) A digitally-enhanced photo of the existent hole

Figure 5-25. The presence of a hole in the meter case indicates a possible tampering case

### 5.4.4 Electronic meter tampering

Electronic meters appeared on the early 1990's and usually offer better tamper-resistant characteristics, as well as improved measuring capabilities, in Figure 5-26 a simplified block diagram of the main components of an electronic meter is given. This diagram illustrates the core components, and most of them are shared with the smart meter architectures.

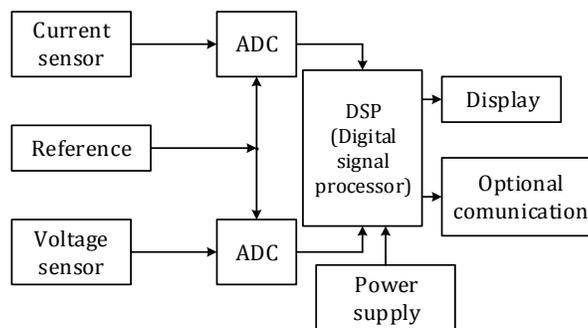


Figure 5-26 Typical architecture of an electronic meter

In Figure 5-27 a PCB image of an electronic meter used by CFE is given. This particular meter is a single phase measuring unit with a RFID module that allows the meter to be operated on a prepaid basis (up to 17% of the costumers of CFE do not pay the electricity bill [116])

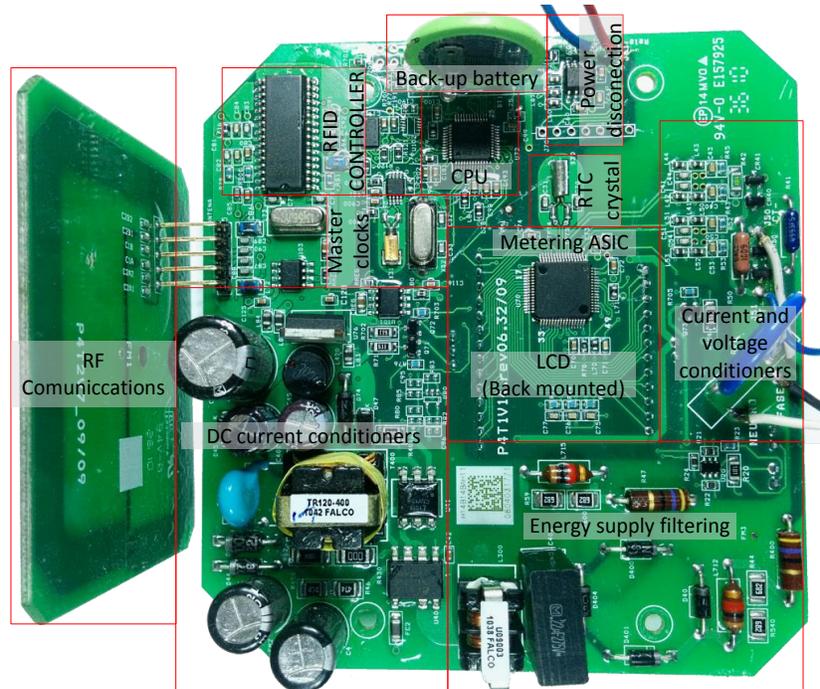


Figure 5-27 The PCB layout of a production electronic meter.

#### 5.4.4.1 TC Burden modification

The current passing through the terminals in electronic meters is sampled by an ADC circuit fed by a current transformer (CT); this circuit uses a burden resistor to transform the current generated by the CT into a voltage signal and thus the burden value determines the current to voltage ratio. A skilled attacker can replace the burden value in order to record a lesser amount of energy. In Figure 5-28 an example of this attack is shown.

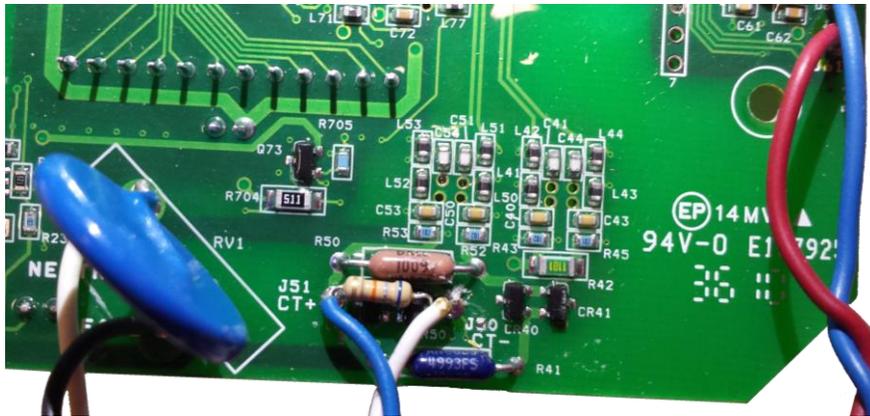


Figure 5-28 Current transformer burden alteration.

#### 5.4.4.2 RTC clock alterations

In electronic meters the amount of registered energy is calculated by integrating the recorded consumption in a given period. If the time base is altered, the amount of recorded energy can be altered. In Figure 5-29 the time clock crystal is compressed and thus the inner quartz runs faster than expected. Ingenious attackers can even replace the clock crystal to whatever value they choose.

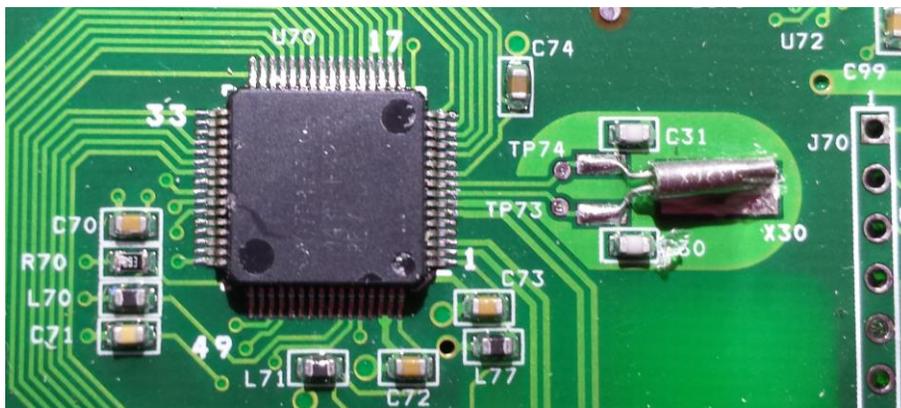


Figure 5-29 RTC crystal alteration.

#### 5.4.4.3 Master clock alterations

Electronic equipment is often synchronized among devices by the use of an internal master clock, usually provided by crystal. In some cases these clocks are used by CPUs to establish the time base for the measured energy. Master clocks can be entry point of attackers, and can be used to change the recorded consumption by altering the crystal frequency. In Figure 5-30 a sample of this attack is given, in this case a crystal was replaced.

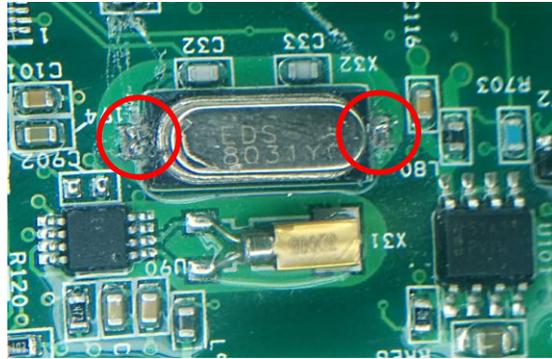


Figure 5-30 A clear sign of oscillator tampering.

#### 5.4.4.4 *Communication channel alterations*

Some brilliant attacks can be done in the communication channel mediums. For example, on AMR devices the open nature of the encoder-receiver-transmitter (ERT) radio used to obtain the meter readings made possible to send forged readings into the data concentrator. The data concentrator were often vans that were driven across cities capturing meter reading that were send on the clear at the 900-920 MHz band. A tool to capture data packets is publically available at [117].

#### 5.4.5 *Smart meter attacks-the future.*

Although smart meters are considered as the most secure devices of all meters, in practice, vulnerabilities have been found, and it is possible that in a near future network attacks will be possible. A worthy case of mention is the FBI case on meter alteration by a team of individuals on the Puerto Rico Electric Power Authority. These alterations consisted on modifying the TC parameters of the smart meters via optical port access [118]. Although in most jurisdictions meter tampering is illegal, software attacks can be mounted in a wide range of smart metering devices by employing an open source program called “termineter” [35].

Other organizations such as IOActive have raised awareness of network attack based on virus/worms architecture, which could potentially bring down the entire grid [119]. In response, NIST has approved NIST-7628-guidelines for Smart Grid Cyber Security [9].

## 5.5 A Central Observer Technique Based On the Harmonic Content, For Energy Theft Assessment

As it can be seen from the previous section, most theft-identifying methodologies use power consumption data to detect suspicious devices, with some methods requiring to quantify the amount of non-technical losses existent on the transmission line to correctly perform a power balance estimation. However, with the introduction of smart metering other variables could be exploited.

This section presents a new methodology for identifying energy theft based on the functionalities provided by a smart meter network. The developed methodology is intended for use in a custom-made meter designed to execute energy theft identification by using current balance techniques and harmonic load signature identification.

### 5.5.1 *Introduction*

Current is a basic electrical variable, and is often taught at introductory undergraduate levels, with its associated laws; one of such laws was established by Kirchhoff in 1845. This law states that “At any node in an electrical circuit, the sum of currents flowing into that node is equal to the sum of currents flowing out of that node”. Although this law could be considered trivial, it can be used to detect energy theft, since in an ideal distribution transformer the amount of current coming out of the secondary winding must be equal to the current passing through the customer’s meters. This ideal condition is likely to occur on most distribution systems, due to the minimal effects of lumped parameters on short lines, which can be neglected. This law should only be broken if a fault is present or energy-theft is occurring.

To properly use Kirchhoff law an additional requirement must be met: the instantaneous current value must be measured at the same time by all units attached to the same conductor (i.e. phase) in order to execute a current balance computation. If the prior condition is met then a simple energy-theft identifying model can be constructed (see Figure 5-31) based on the power balance diagram described on [27].

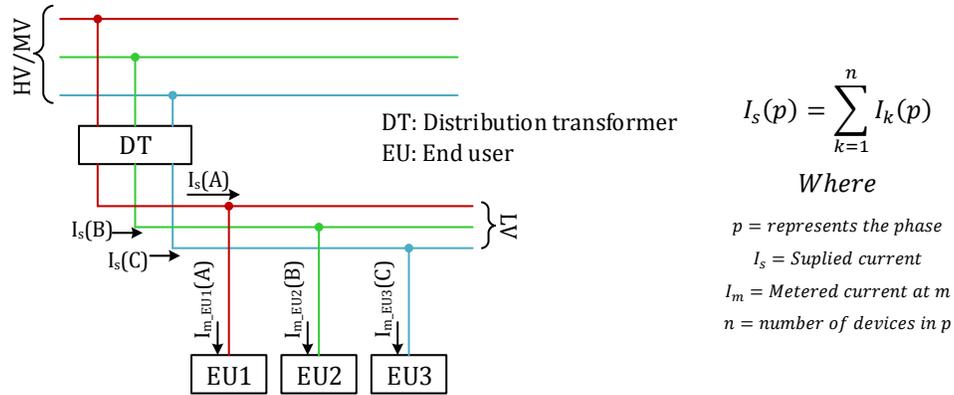


Figure 5-31. The proposed method diagram uses current balance.

Although the preceding algorithm would operate, a tolerance error  $\varepsilon$  must be inserted to account for the measurement errors introduced by the metering units installed at the end user side, on Eq. 5.2 a threshold function is given.

$$I_s(p) < \sum_{j=1}^n I_j(p) + \varepsilon \quad \text{Eq. 5.2}$$

where

$\varepsilon = \text{expected error (acumulated)}; n = \text{number of installed meters}$

### 5.5.2 Algorithm development

Eq. 5.2 provides utilities with the ability to locate problematic areas in a similar way to [27], but it does not offer the ability to pinpoint suspicious meters. In order to do this, an enhanced algorithm is needed. The enhanced algorithm is based on a load identification algorithm described on [120]; in that article, the authors demonstrate a method for identifying the presence of a load, based on the disaggregation of single point measurements by means of classification algorithms that employ harmonic content as their discriminator. According to the authors, each device contains a unique harmonic spectral fingerprint that can be used to identify the presence of a particular load in an aggregate measurement point, by means of artificial neural networks and binary classifiers.

Following the same principles, it is feasible to consider that individual users can be recognized by means of comparing their particular harmonic signatures near their origin (the meter) and

comparing it with a data aggregator (a central observer). Although the authors approach given on [120] correctly identify the presence of loads, it requires an overwhelming complexity to be implemented on a metering device, and thus in this thesis an alternative technique is proposed. This simpler technique was borrowed from the least squares implementation detailed in [3].

The resulting technique uses harmonic components to find the presence of a particular waveform in the data aggregator. For this case the signal is compared to the current difference resulting from the data aggregator minus the reported end-user measurements, as illustrated in Figure 5-32.

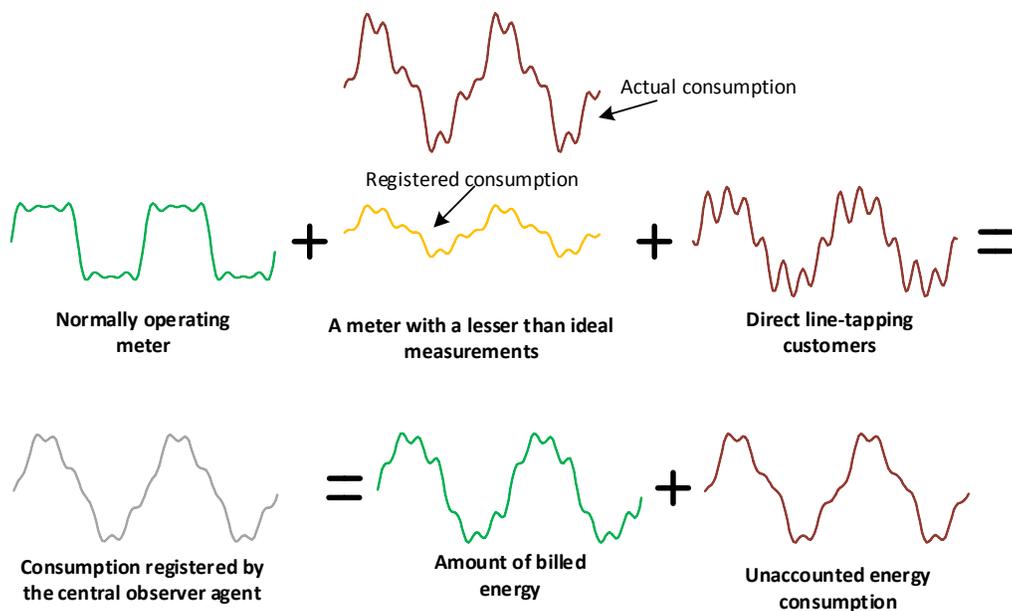


Figure 5-32. The proposed method assumes that current waveform is composed of three user types.

Figure 5-32 depicts that “The current waveform registered in a central observer unit is equal to the aggregated current reported by honest end-users, plus the current signal being measured by malfunctioning units (composed of two components: registered and unregistered), and direct-line tapping customers”. This means that the waveform signal registered by the central observer can be decomposed into two primary components: the accounted and unaccounted energy consumptions; the unaccounted component can be used to identify possible patterns coming out of malfunctioning units.

Furthermore the number of customers can be divided into two type of customers, honest and dishonest ( $m$  and  $l$ ), which although unknown can be assumed to exist in any given network. These coefficients will be used as if they were known *a priori* in the following section, however their value will be determined according to a  $\beta$  value that will be presented in the next sections.

A continuous time equation that describes the signal decomposition given by Figure 5-32 for a single-phase load is given on Eq. 5.3.

$$I_s(t) = \sum_{j=1}^m I_j(t) + \sum_{k=m+1}^{m+l} I_k(t) + DT(t) + \varepsilon = I_B(t) + I_U(t) \quad \text{Eq. 5.3}$$

$$I_s(t) = \sum_{j=1}^m I_j(t) + \sum_{k=m+1}^{m+l} (1 - \beta_k) I_k(t) + \sum_{k=m+1}^{m+l} (\beta_k) I_k(t) + DT(t) + \varepsilon = I_B(t) + I_U(t)$$

$$\vdots$$

$$I_B(t) = \sum_{j=1}^m I_j(t) + \sum_{k=m+1}^{m+l} (1 - \beta_k) I_k(t)$$

$$I_B(t) = \sum_{j=1}^m (1 - \beta_j) I_j(t) + \sum_{j=1}^m \beta_j I_j(t) + \sum_{k=m+1}^{m+l} (1 - \beta_k) I_k(t); \quad \beta_{j \rightarrow m} = 0$$

$$I_U(t) = \sum_{k=m+1}^{m+l} (\beta_k) I_k(t) + DT(t) + \varepsilon;$$

$$I_U(t) = \sum_{j=1}^m (\beta_j) I_j(t) + \sum_{k=m+1}^{m+l} (\beta_k) I_k(t) + DT(t) + \varepsilon; \quad \beta_{j \rightarrow m} = 0$$

$$I_U(t) = \sum_{x=1}^{m+l} (\beta_x) I_x(t) + DT(t) + \varepsilon; \quad \{\beta_1 \quad \beta_2 \quad \dots \quad \beta_m\} = 0 \text{ and } \{\beta_{m+1} \quad \beta_{m+2} \quad \dots \quad \beta_{m+l}\} > 0$$

where

$I_s(t)$  = Suplied current;  $I_B(t)$  = Billed current;  $I_U(t)$  = Unbilled current

$I_j(t)$  = Measured current by trusworthy meters

$I_k(t)$  = Measured current by altered/malfunctioning meters

$DT(t)$  = Unaccounted current drawn by direct tapping customers

$\beta_i$  = Alteration factor of customer  $i$

$\varepsilon$  = Measurement error

$m$  = number of trustworthy meters;  $l$  = Number of malfunctioning meters

The generalization given on Eq. 5.3 produces an obvious deduction, that the amount of unaccounted current  $I_U$  depends on the amount of malfunctioning units plus the current absorbed by direct tapping customers. This equation is the basis of the proposed method; the method employs a linear set of equations to solve the  $\beta_i$  factor associated with each meter, and thus identifies malfunctioning meters by analyzing the  $\beta_i$  factor value. Although the procedure looks simple, it is not directly applicable to real life scenarios due to the measurement errors and direct tapping customers that appear on real life (which are unaccountable), creating unknown variables that result in a linear set of underdetermined equations.

On Eq. 5.4 a matrix representation of Eq. 5.3 is given. This representation considers that the amount of direct tapping customers and  $\varepsilon$  varies with time, however this representation results impractical to solve and instead Eq. 5.5 is the basis of this work.

$$\begin{bmatrix} I_u(t_1) \\ I_u(t_2) \\ \vdots \\ I_u(t_m) \\ \vdots \\ I_u(t_{m+l}) \end{bmatrix} = \begin{bmatrix} I_1(t_1) & I_2(t_1) & \cdots & I_m(t_1) & \cdots & I_{m+l}(t_1) \\ I_1(t_2) & I_2(t_2) & \cdots & I_m(t_2) & \cdots & I_{m+l}(t_2) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1(t_m) & I_2(t_m) & \cdots & I_m(t_m) & \cdots & I_{m+l}(t_m) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1(t_{m+l}) & I_2(t_{m+l}) & \cdots & I_m(t_{m+l}) & \cdots & I_{m+l}(t_{m+l}) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \\ \vdots \\ \beta_{m+l} \end{bmatrix} + \begin{bmatrix} DT(t_1) \\ DT(t_2) \\ \vdots \\ DT(t_m) \\ \vdots \\ DT(t_{m+l}) \end{bmatrix} + \begin{bmatrix} \varepsilon(t_1) \\ \varepsilon(t_2) \\ \vdots \\ \varepsilon(t_m) \\ \vdots \\ \varepsilon(t_{m+l}) \end{bmatrix} \quad \text{Eq. 5.4}$$

where

$I_{x(t_y)}$  = Current metered by unit (x) at instant (y)

$\beta_i$  = Alteration factor of customer i

$DT_{(y)}$  = Unaccounted current drawn by direct tapping customers at instant (y)

$\varepsilon_{(t_y)}$  = Measurement error at at instant (y)

$I_U(t_y)$  = Unbilled current at instant (y)

$m$  = number of trustworthy meters;  $l$  = Number of malfunctioning meters

$$\begin{bmatrix} I_u(t_1) \\ I_u(t_2) \\ \vdots \\ I_u(t_m) \\ \vdots \\ I_u(t_{m+l}) \end{bmatrix} - \begin{bmatrix} L \\ L \\ \vdots \\ L \\ \vdots \\ L \end{bmatrix} = \begin{bmatrix} I_1(t_1) & I_2(t_1) & \cdots & I_m(t_1) & \cdots & I_{m+l}(t_1) \\ I_1(t_2) & I_2(t_2) & \cdots & I_m(t_2) & \cdots & I_{m+l}(t_2) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1(t_m) & I_2(t_m) & \cdots & I_m(t_m) & \cdots & I_{m+l}(t_m) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ I_1(t_{m+l}) & I_2(t_{m+l}) & \cdots & I_m(t_{m+l}) & \cdots & I_{m+l}(t_{m+l}) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdots \\ \beta_m \\ \cdots \\ \beta_{m+l} \end{bmatrix} \quad \text{Eq. 5.5}$$

where

$I_{x(t_y)}$  = Current metered by unit (x) at instant (y)

$\beta_i$  = Alteration factor of customer i ...  $\beta_i \geq 0$

$L$  = Unaccounted current drawn by direct tapping customers plus measurement error (considered constant)

$I_U(t_y)$  = Unbilled current at instant (y)

$m$  = number of trustworthy meters;  $l$  = Number of malfunctioning meters

Since Eq. 5.5 requires solutions of the form of  $\beta_i \geq 0$  and the  $\mathbf{I}, \mathbf{L}$  vectors contain uncertainties due to measurement errors, it cannot be solved by using traditional linear system techniques ( $\mathbf{Ax} = \mathbf{b}$ ) and thus an alternative solving technique is employed. The proposed solution converts Eq. 5.5 into an optimization problem of the form illustrated by Eq. 5.6.

$$\{\text{Maximize, Minimize}\}: \quad f(\mathbf{x}) = \mathbf{c}'\mathbf{x} \quad \text{Eq. 5.6}$$

Subject to:

$$\mathbf{Ax} \leq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}$$

where:

$$\mathbf{A} \in \mathcal{M}_{m \times n}(\mathbb{R}) \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \in \mathbb{R}^n \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$$

and:

$\mathcal{M}_{m \times n}(\mathbb{R})$  denotes the set of matrixes with  $m$  rows,  $n$  columns with real coefficients

' denotes transposition

In this case (Eq. 5.7), a maximizing function is used to search for altered units (which should exhibit high  $\beta_i$  values). Furthermore, the  $\mathbf{I}, \mathbf{L}$  vectors are reduced to a scalar  $\delta$  that multiplies the  $\mathbf{I}$  vector, in such a way it approximates the amount of energy being stolen by altered units (which should be set to 1 when no direct tapping customers are expected).

$$\text{Maximize} \quad f(\boldsymbol{\beta}) = (\beta_1 + \beta_2 + \dots + \beta_m + \dots + \beta_{m+l}) = \mathbf{c}'\boldsymbol{\beta} \quad \text{Eq. 5.7}$$

$$\text{i.e. } \mathbf{c} = \{c_1 = 1, c_2 = 1, \dots, c_{m+l} = 1\}$$

Subject to:

$$\begin{bmatrix} I_1(t_1) & I_2(t_1) & \dots & I_m(t_1) & \dots & I_{m+l}(t_1) \\ I_1(t_2) & I_2(t_2) & \dots & I_m(t_2) & \dots & I_{m+l}(t_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ I_1(t_m) & I_2(t_m) & \dots & I_m(t_m) & \dots & I_{m+l}(t_m) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ I_1(t_{m+l}) & I_2(t_{m+l}) & \dots & I_m(t_{m+l}) & \dots & I_{m+l}(t_{m+l}) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_m \\ \dots \\ \beta_{m+l} \end{bmatrix} \leq \delta \begin{bmatrix} I_u(t_1) \\ I_u(t_2) \\ \vdots \\ I_u(t_m) \\ \vdots \\ I_u(t_{m+l}) \end{bmatrix}$$

$$\beta_i \geq 0$$

where:

$$\delta \mathbf{I} \approx \mathbf{I} - \mathbf{L}$$

The solution proposed by Eq. 5.7 is solved by using the simplex method that is described in annex K of this thesis. Since linear optimizations can have a set of alternate optimal solutions [121], it is important to find all of these solutions by iteratively altering the  $\mathbf{c}$  vector according to previous optimization results in order to find all solutions until  $\mathbf{c}$  proves to be trivial ( $\mathbf{c} = \{c_1 = 0, c_2 = 0, \dots, c_m = 0\}$ ). See Figure 5-33.

In order to provide redundancy the proposed method employs two simultaneous restriction conditions that are solved for each harmonic component ( $I^H, H = 1 \text{ and } H = 3$ ). This enables the optimization process to filter false positives.

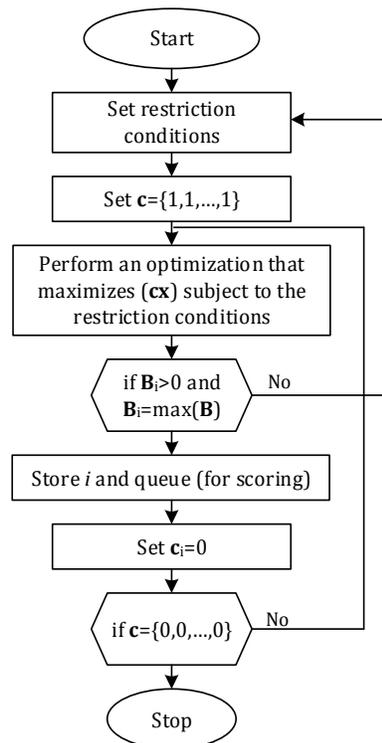


Figure 5-33 Modified Simplex algorithm used to obtain the  $\beta_i$  factors

Since the simplex algorithm will create optimal solutions that involve each of the  $\beta_i$  factors contained in the  $\beta$  vector, there must be a way of identifying altered  $i$  units by using some probabilistic analysis. This is accomplished by creating a table that contains all of the simplex algorithm solutions under different restriction conditions (using a variety of network operating states) and performing a weighted ordering process. The chosen ordering method uses the queue generated during the simplex solution to assign a weighting factor to each unit ( $i$ ), and ordering them in ascending order (see Table 5.2 and Table 5.3).

Table 5.2. Sample output from the simplex method for several rounds.

	Score (queue order)				
	1	2	3	4	5
Round 0	5	3	4	2	1
Round 1	3	5	4	1	2
Round 2	5	4	3	2	1
Round 3	4	5	3	2	1
Round 4	3	5	4	1	2

Table 5.3. Computed score from the sample given in Table 5.2

$i$	Score
5	8
3	10
4	12
2	22
1	23

With the known  $i$  positions (choosing those with the least score) a modified matrix can be assembled that reflects those users that did not show up during the optimization process by assigning them a  $\beta_i=0$  value (see Eq. 5.8). The new system can be solved by linear squares to find approximate solutions to the alteration factors and thus be able to identify altered units.

$$\begin{bmatrix} I_{1(t_1)} & I_{2(t_1)} & \cdots & \cdots & \cdots & I_{m(t_1)} & \cdots & I_{m+s(t_1)} & \cdots & I_{m+l(t_1)} \\ I_{1(t_2)} & I_{2(t_2)} & \cdots & \cdots & \cdots & I_{m(t_2)} & \cdots & I_{m+s(t_2)} & \cdots & I_{m+l(t_2)} \\ 0 & 0 & 1 & \cdots & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots \\ I_{1(t_m)} & I_{2(t_m)} & \cdots & \cdots & \cdots & I_{m(t_m)} & \cdots & I_{m+s(t_m)} & \cdots & I_{m+l(t_m)} \\ \cdots & \cdots \\ \cdots & 1 & \cdots & 0 \\ \cdots & \cdots \\ I_{1(t_{m+l})} & I_{2(t_{m+l})} & \cdots & \cdots & \cdots & I_{m(t_{m+l})} & \cdots & I_{m+s(t_{m+l})} & \cdots & I_{m+l(t_{m+l})} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \vdots \\ \beta_m \\ \vdots \\ \beta_{m+s} \\ \vdots \\ \beta_{m+l} \end{bmatrix} = \delta \begin{bmatrix} I_{u(t_1)} \\ I_{u(t_2)} \\ 0 \\ 0 \\ \vdots \\ I_{u(t_m)} \\ \vdots \\ 0 \\ \vdots \\ I_{u(t_{m+l})} \end{bmatrix} \quad \text{Eq. 5.8}$$

where:

$\beta_3, \beta_4, \beta_{m+s}$  did not show up during the first  $p$  steps of the optimization process

### 5.5.3 Simulation preamble

In order to validate the previously proposed algorithm, a computer simulation was developed to evaluate the theft detection characteristics of the proposed algorithm. This model was constructed for a single-phase low voltage distribution network, which has a variable number of trustworthy customers with a different number of malfunctioning meters simultaneously deployed; in order to create a more complex model, a variant number of direct-wiretapping customers were inserted.

For each customer (legal or not) a unique load pattern was assigned based on the supplied current ( $I$ ) and P.F readings that were obtained in a distribution feeder in southern Mexico [122]. The load profile can be seen in Figure 5-34 (reported at 10 minute intervals). For this particular case, the current load profile is expressed in terms of the maximum current demand.

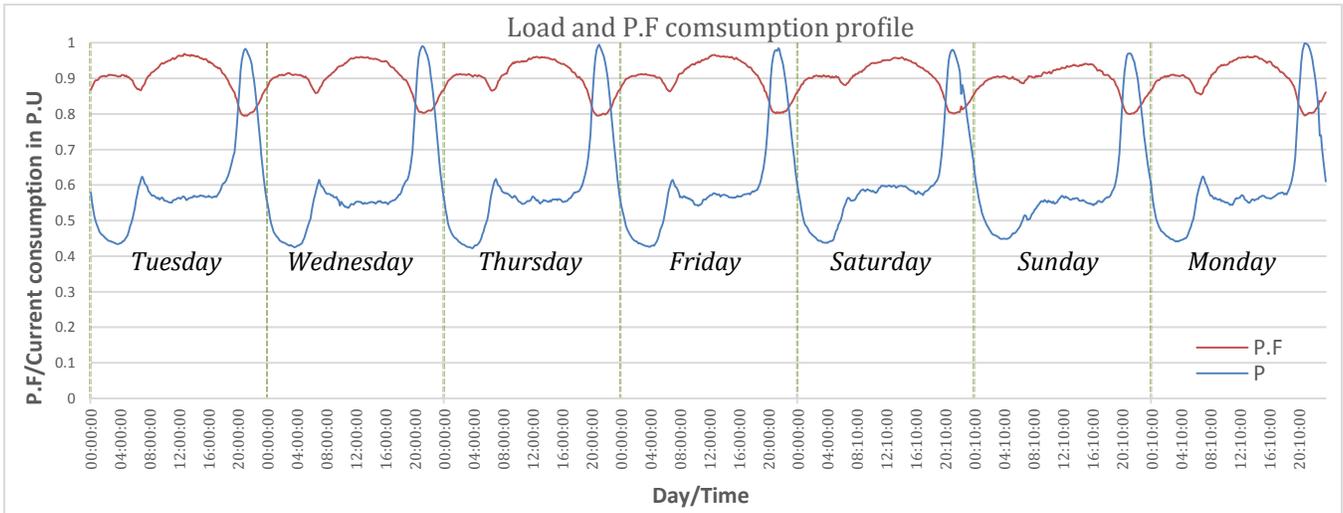


Figure 5-34 Weekly load profile obtained from a feeder in a southern state in Mexico.

An individual current profile was created by adjusting the base demand profile plus a random on/off state that mimics the behavior of equipment operations. Also for each customer the maximum current drawn was assigned randomly, with minimum idle state set to 0.25 Amperes. The final profile is shown in Figure 5-35 (for three sample customers).

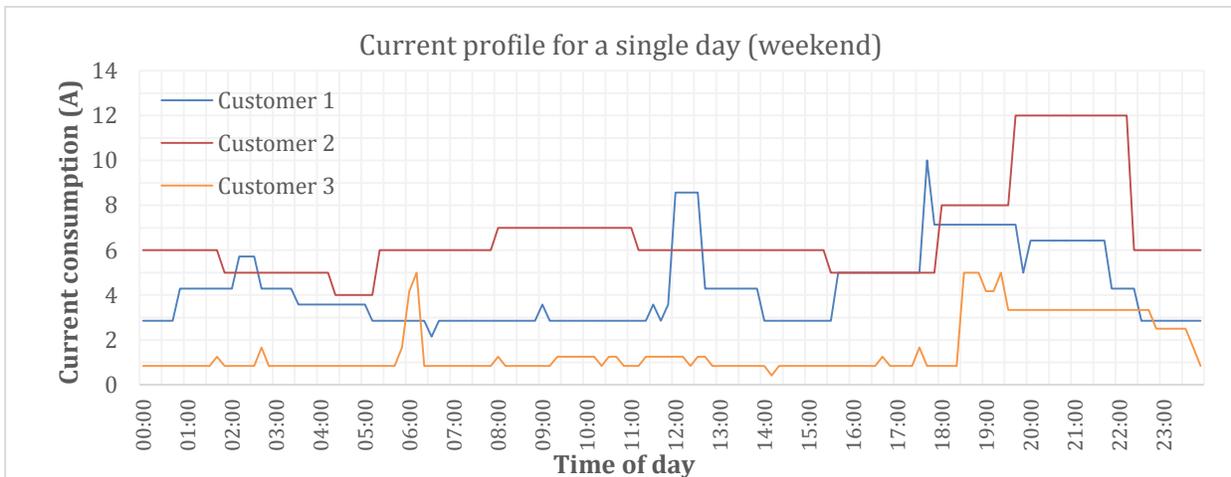


Figure 5-35 Sample daily current profile for three customers, with different average current consumption.

Simultaneously a unique power factor was created for each customer. This P.F profile was created by inserting small random variations variations (using a normal distribution with  $\mu = \text{feeder PF}$  and  $\sigma = .033$ ) and applying smoothing filters built into Matlab™. The result can be observed in Figure 5-36.

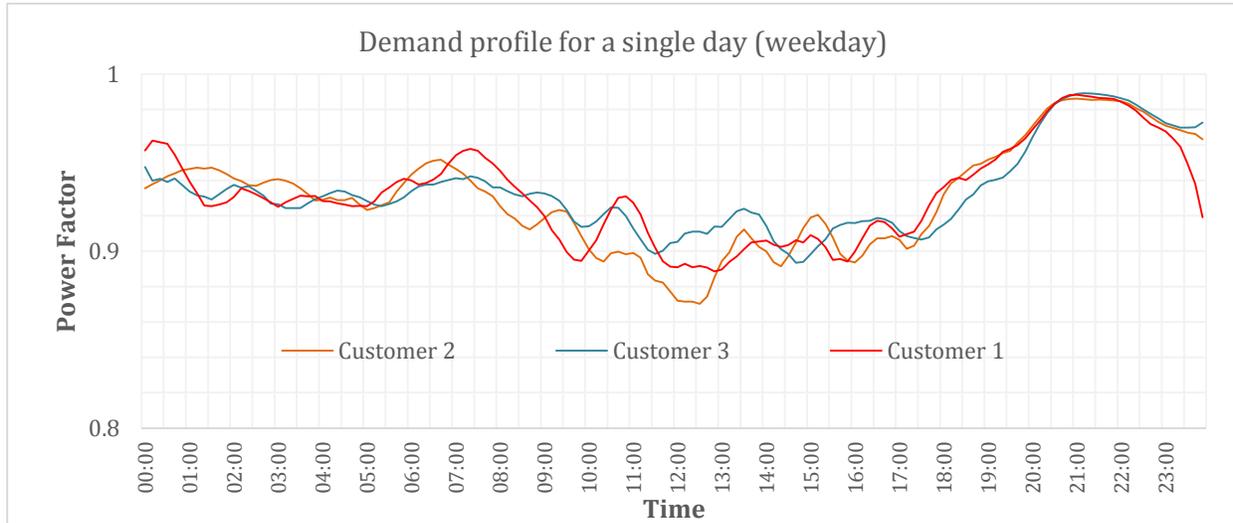


Figure 5-36 Sample daily P.F profile for three customers, where the existence of unique profiles can be observed.

After the unique current load pattern is obtained per customer, a harmonic content profile is created, according to the current load and P.F characteristics. The created harmonic patterns exhibit slow changing patterns during stable load conditions and fast dynamics during fast load changes. The harmonic profiles are computed for the first three odd harmonics, and are checked for concordance with individual user P.F profiles (using the standard P/S method, but not the one proposed by IEEE 1459).

The generated harmonics profiles are based on the harmonic survey done by authors in [123]. The survey contains the amount of harmonics with respect to the fundamental frequency in common office/household environment and is summarized in Table 5.4. These raw results can be used to calculate the average and standard deviation ( $\sigma$ ) properties of the harmonic signals found on a purely non-linear load environment. Although the results are usable, they offer misleading results in the generalization of real-life harmonic profiles, and thus a 50% mix of linear and non-linear load

characteristics was used to generate the unique harmonic profile per customer (by employing half of the values reported as the average and  $\sigma$ ).

Table 5.4. Common harmonic components found of common household/office environments, adapted from [123].

Device	3 <sup>rd</sup> (%F <sub>1</sub> )	5 <sup>th</sup> (%F <sub>1</sub> )	7 <sup>th</sup> (%F <sub>1</sub> )
Fluorescent Lamp	10.7	2.0	1.8
Freezer	11.0	4.7	11.0
Amplifier	32.1	30.7	14.2
Television	55.1	36.8	20.3
Photocopier	37.7	40	30.0
Laptop	49.6	43.8	36.2
PC	52.8	43.5	31.6
Printer	46.7	41.3	36.2
Average	36.9625	30.35	22.6625
$\sigma$	17.8176	17.1972	12.8101

These unique harmonic profiles remain almost constant through the different days, since it is unlikely that customers change their appliances that often. The generated harmonic profiles for a single customer can be observed in Figure 5-37 (plotted on a logarithmic scale). Similarly, in Figure 5-38 the angle components are plotted for the same user, although no information was found regarding the harmonic angle behavior on domestic loads.



Figure 5-37 Sample customer current components for a single day current profile.

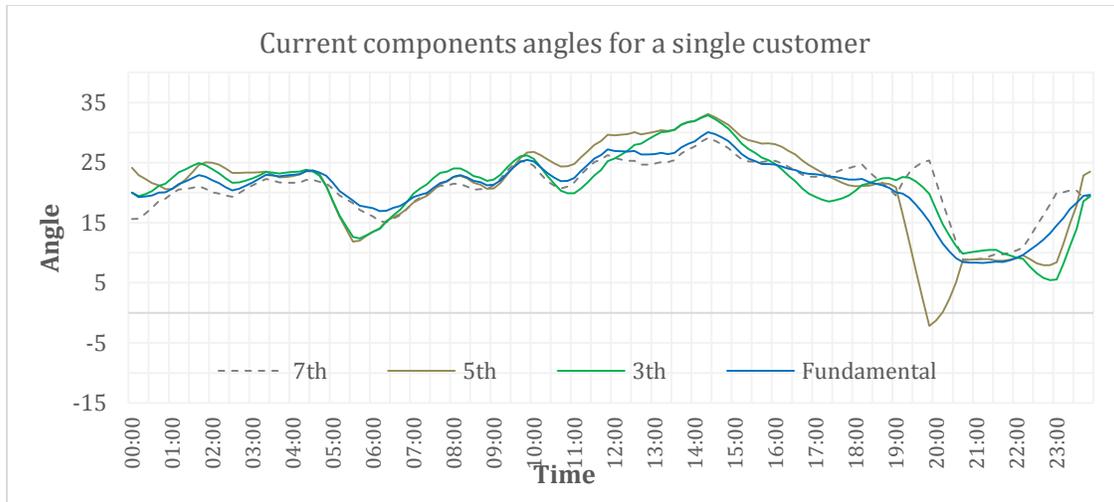


Figure 5-38 Sample customer current angles during a single day.

### 5.5.3.1 Energy theft simulation procedure

Once the harmonic and current demand profiles were generated for each customer, the actual energy theft simulation began. This was done by interpolating the different customer profiles at 1-minute intervals. At each minute interval, a unique waveform pattern was generated for each customer based on the current demand and harmonic content. A sample of the generated waveform data for two customers is given in Figure 5-39.

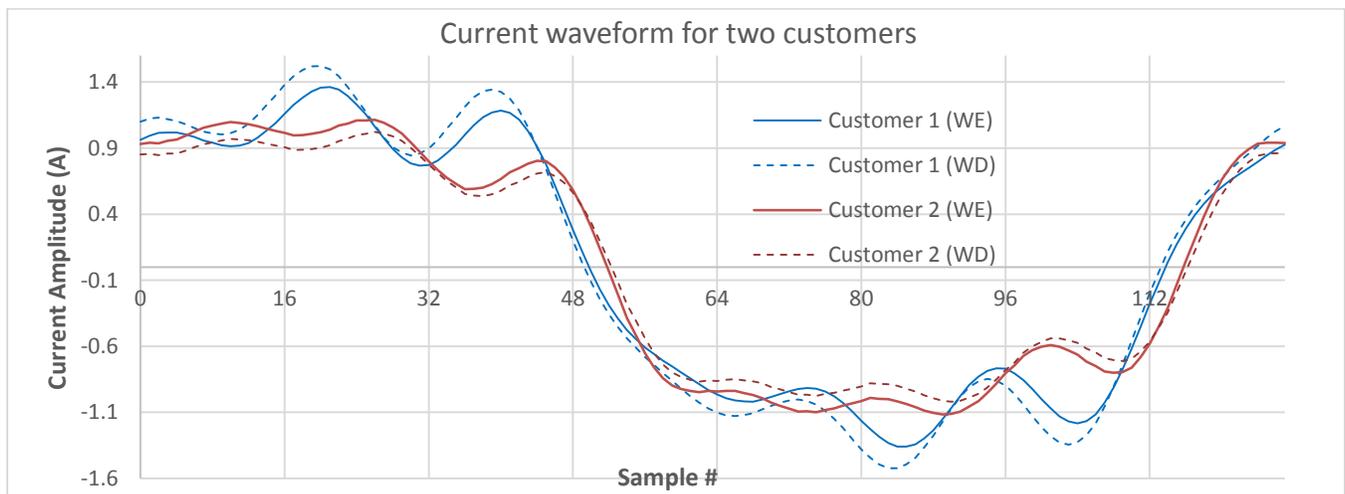


Figure 5-39 Current waveform for two customers in two different days.

The before mentioned waveform was further processed to simulate the digital acquisition (signal degradation) process that occurs on digital meters, where a fixed measurement error value was

assigned to each meter, plus a time variant error that simulates the ADC quantization error and noise effects. The total error was designed to represent the behavior of a 30-Ampere capacity watt-hour meter, which is usually tested under a 2.5-Ampere current, and is rated for a class 0.5% accuracy according to IEEE C12 standard. The result of such ADC simulation is shown in Figure 5-40.

At this step some of the meters were assigned with a  $\beta$  alteration factor that mimics the presence of a current bypass mechanism, this creates a meter that registers a lesser than real consumption, but does not affect the readings recorded by the central observer, thus creating an amount of unbilled current ( $I_u$ ). For some cases, additional unmeasured clients were also incorporated into this result.

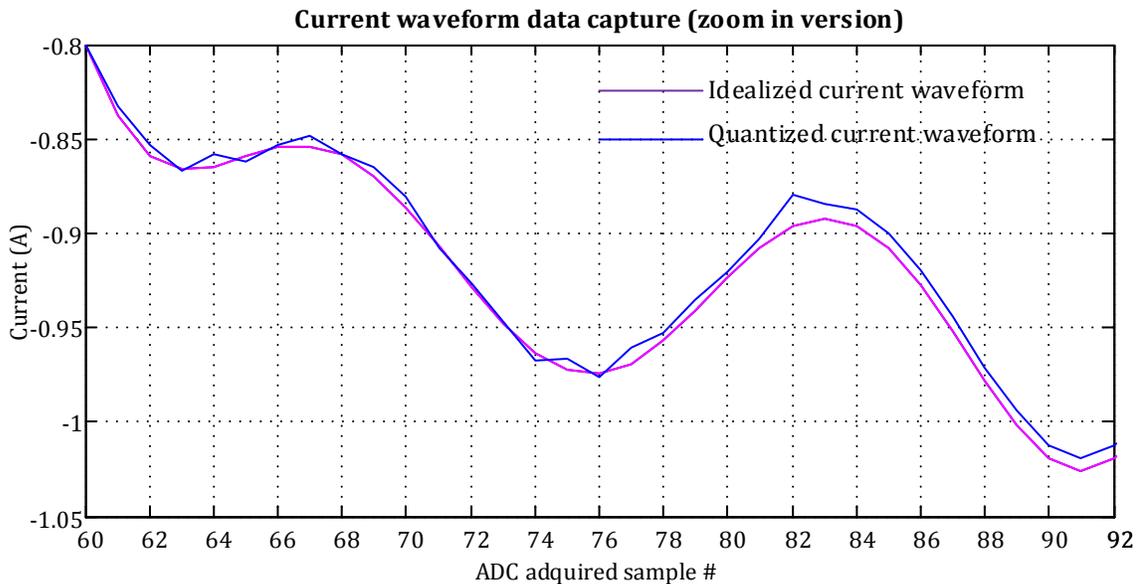


Figure 5-40 Current waveform data capture, with added noise typical of a class 0.5% precision device (with a 2.5A testing current).

Once all of the meters readings were transformed into a time domain representation, two matrixes are used to obtain suspected meter indices by using the simplex method. The restriction matrix is assembled according to the harmonic contents of the waveform expressed in a rectangular form ( $a + jb$ ). A new row is appended to the matrix structure every minute, and is queue for solution until the number of equations is equal to the number of installed meters (Normally operating ( $m$ ) + altered units ( $l$ )) (see Eq. 5.9).

Maximize 
$$f(\boldsymbol{\beta}^H) = (\beta^H_1 + \beta^H_2 + \dots + \beta^H_m + \dots + \beta^H_{m+l}) = \mathbf{c}\boldsymbol{\beta}^H \quad \text{Eq. 5.9}$$

Subject to:

$$\begin{bmatrix} I^H_{1(t_1)} & I^H_{2(t_1)} & \dots & I^H_{m(t_1)} & \dots & I^H_{m+l(t_1)} \\ I^H_{1(t_2)} & I^H_{2(t_2)} & \dots & I^H_{m(t_2)} & \dots & I^H_{m+l(t_2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ I^H_{1(t_m)} & I^H_{2(t_m)} & \dots & I^H_{m(t_m)} & \dots & I^H_{m+l(t_m)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ I^H_{1(t_{m+l})} & I^H_{2(t_{m+l})} & \dots & I^H_{m(t_{m+l})} & \dots & I^H_{m+l(t_{m+l})} \end{bmatrix} \begin{bmatrix} \beta^H_1 \\ \beta^H_2 \\ \dots \\ \beta^H_m \\ \dots \\ \beta^H_{m+l} \end{bmatrix} \leq \delta \begin{bmatrix} I^H_{u(t_1)} \\ I^H_{u(t_2)} \\ \vdots \\ I^H_{u(t_m)} \\ \vdots \\ I^H_{u(t_{m+l})} \end{bmatrix}$$

$$\beta^H_i \geq 0$$

where

$$H = \text{Harmonic component } \{1,3\}$$

Once an optimal solution is found for each harmonic component, a single  $i$  unit is chosen and queued until the  $c$  vector contains only zeros. These steps are repeated several times until a sufficient amount of data is available for analysis. For the reported cases in this chapter a total of  $3(m + l)$  network operating conditions are analyzed to compute a table similar to Table 5.3. After this step, a solution of vector  $\beta$  is computed as per Eq. 5.8 and averaged multiple times to compute a final  $\beta$  vector that it is used to determine altered units.

#### 5.5.4 *Simulation Results*

The aforementioned simulation procedure was used to evaluate the energy theft characteristics of the proposed method under different scenarios. For each of the proposed cases a fixed number of registered clients (those with an installed meter) are assigned to a single-phase low voltage distribution transformer. Then, a variable number of direct tapping customers are attached to the transformer terminals, creating different levels of energy theft that are discussed in detail under each scenario.

For cases 1-3, tables 5.3, 5.4 and 5.5 are used to report the  $\beta_i$  factors, the columns indicate the number of direct wiretapping users attached to the distribution transformer, the framework developed for this test, creates a unique but repeatable network structure that depends on the total number of customers (metered + unmetered). It works under the same principals of a PRNG, and thus enables to create repeatable networks configurations that can be fed with different load profiles. Thus, the networks associated with each column are unique (harmonic profile, P.F, current demand) and are only repeatable, if the same number of total customers is selected as its seed value.

This means that each column must be treated as a different network unless the total number of customers is equal (i.e. columns can only be compared across different days only if they contain the same number of total users)

The total number of altered meters ( $A_U$ ) is calculated by multiplying the number of altered meters per network and the total number of simulated networks (that depends on the number of columns that are present for single day load profile)

The effective detection rate is computed by Eq. 5.10

$$Ed = \left(1 - \frac{F_N}{A_U}\right) * 100 \quad \text{Eq. 5.10}$$

where

$F_N = \text{False negatives (units that are altered but are reported as beign ok)}$

#### 5.5.4.1 Case 1. More than 20 users connected to a monophasic transformer, with three altered units.

The first simulated case considers that there are 20 registered customers connected to the distribution transformer, with three of these customers having an altered meter. The raw results for this case are shown in

Table 5.5 (with the  $\beta_i \leq 0$  set to zero). For this table two different day profiles are simulated, at each column a different number of direct tapping customers is considered. The two-day window is intentionally used to filter false positives (units that are ok but are reported as tampered), although some false negatives can arise (units that are malfunctioning but are reported as ok). In this table the altered units are highlighted in yellow, while the red cells indicate that the threshold level has been reached. The heuristic interpretation for this case is given in Table 5.6

Table 5.5. Reported suspicion factors, for a 20+ users connected to a single phase.

Property	WEEKDAY LOAD PROFILE						WEEKEND LOAD PROFILE					
	20	20	20	20	20	20	20	20	20	20	20	20
Customers with meters	20	20	20	20	20	20	20	20	20	20	20	20
Altered units	3	3	3	3	3	3	3	3	3	3	3	3
#Direct-tapping users	5	4	3	2	1	0	5	4	3	2	1	0
Meter 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1263	0.0628	0.0000	0.3745	0.0000
Meter 2	0.0000	0.0000	0.0162	0.0000	0.0000	0.0000	0.0000	0.1473	0.0000	0.0000	0.0000	0.0151
Meter 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0197	0.0000	0.1332	0.0000	0.2510	0.0000	0.0000
Meter 4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0538	0.0000	0.0000	0.0000
Meter 5	0.0000	0.0000	0.1239	0.1857	0.0000	0.0666	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 6 ( $\beta = 0.6$ )	0.4916	1.0000	0.8034	1.0000	0.1521	1.0000	0.4707	1.0000	1.0000	1.0000	0.2493	1.0000
Meter 7	0.0000	0.0000	0.0140	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.6382	0.0000
Meter 8	0.0000	0.2253	0.0000	0.0000	0.2906	0.0000	0.0000	0.0686	0.0000	0.0150	0.0090	0.0000
Meter 9	0.2515	0.0000	0.0000	0.0000	0.0524	0.0000	0.0000	0.0000	0.0000	0.0000	0.2610	0.0000
Meter 10	0.0000	0.0907	0.0000	0.0000	0.3906	0.1680	0.3703	0.0000	0.1335	0.0000	0.0000	0.0002
Meter 11	0.0000	0.0000	0.0000	0.0459	0.0000	0.0000	0.0000	0.0000	0.0000	0.0909	0.0000	0.0000
Meter 12 ( $\beta = 0.5$ )	0.3539	0.1829	0.0677	0.0000	0.7190	0.4618	0.4532	0.7140	0.0787	0.1454	1.0000	0.0735
Meter 13	0.1694	0.0000	0.0000	0.0000	0.0000	0.0653	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 14	0.0000	0.0000	0.0000	0.6318	0.0000	0.0000	0.0000	0.0000	0.0717	0.2178	0.0000	0.0000
Meter 15	0.0000	0.0931	0.0000	0.0000	0.0000	0.0485	0.0000	0.0000	0.0000	0.0000	0.0000	0.1033
Meter 16	0.0000	0.0000	0.0000	0.0000	0.1752	0.0000	0.0555	0.0000	0.0000	0.0928	0.0000	0.0000
Meter 17	0.5858	0.0000	0.0000	0.2325	0.0000	0.0079	0.0000	0.0000	0.0000	0.0000	0.0000	0.0886
Meter 18 ( $\beta = 0.4$ )	1.0000	0.3000	1.0000	0.5603	1.0000	0.0762	1.0000	0.9681	0.0976	0.0590	0.3747	0.1275
Meter 19	0.0402	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0025	0.1700	0.0000	0.0000	0.0000
Meter 20	0.0000	0.0000	0.0000	0.0000	0.0000	0.1496	0.0000	0.3443	0.0898	0.0000	0.0000	0.0576
Energy Losses (PU)	0.2550	0.1873	0.1817	0.1494	0.0902	0.0676	0.2503	0.1879	0.1820	0.1497	0.0905	0.0676
Avg Value	0.1446	0.0946	0.1013	0.1328	0.1390	0.1032	0.1179	0.1752	0.0878	0.0935	0.1453	0.0732

\*Numbers on RED indicate false positives, while GREEN signify false negatives.

By analyzing the heuristic results presented in Table 5.6, the number of false negatives can be estimated to represent 22.3% of the altered units (obtained by dividing the number false negatives vs the number of altered units)(each column is counted as a different network). These rates could be improved by using more days to assess the overall meter suspicion factors (see section 5.5.4.3)

Table 5.6. Interpreted suspicion factors, for a 20+ users connected to a single phase, (77.7% effective detection rate)

Property	INTERPRETATION					
	20	20	20	20	20	20
Customers with meters	20	20	20	20	20	20
Altered units	3	3	3	3	3	3
#Direct-tapping users	5	4	3	2	1	0
Meter 1	OK	OK	OK	OK	OK	OK
Meter 2	OK	OK	OK	OK	OK	OK
Meter 3	OK	OK	OK	OK	OK	OK
Meter 4	OK	OK	OK	OK	OK	OK
Meter 5	OK	OK	OK	OK	OK	OK
Meter 6 ( $\beta = 0.6$ )	SUSP	SUSP	SUSP	SUSP	SUSP	SUSP
Meter 7	OK	OK	OK	OK	OK	OK
Meter 8	OK	OK	OK	OK	OK	OK
Meter 9	OK	OK	OK	OK	OK	OK
Meter 10	OK	OK	OK	OK	OK	OK
Meter 11	OK	OK	OK	OK	OK	OK
Meter 12 ( $\beta = 0.5$ )	SUSP	SUSP	OK	OK	SUSP	SUSP
Meter 13	OK	OK	OK	OK	OK	OK
Meter 14	OK	OK	OK	SUSP	OK	OK
Meter 15	OK	OK	OK	OK	OK	OK
Meter 16	OK	OK	OK	OK	OK	OK
Meter 17	OK	OK	OK	OK	OK	OK
Meter 18 ( $\beta = 0.4$ )	SUSP	SUSP	SUSP	OK	SUSP	OK
Meter 19	OK	OK	OK	OK	OK	OK
Meter 20	OK	OK	OK	OK	OK	OK

\*Situations on RED indicate false positives, while GREEN signify false negatives.

5.5.4.1 Case 2. More than 15 users connected to a monophasic transformer, with two altered units.

The second case considers that there are 15 registered customers connected to the distribution transformer, with two of these customers having an altered unit. The raw results for this case are shown in Table 5.7 (with the  $\beta_i \leq 0$  set to zero). For generating this table two different day profiles were simulated; at each column a different number of direct tapping customers is considered. The two-day window is used to filter false positives. In this table the altered units are highlighted in yellow, while the red cells indicate that the threshold level has been reached.

Table 5.7. Reported suspicion factors, for a 15+ users connected to a single phase, and two altered units.

Property	WEEKDAY LOAD PROFILE						WEEKEND LOAD PROFILE					
	15	15	15	15	15	15	15	15	15	15	15	
Customers with meters	15	15	15	15	15	15	15	15	15	15	15	
Altered units	2	2	2	2	2	2	2	2	2	2	2	
#Direct-tapping users	5	4	3	2	1	0	5	4	3	2	1	0
Meter 1	0.0000	0.1698	0.6358	0.0000	0.1948	0.0000	0.0000	0.0000	0.6459	0.0000	0.0000	0.0000
Meter 2	0.0000	0.2707	0.0000	0.0000	0.0000	0.0024	0.2905	0.0000	0.0000	0.0000	0.0000	0.5269
Meter 3	0.0377	0.1241	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 4	0.0000	0.0000	0.0000	0.1609	0.0000	0.0183	0.0000	0.3843	0.0000	0.9077	0.0000	0.4628
Meter 5	0.0000	0.1969	0.1868	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 6	0.0000	0.0000	0.0000	0.3078	0.2766	0.0000	0.0000	0.1474	0.0000	0.0376	0.7847	0.0000
Meter 7( $\beta = 0.6$ )	1.0000	1.0000	0.7258	1.0000	1.0000	0.3263	0.1738	0.1458	1.0000	1.0000	0.0000	0.6649
Meter 8	0.5612	0.0000	0.0000	0.0000	0.0000	0.0000	0.1543	0.0000	0.1848	0.3231	0.0000	0.0000
Meter 9	0.0000	0.0000	0.0000	0.0000	0.0703	0.0000	0.1037	0.3932	0.0000	0.0000	0.0000	0.0000
Meter 10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0146	0.0000	0.0000	0.0000	0.0000
Meter 11	0.0000	0.0000	1.0000	0.0000	0.0000	0.0011	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 12	0.0000	0.0000	0.0000	0.0000	0.1366	0.4805	0.0000	0.0000	0.0000	0.0000	0.6355	0.0000
Meter 13	0.3729	0.0000	0.0000	0.0000	0.0000	0.2275	0.0000	0.0000	0.0000	0.0000	0.0000	0.0055
Meter 14( $\beta = 0.5$ )	0.5038	0.5565	0.2495	0.9423	0.1287	1.0000	1.0000	1.0000	0.4242	0.7587	0.8501	1.0000
Meter 15	0.3503	0.0000	0.0000	0.0130	0.1013	0.0496	0.2879	0.0000	0.0000	0.0596	1.0000	0.0000
Energy Losses (PU)	0.3064	0.2051	0.1748	0.1626	0.1351	0.0415	0.3045	0.2061	0.1757	0.1632	0.0905	0.0419
Avg Value	0.1884	0.1545	0.1865	0.1616	0.1272	0.1404	0.1340	0.1390	0.1503	0.2058	0.1352	0.1773

\*Numbers on RED indicate false positives, while GREEN signify false negatives.

The heuristic interpretation for this case is given in Table 5.8. By analyzing the heuristic results presented in Table 5.8, it can be observed that the number of false negatives in this case is 8.4%. Although no former analysis was done, this could be due to reduced number of metering units (compared with case #1) that enables to compute more simplex solutions per day, and results in a  $\beta$  vector that contains more solutions, increasing the overall confidence level.

Table 5.8. Interpreted suspicion factors, for a 15+ users connected to a single phase, and two altered units (91.6% effective detection).

Property	INTERPRETATION					
Customers with meters	15	15	15	15	15	15
Altered units	2	2	2	2	2	2
#Direct-tapping users	5	4	3	2	1	0
Meter 1	OK	OK	SUSP	OK	OK	OK
Meter 2	OK	OK	OK	OK	OK	OK
Meter 3	OK	OK	OK	OK	OK	OK
Meter 4	OK	OK	OK	OK	OK	OK
Meter 5	OK	OK	OK	OK	OK	OK
Meter 6	OK	OK	OK	OK	SUSP	OK
Meter 7 ( $\beta = 0.6$ )	SUSP	SUSP	SUSP	SUSP	OK	SUSP
Meter 8	SUSP	OK	OK	OK	OK	OK
Meter 9	OK	OK	OK	OK	OK	OK
Meter 10	OK	OK	OK	OK	OK	OK
Meter 11	OK	OK	OK	OK	OK	OK
Meter 12	OK	OK	OK	OK	SUSP	OK
Meter 13	OK	OK	OK	OK	OK	OK
Meter 14 ( $\beta = 0.5$ )	SUSP	SUSP	SUSP	SUSP	SUSP	SUSP
Meter 15	SUSP	OK	OK	OK	OK	OK

\*Situations on RED indicate false positives, while GREEN signify false negatives.

5.5.4.2 Case 3. More than 15 users connected to a single phase, with one altered unit.

Case #3 considers that there are 15 registered customers connected to the distribution transformer, with a single customer having an altered unit. The raw results for this case are shown in Table 5.9 (with the  $\beta_i \leq 0$  set to zero). Similarly to the previous cases the factors are grouped according to the number of direct tapping customers, with a two-day window employed to filter false positives. The highlighted row identifies the altered unit, while the red cells indicate that the threshold level has been reached. The heuristic interpretation for this case is given in Table 5.10

Table 5.9. Reported suspicion factors, for a 15+ users connected to a single phase, and a single altered unit.

Property	WEEKDAY LOAD PROFILE						WEEKEND LOAD PROFILE					
	15	15	15	15	15	15	15	15	15	15	15	
Customers with meters	15	15	15	15	15	15	15	15	15	15	15	
Altered units	1	1	1	1	1	1	1	1	1	1	1	
#Direct-tapping users	5	4	3	2	1	0	5	4	3	2	1	0
Meter 1	0.0000	0.1722	0.5378	0.0000	0.1914	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
Meter 2	0.0000	0.2799	0.0000	0.0301	0.0000	0.0267	0.3030	0.0213	0.0335	0.0892	0.0000	1.0000
Meter 3	0.0195	0.1442	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 4	0.0000	0.0000	0.0000	0.1224	0.0000	0.0300	0.0000	0.3865	0.0000	1.0000	0.0000	0.8659
Meter 5	0.0000	0.2214	0.1637	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Meter 6	0.0000	0.0000	0.0000	0.2466	0.2531	0.0000	0.0000	0.1888	0.0000	0.0947	0.1681	0.0000
Meter 7	0.0226	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2567	0.0000	0.0000
Meter 8	0.2927	0.0000	0.0000	0.0000	0.0000	0.0000	0.1724	0.0000	0.3347	0.4201	0.0000	0.0000
Meter 9	0.0000	0.0000	0.0000	0.0000	0.0777	0.0000	0.1125	0.3779	0.0000	0.0000	0.0000	0.0000
Meter 10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0507	0.0000	0.0000	0.0000	0.0000
Meter 11	0.0000	0.0000	0.8405	0.0000	0.0000	0.0131	0.0000	0.0000	0.0285	0.0000	0.0000	0.0000
Meter 12	0.0000	0.0000	0.0000	0.0000	0.1256	0.4971	0.0329	0.0000	0.0000	0.0000	0.1447	0.0000
Meter 13	0.1941	0.0000	0.0000	0.0000	0.0000	0.2444	0.0000	0.0000	0.0000	0.0000	0.0000	0.0365
Meter 14	0.0000	0.0786	0.0000	0.1563	0.0890	0.4570	0.0542	0.0000	0.9942	0.0933	0.0023	0.4026
Meter 15 ( $\beta = 0.6$ )	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1777	0.8752	1.0000	0.7371
Energy Losses (PU)	0.2942	0.1978	0.1612	0.1160	0.0950	0.0454	0.2923	0.1992	0.1622	0.1168	0.0955	0.0457
Avg Value	0.1019	0.1264	0.1695	0.1037	0.1158	0.1512	0.1117	0.1350	0.1712	0.1886	0.1352	0.2028

\*Numbers on RED indicate false positives, while GREEN signify false negatives.

By analyzing the heuristic results presented in Table 5.10, it can be observed that the number of false negatives has decreased to 0.0%, which can be attributed to the lower count of altered units.

Table 5.10. Interpreted suspicion factors, for a 15+ users connected to a single phase, and a single altered unit (100% effective detection).

Property	INTERPRETATION					
Customers with meters	15	15	15	15	15	15
Altered units	1	1	1	1	1	1
#Direct-tapping users	5	4	3	2	1	0
Meter 1	OK	OK	SUSP	OK	OK	OK
Meter 2	OK	OK	OK	OK	OK	OK
Meter 3	OK	OK	OK	OK	OK	OK
Meter 4	OK	OK	OK	SUSP	OK	OK
Meter 5	OK	OK	OK	OK	OK	OK
Meter 6	OK	OK	OK	OK	SUSP	OK
Meter 7 ( $\beta = 0.6$ )	OK	OK	OK	OK	OK	OK
Meter 8	SUSP	OK	OK	OK	OK	OK
Meter 9	OK	OK	OK	OK	OK	OK
Meter 10	OK	OK	OK	OK	OK	OK
Meter 11	OK	OK	OK	OK	OK	OK
Meter 12	OK	OK	OK	OK	SUSP	OK
Meter 13	OK	OK	OK	OK	OK	OK
Meter 14 ( $\beta = 0.5$ )	OK	OK	OK	OK	OK	SUSP
Meter 15	SUSP	SUSP	SUSP	SUSP	SUSP	SUSP

\*Situations on RED indicate false positives, while GREEN signify false negatives.

#### 5.5.4.3 Case 4. More than 45 users connected to a single phase, with 5 altered units

Finally, on case #4, 45 registered customers are connected to the distribution transformer, with five customers having an altered unit. The raw results for this case are shown in Table 5.11. In this case a seven-day window employed to filter false positives. The highlighted rows identifies the altered units, while the red cells indicate that the threshold level has been reached, the heuristic interpretation for this case is given on the load info for Table 5.11

By analyzing the heuristic results presented in Table 5.11 it can be observed that the number of false negatives obtained was 0.0% even if the number of clients is significantly increased. This was done by using more days to filter out false positives (7 days).

Table 5.11. Reported suspicion factors, for a 45 legally connected users plus 15 wire tappers to a single phase.

Property	WEEKDAY LOAD PROFILE							LOAD INFO			
	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Number of reports	Avg. I	Min I	Max I
Meter 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	4.3482	3.2011	7.0212
Meter 2	0.0000	0.0000	0.0000	0.0333	0.0180	0.0000	0.0846	0	4.6727	3.4494	7.3065
Meter 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	4.5529	3.3817	7.1681
Meter 4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	4.9278	3.6208	7.7159
Meter 5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1811	1	3.6368	2.6797	5.6739
Meter 6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0123	0.0000	0	5.1402	3.7714	8.1327
Meter 7	0.3608	0.0032	0.3292	0.0000	0.0000	0.0000	0.2684	3	2.2361	1.6584	3.5437
Meter 8	0.0000	0.0000	0.0000	0.0085	0.0000	0.0428	0.0000	0	5.2668	3.8793	8.0390
Meter 9( $\beta = 0.65$ )	0.5691	1.0000	0.7557	1.0000	1.0000	0.7798	1.0000	7	2.4217	1.7907	3.5942
Meter 10	0.0080	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	5.0312	3.7443	7.9190
Meter 11	0.0000	0.0000	0.0000	0.1761	0.0000	0.0000	0.0000	1	6.4970	4.8054	10.2162
Meter 12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	5.6159	4.1797	8.9991
Meter 13	0.0000	0.0000	0.0000	0.0000	0.0879	0.0000	0.0651	1	4.3741	3.2109	6.9983
Meter 14	0.0000	0.0325	0.2661	0.0000	0.0313	0.0951	0.0000	2	2.5817	1.9008	4.1050
Meter 15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	3.5358	2.6004	5.5032
Meter 16	0.0000	0.0014	0.0000	0.0308	0.0000	0.0000	0.0000	0	5.5480	4.0690	8.7749
Meter 17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0104	0.0000	0	4.9435	3.6173	7.6907
Meter 18( $\beta = 0.60$ )	0.0000	0.9657	1.0000	0.5506	0.1897	1.0000	0.8222	6	6.1197	4.5056	9.6816
Meter 19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0318	0	5.3689	3.9308	8.2545
Meter 20	0.0000	0.4310	0.0000	0.0456	0.0242	0.0000	0.0000	1	3.6409	2.6774	5.8041
Meter 21	0.0000	0.0042	0.0254	0.0008	0.0000	0.0000	0.0000	0	3.0224	2.2216	4.7086
Meter 22	0.2699	0.8638	0.0000	0.0000	0.0236	0.0000	0.0077	2	4.5161	3.3245	7.1291
Meter 23	0.0000	0.0000	0.0000	0.0000	0.0291	0.0000	0.0000	0	5.6374	4.1455	8.9461
Meter 24	0.0000	0.0000	0.0000	0.1342	0.0263	0.0000	0.0000	1	3.7305	2.7410	5.8623
Meter 25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	4.9621	3.6453	7.8731
Meter 26	0.0000	0.0000	0.0000	0.1083	0.0000	0.0000	0.0000	1	4.4467	3.2761	6.8539
Meter 27( $\beta = 0.55$ )	0.4373	0.3497	0.6409	0.1935	0.0216	0.0000	0.2179	5	6.4614	4.7501	10.1410
Meter 28	0.0575	0.0616	0.1716	0.0721	0.0905	0.0000	0.0000	2	5.6286	4.1469	8.9779
Meter 29	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	5.5308	4.0580	8.5604
Meter 30	0.0000	0.0261	0.0000	0.0000	0.0000	0.0000	0.0000	0	5.4829	3.9979	8.6413
Meter 31	0.0000	0.0000	0.0000	0.0611	0.0000	0.0000	0.0545	0	4.1730	3.0960	6.4628
Meter 32	0.0000	0.0000	0.1753	0.0000	0.0212	0.0000	0.0000	1	5.3067	3.8824	8.3640
Meter 33	0.1906	0.0241	0.0753	0.0000	0.0000	0.0244	0.1136	2	5.0393	3.6945	7.7912
Meter 34	0.0000	0.0000	0.0000	0.0000	0.0000	0.0139	0.0000	0	5.5947	4.0801	8.5405
Meter 35	0.4215	0.0000	0.1793	0.0000	0.0000	0.0827	0.0228	3	4.4404	3.2445	7.0151
Meter 36( $\beta = 0.50$ )	0.1058	0.0000	0.0000	0.4183	0.1274	0.3024	0.3647	5	3.0685	2.2564	4.8773
Meter 37	0.1709	0.2178	0.0000	0.0000	0.0000	0.0000	0.0000	2	5.3168	3.8993	8.4272
Meter 38	0.0000	0.0283	0.1882	0.0000	0.0000	0.1357	0.0538	2	5.7599	4.2770	8.9824
Meter 39	0.0805	0.0000	0.0000	0.1769	0.0000	0.0000	0.0000	1	3.9088	2.8794	6.2743
Meter 40	0.0232	0.0000	0.0000	0.0000	0.0000	0.0057	0.0847	0	2.7244	2.0116	4.3075
Meter 41	0.0000	0.0182	0.0000	0.0000	0.0000	0.0000	0.0000	0	4.6883	3.4712	7.2936
Meter 42	0.0525	0.0297	0.3863	0.0000	0.1048	0.2218	0.0000	3	3.2739	2.4224	5.1199
Meter 43	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0	5.2811	3.8710	7.8588
Meter 44	0.0000	0.0000	0.1638	0.1894	0.0000	0.0000	0.1924	3	4.5466	3.3239	7.0442
Meter 45( $\beta = 0.45$ )	1.0000	0.5879	0.3028	0.1011	0.1816	0.2126	0.4326	7	6.9150	5.0663	10.8164
Energy Losses (PU)	0.2838	0.2838	0.2838	0.2839	0.2838	0.2837	0.2837				
Avg. Value (threshold)	0.0833	0.1032	0.1035	0.0733	0.0439	0.0650	0.0891	1.37+1.94 (Mean+ $\sigma$ )			



## **CHAPTER 6**

### **6. PROPOSED SMART METER ARCHITECTURE**

#### **6.1 Introduction**

Smart meter components are primarily composed of measuring blocks, computing units and communication means. Due to their complexity each one deserves its own in depth discussion, during this chapter the reader will firstly be introduced to the basic CPU types, board level communications, measurement devices, and finally communication units. At the end of chapter details of the implementation will be given.

#### **6.2 Hardware Selection**

One of the most important aspects of smart meter design is the development of future proof technology, since some electromechanical meters lasted several decades in service, smart meters are also expected to last at least 20 years [124], but information technology is constantly changing, risking deployment obsolescence. It has been recommended that devices should be field upgradeable at the metrology and communication level [124], to support emerging technologies.

According to [41] a “future proof” smart meter should be able to

- Extend functions/services involved in metrology
- Support newer communication standards
- Interoperability between different NAN schemes.

Some characteristics a smart meter design must be capable of handling are enlisted below, according to [41].

- Design must be modular allowing in field retrofitting.
- Reserve 50-80% of computing capacity/flash storage
- Communication protocols should be implemented in software, to allow stack changes
- Use of generic communication chips to enable different channels, modulation schemes, data speeds, or transmission power.

It also has been suggested by [41] that smart meter design should be modular, in Figure 6-1 a concept smart meter architecture is proposed, as it can be seen, a set of dedicated microprocessors are

responsible for controlling certain areas/devices, improving overall system response and allowing upgradeability.

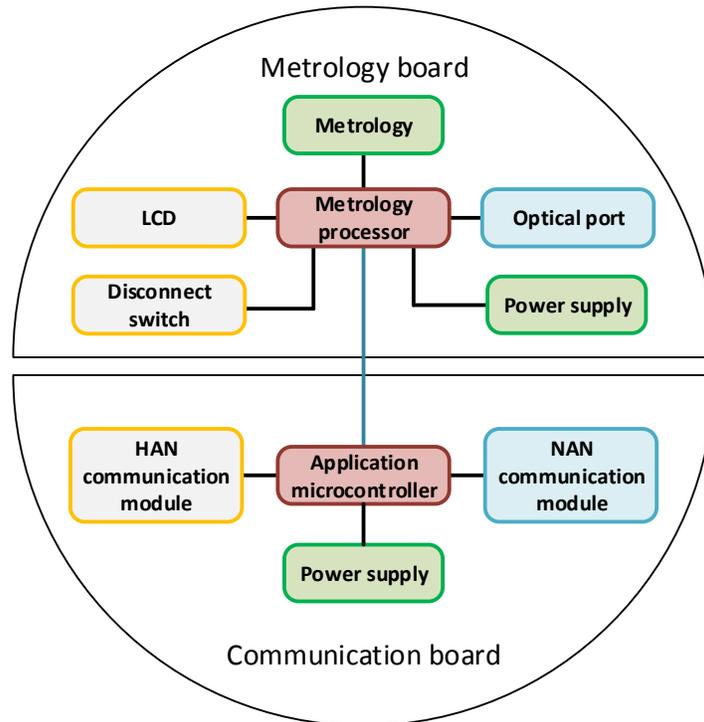


Figure 6-1 Dual microcontroller smart meter architecture, adapted from [41].

Following the before mentioned upgradeable requirements, current designs shall reserve sufficient computing capacity, as well as provide means to operate in a reduced functionality (safe mode) in case of a major failure [124], while providing security and logging capabilities. These requirements require state of the art components for core components such as computing units, communication interfaces and digitalizing circuits.

### 6.2.1 Election of a Computing Unit for a Smart Meter

Computing units can be SoC or Microcontrollers (MCU); some important factors to consider are the available memory, programming space, crypto hardware, operating frequency, power consumption and availability of development tools. Although SoC offer great options for commercial deployment, microcontrollers offer a better starting point for educational purposes due to their flexibility. At the time of writing RISC are commonly used in embedded applications, mostly due to their simplicity and broad availability. Some common available RISC architectures are ARM and MIPS.

### 6.3 MIPS Architecture

The MIPS (Microprocessor without interlocking pipelines) architecture was developed in the 1980's by Professor John Hennessey at Stanford University, it is used as a teaching tool in many universities as an introduction to computer architecture [125]. According to *"The architecture of the MIPS is an ideal example of a simple clean RISC machine which makes it easy to learn and understand"*. MIPS architecture is based on the following principles [126]:

- *All instructions complete in the same time*
- *All instructions operate with the same word size*
- *Simultaneous program data fetch and ram access.*
- *Dependability on the compiler optimizations.*

#### 6.3.1 MIPS instruction set

MIPS is a RISC set computer based on a load/store architecture, fundamental math operations, and program flow instructions can only be done in a limited set of registers (32 in total), with additional instructions to store and load data to this registers from memory addresses. Each instruction takes its arguments, operands and options in a 32 bits field, since its address space it also a 32 bits wide each instruction is fetched in a single clock cycle. Some other implementations enable 16-bit instructions for certain basic operations, improving code density.

A computer Instruction Set Architecture (ISA) describes the basic operations/instructions available in a particular device. These instructions are divided in 3 groups for the MIPS architecture and are summarized on the following paragraphs

**R-type instructions:** These instructions perform operations between two numbers stored in a register ( $S, T$ ); the result is put back into the result register ( $D$ ). The operations are encoded in raw form as illustrated by Table 6.1, where the opcode and function fields denotes the used operation. An example of an R-type instruction is the ADD operation, which can be written in a compiler by:

*add \$rd, \$rs, \$rt; // \$rd = \$rd + \$rt*

Table 6.1. R-Type instruction encoding

B <sub>31-26</sub>	B <sub>25-21</sub>	B <sub>20-16</sub>	B <sub>15-11</sub>	B <sub>10-6</sub>	B <sub>5-0</sub>
Opcode	Register S	Register T	Register D	Shift amount	Function

**I-Type instructions:** These instructions perform operations between an immediate value and an internal S register; the result is put back into the result register (T). The operations are encoded in raw form as illustrated by Table 6.2 where the opcode and function fields denotes the used operation. Some instructions that use this format are the load and store instructions, as well as the arithmetic operations, an example of an I-type instruction is the ADDI operation (add with immediate), which can be written in a compiler by:

*addi \$rt,\$rs,immed; //\$rt = \$rs + immed*

Table 6.2. I-Type instruction encoding

<b>B<sub>31-26</sub></b>	<b>B<sub>25-21</sub></b>	<b>B<sub>20-16</sub></b>	<b>B<sub>15-0</sub></b>
Opcode	Register S	Register T	Immediate value ( $-2^{15} \rightarrow 2^{15}$ )

**J type Instructions:** These instructions perform jumps that cause the program flow to be altered. The operations are encoded in raw form as illustrated by Table 6.3 where the opcode and function fields denote the used operation. An example of a J-type instruction is the Jump operation, which can be written in a compiler by:

*j \$immed; //Jump to address immed*

Table 6.3. J-Type instruction encoding

<b>B<sub>31-26</sub></b>	<b>B<sub>25-0</sub></b>
Opcode	Target address

### 6.3.2 *Pipelining*

Pipelining allows maximizing microprocessor use by overlapping instruction steps. This technique can be explained by the laundry example [127], suppose an individual has to wash three loads of clothes (a), put them in the dryer (b), fold them (c) and finally put them in the cabinet (d). If each activity takes an hour to complete, he could perform the activities in series, that is to say “wash load 1, dry, fold, and store; wash load 2, dry, fold, and store; wash load 3 dry, fold, and store” he would finish in 12 hours (See Figure 6-2). If instead he performs the same activities in parallel by using the available resources (it also can be seen as the assembly line of vehicles), he can finish in a lesser time, as illustrated in Figure 6-2.

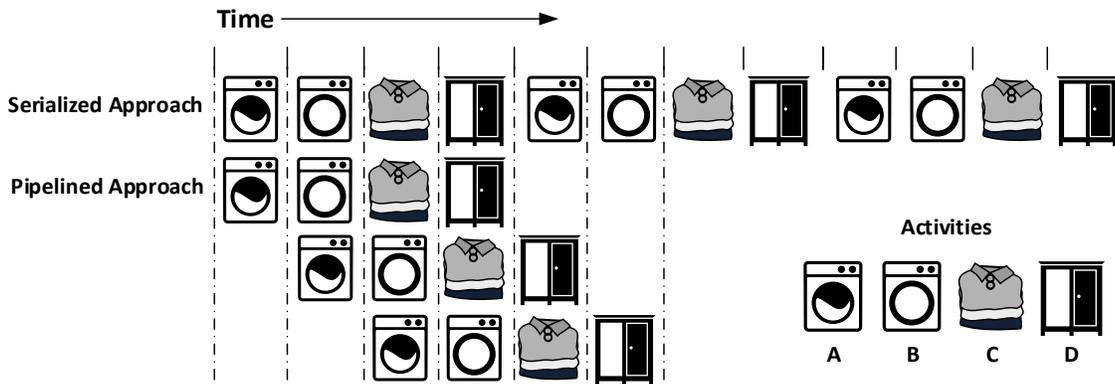


Figure 6-2 Pipeline approach as a parallel execution mode.

This “parallel” technique allows using stand by units while the process finishes. Although the concept looks easy, it introduces certain restrictions. Going back to the laundry example, each activity must be performed sequentially (i.e. first wash then dry), leaving some steps without duty until the process reaches that stage, it also requires that each activity takes the same time to keep the process in sync, and it requires someone to “feed” activities to each step (dispatcher). Coming back to the electronic world, the laundry duties are instructions that must be executed in a repeatable manner yet causing unpredictable results to the dispatcher. The MIPS microcontroller archives this behavior by using an internal 5-stage pipeline; these steps are described in section 6.4.1

### 6.3.2.1 Pipeline Problems

Pipelining ideally completes one instruction per cycle once the first passes through the pipeline stages, but certain problems can cause disruptions in the pipeline process. One common example is data dependencies. Suppose a simplified three-stage pipeline exists (see Figure 6-3), where the result is available at the completion of the third step. If the first instruction computes  $a = 1 + 2$ , and a second instruction computes  $b = a + 1$ , the pipeline will stall due to a data dependency as shown in Figure 6-3. Data dependencies occur when an instruction operand depends on previous results (which are most sequential algorithms) [127]. Pipeline stalls stop the whole execution pipeline, causing overall lesser than 1 instruction per cycle throughput, these problems worsen with higher pipeline levels (nowadays some microprocessors have up to 14 stages [128]), since the result takes longer to complete. There are several programming techniques designed to avoid pipeline stalls, as

well as hardware workarounds, but a simple approach is to transform parts of code into parallel multistep computation, alternating between parts of operations to prevent pipeline stalls.

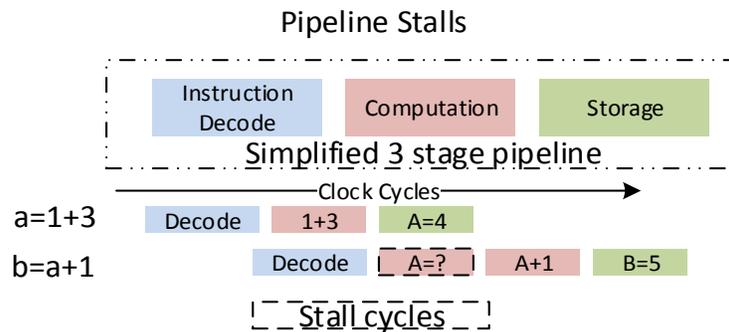


Figure 6-3 Simplified pipeline architecture showing pipeline stalls

Hardware solutions to prevent hardware stalls are available for some MIPS architectures. Special bypasses allow to omit register storing and to redirect the result to the requesting pipeline instruction. This mostly works for register-to-register operations and simple arithmetic operations, but it is important to always check the type of stalls that can't be prevented in hardware [129].

Another pipeline problem occurs during branch instructions. Since the pipeline must be always be full, the microprocessor has to assume the comparison will return false or truth to load the appropriate instructions. If the assumption is correct the pipeline could run at maximum efficiency, but if it was wrong it must discard the pipeline to begin a new execution. These kind of problems have been solved in the past by using probability analysis and heuristic code analysis, that tend to reduce pipeline flushes, yet they occur, for the MIPS R2 architecture, the branch instructions complete early on the pipeline process, allowing flushing and loading the pipeline without interruption. To archive this the instruction following the branch will always be executed; this instruction can be filled with useful data or a "No Operation" (NOP) instruction depending on the programmer/compiler ability.

Other issues that must be considered with pipelined implementations are interrupts; interrupts enable CPU units to perform a certain task under demand, commonly by means of a trigger signal. Since the instructions on a pipeline are already queue, a pipeline flushing and context-restore (saving the current program state) mechanism must be implemented, for RISC processors the queue flushing is often done in hardware, while the context handling mechanism is often done in code.

### 6.3.3 Memory architecture

MIPS is a load/store architecture, which means it fetches data from an external memory unit (although physically located inside the microcontroller). This external architecture causes speed limitations on the data access speed. To overcome this speed limitation different types of memory are available on the MIPS architecture. The MIPS core uses a traditional Non Uniform Memory Access (NUMA) architecture, which enables low access times for commonly used data though the use of a cache unit and a separate code and data bus. NUMA architectures enables CPUs to minimize waiting times by hierarchizing memory types, as shown in Figure 6-4, the memory access time depends on the relative distance from the CPU core, and as a general rule, memory size increases as speed decreases due to manufacturing costs.

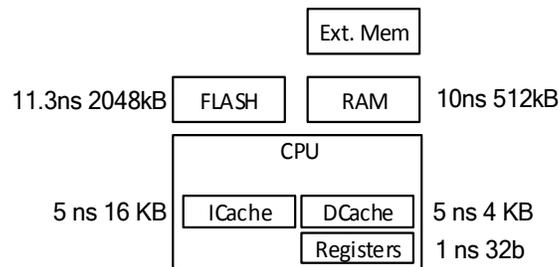


Figure 6-4 NUMA Memory hierarchy available on PIC32MZ a type of MIPS microcontroller, showing different memory capabilities.

#### 6.3.3.1 Cache memory

Cache memory functions as a small buffer that transparently enables the CPU to use larger and slower memory types [127]. Cached or buffered data is allocated inside a very fast but limited size memory unit, these units often store additional data that enables it to index data. When the CPU requests a particular address, the cache unit determines if it has a local copy before accessing the RAM unit; if it is inside the cache, the unit works with this copy. If it is not cached it fetches it from the RAM and makes a local copy for future use. Since the cache is limited in size, several algorithms have been devised to determine which data should be replaced or added to cache.

Since cache units are designed to improve execution speed by minimizing access time, programs can be optimized to use the cache adequately, improving speed and reducing power consumption in power aware applications [130]. These program improvements can be summarized in the following statements:

- Prefer divide and conquer algorithms vs sequential ones.

- Break up routines into segments that utilize variables that fit into the data cache.
- Perform loop-unrolling techniques that fit into the instruction cache.

Although cached architectures enable faster overall execution speed, they introduce problems known as cache incoherencies. This problem arises when two or more devices access the same data container (e.g. RAM memory), for example in multicore microprocessors, and in general where multiple devices can alter the data contents. There are additional problems introduced in the world of crypto security when cache is enabled. Side channel attacks can be performed by analyzing small timing differences during crypto execution, as well as power consumption differences. These side channel attacks can be further exploited in multitasking or non-exclusive operating systems, by reading the cache after crypto is executed. These side channel attacks must be addressed according to the crypto primitives used and characteristics of the system.

#### 6.4 The PIC32MZ Architecture

The PIC32MZ microcontroller is based on the MIPS microAptiv™ core that contains all of the MIPS release 2 characteristics, plus a dedicated DSP core. It provides two cache units designed to reduce processor-stalling states. The core in this chip is rated to run up to 200 MHz. The core components of this architecture are illustrated by Figure 6-5, while a detailed view is available it Figure 6-6.

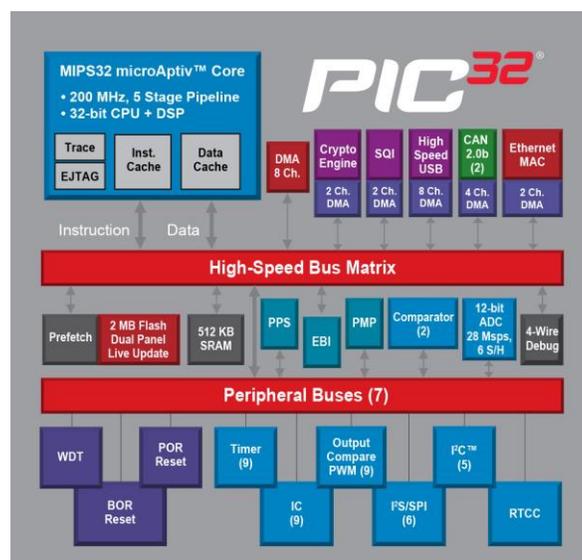


Figure 6-5 Core elements of a PIC32MZ microcontroller

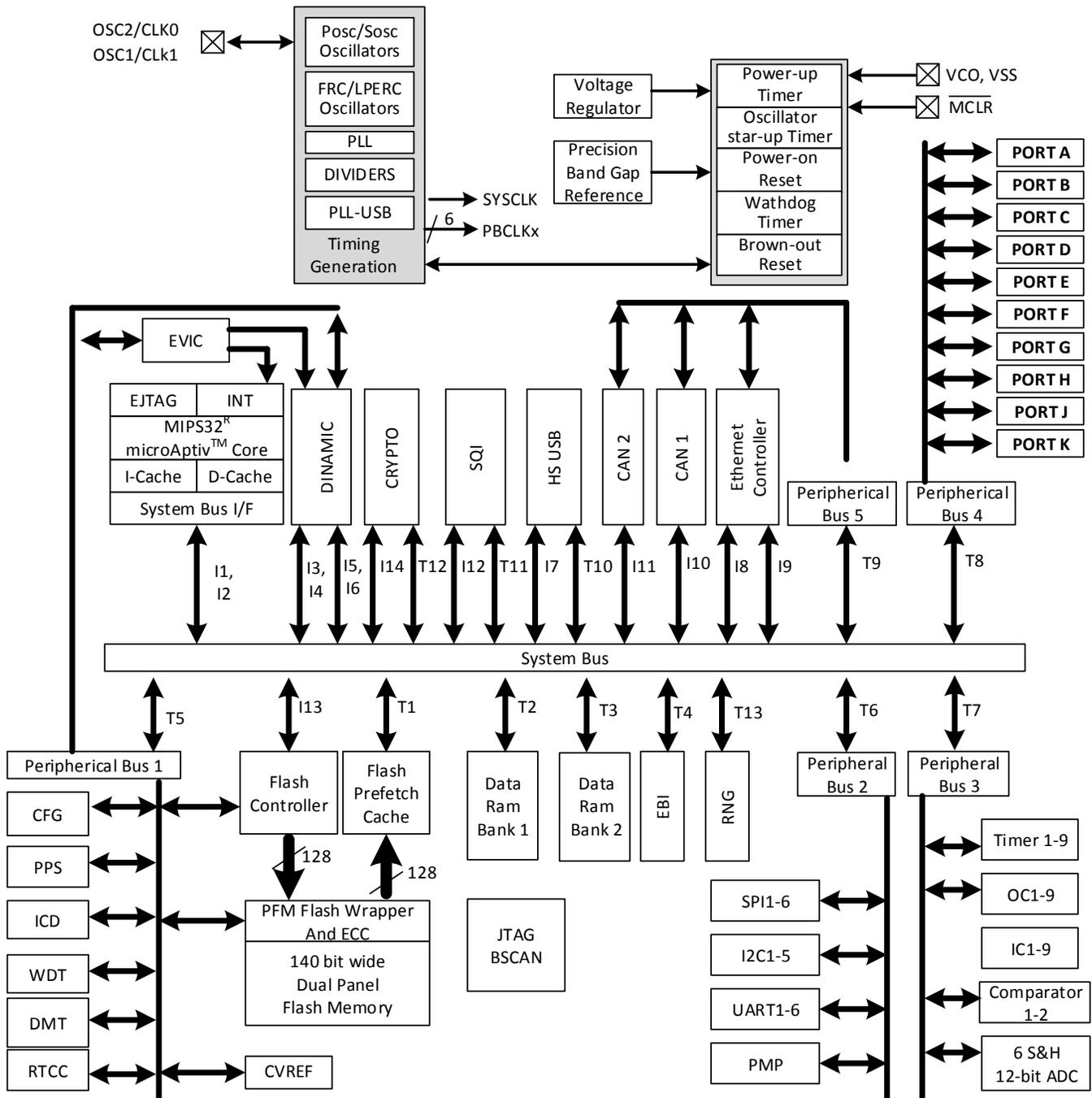


Figure 6-6 Detailed hardware architecture of the PIC32MZ unit.

#### 6.4.1 The PIC32MZ pipeline

The PIC32MZ series uses a MIPS R2 MicroAptiv core, which uses a 5-stage pipeline with hardware improvements to prevent stalls during most operations [129]. The MicroAptiv core allows simultaneous 16 bit and 32-bit instruction size execution, useful for improving code density, as well

as parallel single cycle 32 x 32 multiply and Accumulate (MAC) unit. The pipeline stages are further discussed in the next sections.

### **I Stage**

During **Instruction** fetch data is retrieved from the SRAM unit. In case of 16-bit instructions, these are converted to 32-bit instructions.

### **E Stage**

During **Execution** operands are fetched from the register file (Internal CPU registers). Arithmetic Logic Unit (ALU) operations begin certain operations such as address computation, and branch decisions are performed.

### **M Stage**

During the **Memory** fetch step, memory addresses are loaded according to the registers. Most ALU computations are completed in this step and are available for bypassing (see section 6.3.2.1). During operations involving multiplication a parallel hardware is calculating the result.

### **A Stage**

During the **Align** stage, boundary conditions are performed, in the case of 16x16 bit multiplications, result is available for register bypassing (see section 6.3.2.1); otherwise, load instructions are completed.

### **W Stage**

During the **Write** stage, actual results are stored into the registers or to the RAM memory. During this phase 32x32 bit operations complete, allowing just in time register saving.

The simplified MicroAptiv core (MIPS R2) pipeline architecture is illustrated in Figure 6-7. As it can be seen numerous bypassing and multiplexing techniques exist, allowing high instruction throughput rate. Nevertheless, certain instruction slots (nondependent instructions) must be introduced to prevent stalling. These cases are resumed in Table 6.4. One thing to notice is that multiply instructions depend on an external unit that requires an additional cycle for 32x32 bit operations.

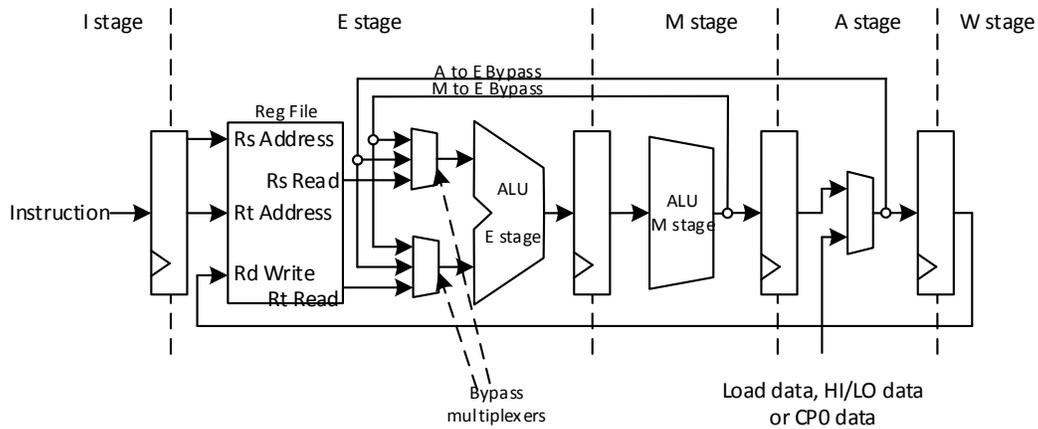


Figure 6-7 Simplified MIPS pipeline embedded in the PIC32MZ, adapted from [129].

Table 6.4 Wait slots required before issuing data dependent instructions, adapted from [129]

Instruction type [PC+0]	Instruction type [PC+1]	Required slots
MUL (16x16) <i>rs, ..., ...</i>	Instruction <i>..., rs, ...</i>	1
MUL (32x32) <i>rs, ..., ...</i>	Instruction <i>..., rs, ...</i>	2
MUL (32x32) <i>rs, ..., ...</i>	MUL (32x32) <i>rt, ..., ...</i>	2
[Lw, Lh, ls] <i>rs, ...</i>	Instruction <i>..., rs, ...</i>	2
MFC0 <i>rs</i>	Instruction <i>..., rs, ...</i>	2
[MFHI, MFLO] <i>rs</i>	Instruction <i>..., rs, ...</i>	2
[MFHI, MFLO] <i>rs</i>	MUL <i>rt, ..., ...</i>	1
MADD	every one, except MADD	0
MADD (16x16)	MADD (16x16)	0
MADD (32x32)	MADD (32x32)	1
Jumps, Branches	Any will be executed	0
Cache Flushes	-----	3

### 6.4.2 PIC32MZ interrupts

The PIC32MZ unit implements a series of interlocks that enables interrupts to occur faster than typical MIPS implementations; this is done thru a hardware-based context-restore mechanism that transparently switches the working registers. In total a set of 7 shadow registers (each containing a unique set of 32 registers) are available for fast context switching. The PIC32MZ also features a table driven interrupt vector that enables to execute a particular interrupt without requiring multiple jumps.

### 6.4.3 *PIC32MZ Assembly language*

High-level languages, such as C are preferred due to their portability and easy to understand syntax, but they might introduce an unnecessary overhead that can affect performance, particularly in CPU intensive tasks. Assembler Language or Machine Language lets programmers optimize parts of their code at the cost of portability and readability, Assembly language is a CPU dependent code (ISA), meaning only a limited set of architectures can be compatible with the same code. The PIC32MZ microcontroller uses the MIPS R2

### 6.4.4 *Cache memory in the PIC32MZ*

As shown in Figure 6-4, the PIC32MZ series uses a dual cache architecture, allowing fetching instructions and data at the same time. It resembles a modified Harvard architecture, where additionally peripherals are mapped to the memory space but are internally addressed by several buses. This allows different write speeds according to the application requirements; the I-cache (instruction cache) and D-cache (data cache) allow up to 200 MIPS, provided program flow resides in the cache. When execution resides outside the cache, the maximum speed is limited to 88 MIPS at the best [131]. Furthermore, the cache units can be dynamically enabled or disabled to cope with the designer needs. As an additional benefit, proper cache use allows to lower power consumption by putting memory in a standby mode.

### 6.4.5 *Direct Memory Access (DMA)*

Traditional computer architecture required CPU intervention to transfer data between different devices and memory units. While this technique could be fine for small chunks of data, this technique became a bottleneck when target devices operating speed were slower than the CPU or large amounts of data were needed to be transferred. In certain cases, polling mechanisms or interrupt driven transfers introduced large computational burdens, thus wasting valuable computing power.

Due to the limited CPU-RAM bandwidth, computer architects devised a method for transferring data among I/O units by using a parallel interface called the Direct Memory Access (DMA) unit. A DMA unit allows transferring a certain amount of data from A to B without involving the processor,

handling all required transfer operations within itself (wait states, interrupts, and permissions). DMA units are usually configured by the CPU to transfer data with certain parameters and are commonly set to fire an interrupt when done, thus reducing the amount of work done by the CPU.

A DMA can be thought as a limited scope CPU. In certain architectures, such as the PIC32MZ, the DMA unit can auto-clear system interrupts via internal hardware mechanisms and perform logical operations during transfers. Although DMA can be seen as efficiency module, it can introduce problems in cache-enabled architectures. Since cache is managed inside the CPU, the DMA cannot invalidate cache entries resulting in cache coherency problems. This problem is illustrated in Figure 6-8. DMA enabled architectures require special handling mechanisms via software or hardware. In particular, on the PIC32MZ, variables can be addressed in the virtual space, allowing DMA affected memories to always be read from the RAM, and bypassing the cache.

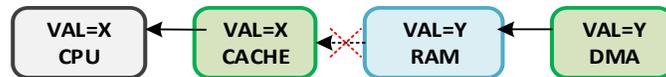


Figure 6-8 Cache coherency problems due to the DMA unit.

#### 6.4.6 *Bus access groups*

PIC32MZ series microcontrollers offer different busses to access peripheral devices, allowing simultaneous access to bus initializers (master units, such as the CPU and DMA units) to different targets (Peripherals, Ports, RAM, and FLASH). These multiple access buses enable zero latency access times if target access are scheduled or organized before application development.

To improve overall speed RAM is divided in two banks, allowing simultaneous use of CPU processing and DMA transfers to/from peripherals. Furthermore, bus arbitration priorities can be dynamically assigned to the DMA and CPU units to improve latency. In Figure 6-9, the available access groups are shown. The dual bank architecture allows the DMA to operate on a semi-transparent mode, this is, it does not halt the CPU access to RAM memory during transfers. This type of memory partitioning can be further configured by permission groups that can be configured to meet the security requirements.

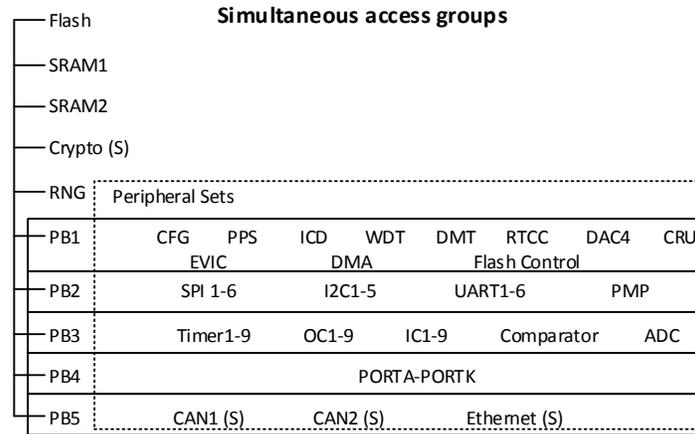


Figure 6-9 Simultaneous access group, i.e. initiators do not require bus arbitration schemes.

#### 6.4.7 Description of commonly hardware modules

Microcontroller units, often have several peripheral sets to accommodate wide designer demands. Although some of them are useless for specific purposes, some core modules can be used by a wide array of applications; these units are further discussed below.

#### Oscillators

The PIC32MZ is capable of executing instructions with various oscillator sources, such as crystals, RC circuits, or via external and internal clock sources. It features a backup clock that can be used as an emergency clock source to report a major fault. The external clock can be multiplied and/or divided to accommodate CPU speed requirements, although limits should be observed. The PIC32MZ allows to dynamically changing the clock sources in code, or by adjusting the internal bus clocks. These facilities enable to change power consumption patterns according to the computing demand.

#### Peripheral Pin Select (PPS)

Until recent years, designers needed to plan PCB layout according to the fixed pin mapping of the devices, restricting component layout. In recent years, reconfigurable hardware has been migrated from the FPGA to the microcontroller world. By using high-speed switches, it is possible to configure peripheral functions to groups of available physical pins, thus enabling to group together used peripherals. This also enables to support new hardware via software updates by redirecting physical pins to virtual mappings, which in practice can make designs future proof [41]. This virtual pin mapping also allows using modules in parallel, such as external interrupts and input capture modules.

## **PORTx**

Traditional I/O output ports are available on the PIC32MZ. These ports can be used to support protocols via software (bit banging) or as general-purpose pins. Pin states are mapped to memory space, allowing transparent load/store operations to improve responsiveness. Each port has additional registers called PORTxCLR to clear specific bits in an atomic manner, as well as PORTxSET to set bits. For toggling operations, an additional PORTxINV is available. I/O port operating speed can be configured through the PB4DIV setting.

## **SPI**

Up to eight SPI communication channels are supported, each having a 128 byte internal buffer that can be configured according to the word size (8, 16, and 32 bits). Each channel can be configured to interrupt according to various conditions and supports auto framing protocols. SPI speed is limited to 25 MHz and can be configured with SPIxBAUD and PB2DIV register settings

## **Input Capture Channels (ICx)**

Up to nine input capture channels are available for assigning via PPS. These channels enable to register the exact time on which the signal was received, expressed in terms of the internal counters values. These counters can be synched to external clock sources or to system frequency components. Each of the channels has a hardware 4-level FIFO register that enables to record fast changing signals.

### **6.4.8 PIC32MZ limitations.**

Although the PIC32MZ unit offers execution speeds of up to 200 MHz, this parameter could be misleading, since in real life this limit is only achievable if the data is executed from the cache units and no pipeline stalls occur. A more realistic value was given in section 6.4.4, that limits the CPU operation to 88MHz if the cache unit remains disabled. However, it can be hard to find clock configurations that archive that speed, and thus a more realistic 80 MHz speed limit is proposed. This speed configuration enables the CPU to use all the peripherals without any waiting period, simplifying the development of pipelined optimizations by simplifying the hardware access times.

## **6.5 Proposed Smart Meter Architecture**

Following the principals laid in section 6.1 a dual controller setup was adopted; this enables the design to have a dedicated microcontroller for network communications and a dedicated microcontroller for measuring signals. This allows setting two simultaneous process to a high

priority rating where the network controller is only dedicated to handle the data being transferred/requested, and the measuring unit is only dedicated to provide high quality measurements.

### 6.5.1 Dual microcontroller setup

The author proposes the architecture shown in Figure 6-10, which is based on the requirements of a smart meter capable of measuring energy with time-stamped characteristics that is intended to satisfy the requirements imposed by the algorithm developed on Chapter 5 of this thesis. From the smart meter components view, a dual metering and communications microcontroller architecture is proposed (MCU units are highlighted on yellow), with its associated hardware (signal acquisition modules, radio interfaces, and power supplies). On top of this basic design (mostly based on the ones given in Figure 1-5 and Figure 6-1) the PMU components were incorporated (GPS receiver unit, and frequency measurement hardware). Finally some voltage references and accurate crystals were added to the design in order to comply with a 0.5% class precision (according to the C12.10 standard).

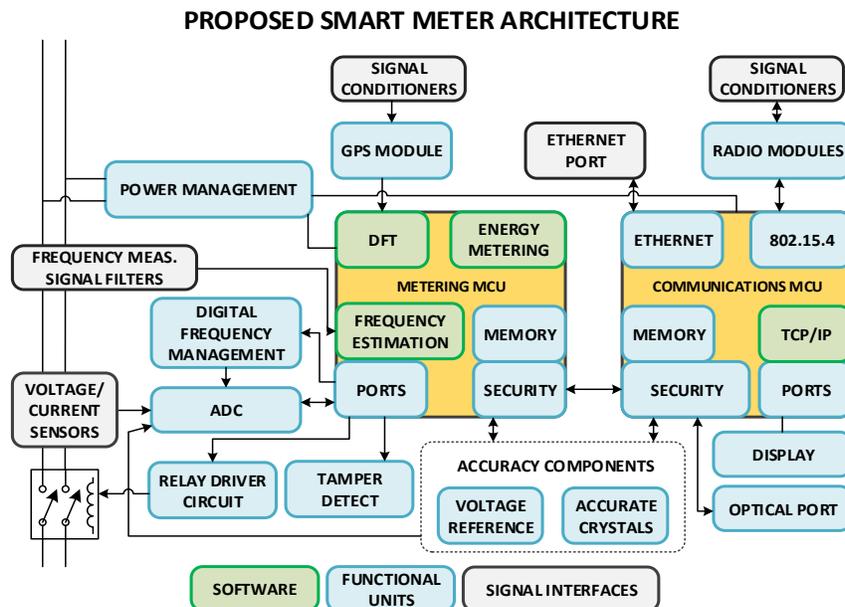


Figure 6-10 Proposed smart meter architecture, featuring a dual microcontroller setup.

In the next sections an in-depth description of the composing hardware is given, inserting circuit descriptions or software flowcharts where applicable.

### 6.5.2 *Analog to Digital conversion hardware implementation.*

An essential component of energy metering is the ADC module. As mentioned in Chapter 2, an assortment of ADC architectures are on the market, each having certain benefits and limitations, and thus the designer should select the device according to the application requirements and possible advantages or drawbacks, including the economical factor. From the technical viewpoint energy metering requires a high precision and a relative low conversion speed (up to the intended harmonic components) [2]. This often leads designers to choose successive approximation ADCs. Although SAR based ADCs do a good job, they require careful planning to prevent aliasing effects, and are susceptible to noise [132]. On the other hand, Sigma-Delta converters offer almost no aliasing effects due to their oversampling technique, and tend to lower noise levels due to their internal signal averaging (decimator) filter, but suffer from speed limitations.

Although most microcontrollers offer internal ADCs, an off-the-chip ADC offers greater application flexibility and improved dedicated hardware performance, even though cost rises. On the other hand, SoC solutions offer reduced PCB size/complexity and a lower Bill of Materials (BOM), providing at the same time increased security by reducing possible attack points [133]. Security in smart meters is of particular interest due to potential attempts to lower registered consumption.

The PIC32MZ series microcontroller offers a high-speed 10-bit SAR based ADC module, which due to its limited resolution was excluded. The selected ADC module was the Texas Instruments™ ADS131E06 unit, which employs a delta-sigma architecture. Its communication is SPI based and it offers 6 simultaneous differential sampling channels with programmable gain amplifier (PGA). The ADC sampling rate can be driven externally to synchronize its sampling rate with the electrical system frequency. This ADC can be configured to oversample up to 256-samples reducing the overall system noise, a detailed explanation of this chip can be found at [134].

The ADC module archives different ENOB levels that depend on the sampling speed, the sampling rate is chosen according to the intended harmonic, in this case the 51<sup>st</sup> harmonic is required by IEEE C37.118, thus a minimal of 102 samples per cycle are required to meet the Nyquist requirement. Since digital systems work in powers of 2, a total of 128 samples per cycle window were selected, given a 60 Hz fundamental frequency a total of 7.680 *kSPS* are required according to Eq. 6.1.

$$kSPS = W_{size} * F = 128 * 60Hz \quad \text{Eq. 6.1}$$

where

$$kSPS = \text{kilo Samples per second}$$

$$W_{size} = \text{window size (samples per cycle)}$$

Table 6.5 enlists the ENOB levels according to the sample rate and PGA gain. From this table it can be determined that an 18.0-bit resolution is expected from the ADC unit.

Table 6.5 Internal Register Configuration of the ADS131E08 for various ENOB, taken from [134]

DR BITS (CONFIG1 Register)	OUTPUT DATA RATE (kSPS)	-3 dB BANDWIDT H (Hz)	PGA GAIN									
			x1		x2		x4		x8		x12	
			DYNAMIC RANGE (dB)	ENOB								
000	64	16788	74.1	12.31	74.1	12.30	74.0	12.29	74.0	12.29	73.9	12.27
001	32	8384	89.6	14.89	89.6	14.88	89.4	14.85	88.6	14.71	87.6	14.55
010	16	4192	102.8	17.07	102.8	16.99	100.6	16.72	97.1	16.12	94.2	15.65
011	8	2096	108.2	18.0	107.4	17.9	105.2	17.5	101.6	16.9	98.9	16.5
100	4	1048	111.4	18.6	109.4	18.4	107.4	18.1	103.5	17.4	100.5	17.0
101	2	524	114.6	19.1	113.7	19.0	111.4	18.6	107.7	18.0	104.9	17.5
110	1	262	117.7	19.6	116.8	19.5	114.5	19.1	110.7	18.5	108.0	18.0

Since the chosen ADC unit allows using a dynamic frequency clock in between a 1.7 and 2.25 MHz to adjust its sampling frequency [134] the Eq. 6.2 is presented. This equation allows obtaining the frequency ranges that can be sampled by the chip, by solving this equation the effective measuring range was determined to be valid from 51.87 Hz to 68.66 Hz

$$F = \frac{Master_{clock}/256}{W_{size}} \quad \text{Eq. 6.2}$$

In order to use this module other auxiliary circuits are needed, and are discussed on the following sections

### 6.5.2.1 Accurate voltage reference

Most ADC units require a  $+V_{CC}$  reference voltage to establish the maximum reference value ( $V_{REF+}$ ) and a  $VSS$  or a  $-V_{CC}$  reference to serve as the low reference level ( $V_{REF-}$ ), making the  $[V_{REF+} - V_{REF-}]$  difference the voltage range of the ADC unit.

A voltage reference provides ADC units with the voltage thresholds to adjust the upper and lower sampling bounds, for the ADS131E06 these can be in the form of a dual voltage supply (+V and -V), this enables the unit to measure signals that alternate around a 0 volts level (like electrical AC signals).

Voltage references can be created by using resistor ladder networks or by using dedicated circuits that offer a fixed output independent of the input voltage or current demand, these dedicated voltage references primary exist in two forms, shunt and series. Shunt voltage references are known for their voltage stability independent of the input voltage variations, shunt voltage references operate under the principals of a variable resistor ladder network, but instead use the silicon properties to vary the breakdown voltage [135]. The typical shunt voltage reference is shown in Figure 6-11, in this case the LM4030 was chosen according to the ADS131E06 voltage reference characteristics.

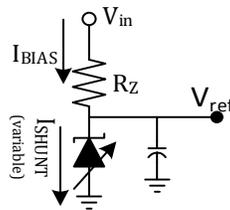


Figure 6-11 Typical shunt voltage reference architecture, adapted from [135]

The LM4030 unit offers a fixed 2.5 reference voltage that can be adjusted to generate positive and negative values, it is designed to achieve a 0.05% accuracy under a variety of current demands and input voltage variations [136]. It requires an external resistor to improve its hysteresis characteristics, which value depends on the current demand at the *output* point (see Eq. 6.3).

$$R_Z = (V_{IN} - V_{REF}) / (I_{MIN\_OPERATING} + I_{LOAD\_MAX}) \quad \text{Eq. 6.3}$$

By considering the current requirements of the ADS131E06 unit, the value of a 68-ohm resistor was obtained by evaluating Eq. 6.3. The final circuit that feeds the ADC module is shown in Figure 6-12.

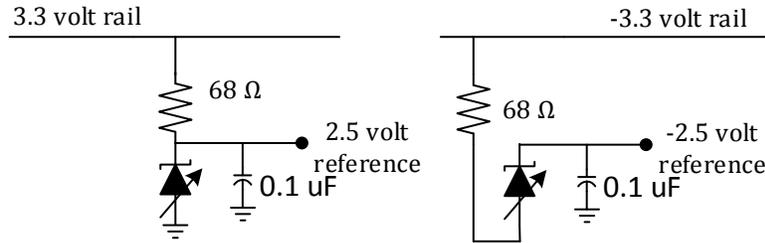


Figure 6-12 Dual supply voltage references employed by the ADC circuit

### 6.5.2.2 Differential voltage signal acquisition

Traditional ADCs are often restricted to operate with positive signals, and thus operate by using single ended measurements (by setting  $V_{REF-}$  to ground). Those ADCs require a DC offsetting mechanism to enable AC signal measurements, these DC offsets can be implemented by using operation amplifiers (OpAmps) or resistor networks [31].

The ADS131E06 unit is designed to operate with negative and positive signals, and is thus able to employ advanced measuring techniques. This unit can be operated on a differential mode that uses two linked channels to perform differential signal measuring, differential measurement techniques simplify signal acquisition circuits and improve noise characteristics by doubling the ADC range and removing DC components (see Figure 6-13).

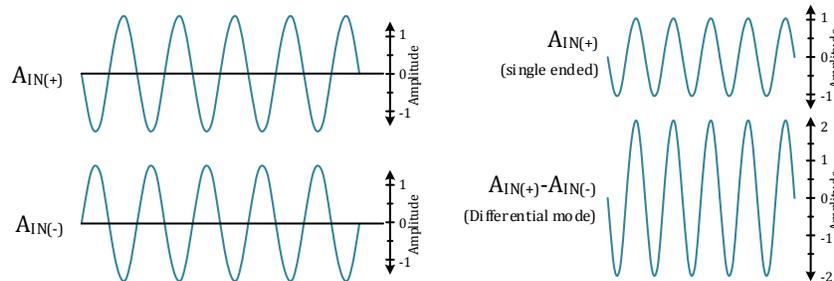


Figure 6-13 ADC ranges under different operational modes

For the designed meter, the voltage signals are acquired by means of a resistor ladder that outputs a differential signal, the diagram of this circuit is shown in Figure 6-14, where  $R_1$  and  $R_2$  are used as the voltage divisors, and  $R_3$  and  $C$  are used as anti-aliasing filters.

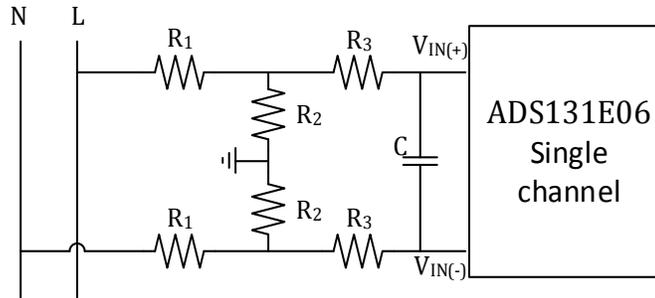


Figure 6-14 Ladder circuit used for differential voltage acquisition

In order to design an adequate voltage divisor, several aspects must be considered to achieve the intended accuracy levels, and prevent damages to the unit in case of voltage variations, some of these aspects are listed in the following sections.

- The input resistance value for voltage probes should be ideally infinity
- The maximum voltage received at the low voltage side must be below the ADC limits (for the ADS131E066 this limit is 2.4v), but the usable range should be maximized at all times
- The voltage across the resistor terminals should not allow arc gaps to form.
- The resistors used should have low thermal variability coefficients, as well as low tolerances to archive the intended accuracy.

Based on the before mentioned requisites a 230 L-N (peak) voltage (equivalent to a 30% voltage swell in a 127  $V_{RMS}$  signal) was established as the maximum allowable value,  $R_1$  was selected to be 499K $\Omega$ , while  $R_2$  was selected as 5.2K

By employing the voltage divisor rule (see Eq. 6.4) it can be proof that the maximum voltage is below 2.4 volts under the previously discussed parameters.

$$V_{output} = V_{input} * \frac{R_2}{R_1 + R_2} \quad \text{Eq. 6.4}$$

### 6.5.2.3 Differential current signal acquisition

Current signal acquisition is often done thru a current transformer that lowers the primary current into a lower secondary current, since ADCs do not measure current a burden resistor is often added to read the voltage drop across the burden resistor terminals.

The selected CT is an ACT-0750-020 unit manufactured by *continental control systems* that according to the manufacturer offers the following characteristics.

- Accuracy:  $\pm 0.50\%$  from 1% to 120% of rated primary current
- Phase angle:  $\pm 0.25$  degrees (15 minutes) from 1% to 120% of rated current
- Primary rating: 20 Amperes
- Internal burden resistance  $26.4\Omega$  \*at 0.333 mv output.

With the before mentioned characteristics a resistor based differential circuit was proposed, in this case the burden resistance is maintained to prevent core saturation under high currents, the proposed circuit is shown in Figure 6-15, where  $R_1 + R_2 = 26.4\Omega$  and  $R_1 = R_2$ , the  $R_3$  resistor is set in accordance with the  $C$  value in order to generate a low pass filter near the 51<sup>st</sup> harmonic ( $R_3 = 422\Omega$  and  $C = 0.1\mu F$ )

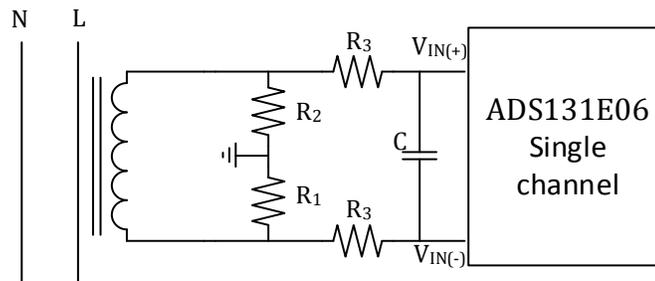
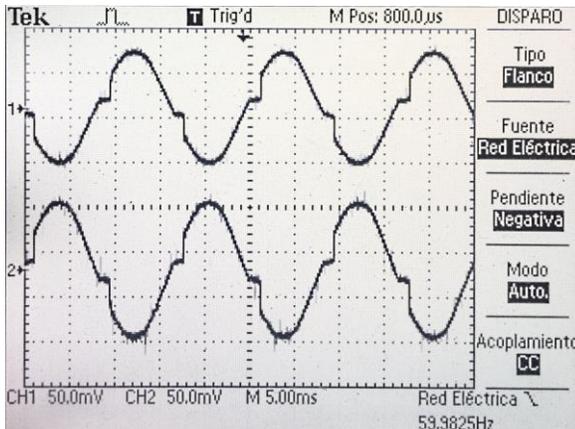
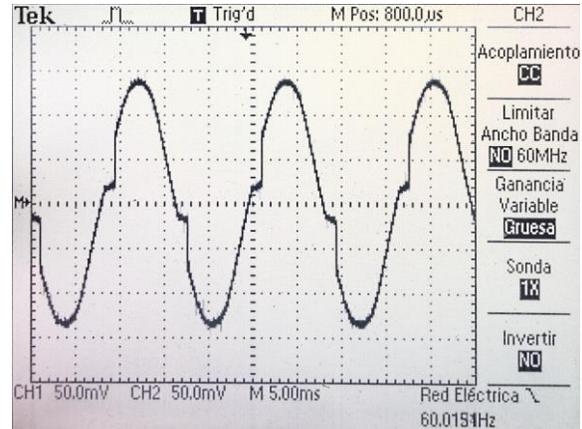


Figure 6-15 Differential measurement circuit used for TC current acquisition via a burden resistor.

In Figure 6-16a a sample current signal as viewed from the  $V_{IN(+)}$  and  $V_{IN(-)}$  terminals can be observed. This particular signal exhibits high noise characteristics due to the non-linear nature of the load, but its effects are minimized by filtering high frequency noise and employing  $V_{IN(+)} - V_{IN(-)}$  (Figure 6-16b).



a) Current seen by each terminal



b) current seen by the ADC unit

Figure 6-16 Actual signals (from a non-linear load) measured by the differential current channel

### 6.5.3 Frequency measurement-Hardware Implementation

The ADC unit offers the possibility of dynamically changing the sampling frequency; this can be employed to comply with certain measurement standards, like the IEEE C37.118. In order to use the feature a frequency measurement mechanism must be first developed, this can be implemented in software or hardware, software solutions can be based on the zero cross detection with filtering mechanisms to improve the response when harmonics are present [31]. Hardware solutions can be based on filtering techniques that transform the sinusoidal waveform into square waveforms that can be directly used by other hardware units, each of this solutions offer advantages and drawbacks that must be analyzed by the designer.

For the proposed meter a hardware solution was selected, the solution is based on a low pass Bessel filter that exhibits a linear phase response; this enables the filter to operate under fast frequency sweeps. These fast frequency sweeps are often required for standard compliance, like the IEEE C37.118 dynamic test.

Although the proposed solution relies on the use of a Bessel filter additional steps are required to transform an AC signal into a square wave that can be considered digital (see Figure 6-17), this processes are described in the next sections.

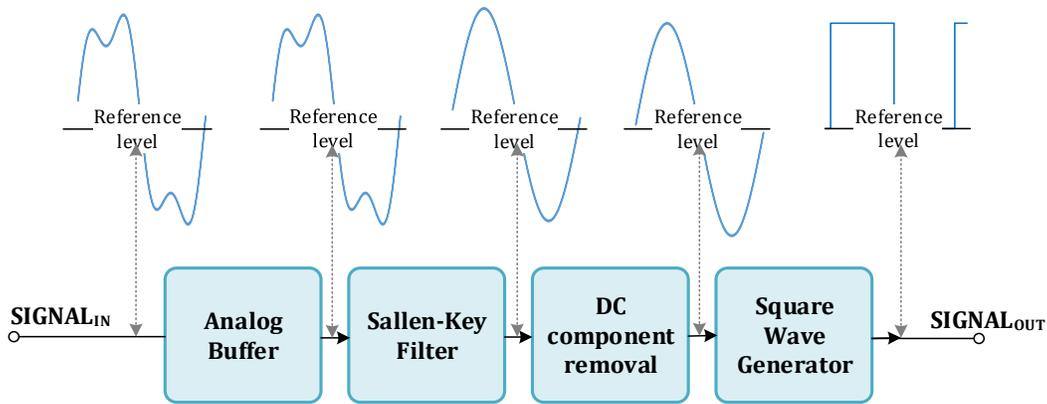


Figure 6-17 Frequency measurement device components.

### 6.5.3.1 Analog buffer

The chosen frequency source was the voltage signal, due to its constant nature (it does not vary with the load characteristics), the voltage signal is extracted from the voltage divisor circuit described in section 6.5.2.2, but since the use of filters can cause the original signal to degrade, a buffering mechanism is employed.

This buffering mechanism is based on the voltage follower circuit of an OpAmp device; it uses a unity gain factor and an inverting function that does not alter the measured frequency. The used OpAmp is the TL974 from Texas instruments, that fully satisfies the design requirements (speed and voltage), once the signal is buffered it is fed into the next stages of the filtering mechanism, the block diagram for this circuit can be observed at Figure 6-18

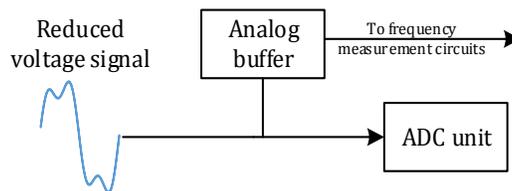


Figure 6-18 Designed buffered architecture.

### 6.5.3.2 Sallen Key Filter

As mentioned by the introduction a Bessel filter was used to remove higher harmonic components, that could cause erroneous frequency readings, the employed Bessel filter was constructed from the Sallen-Key filter topology described in section 2.3.5. The designed filter is a third order filter

that ensures fast harmonics attenuation while preserving a constant delay group near the 60 Hz band, the filter architecture is described by Figure 6-19.

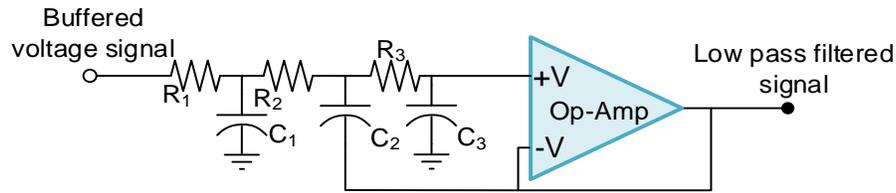


Figure 6-19 Architecture of the third order Sallen-Key filter topology used to filter out high frequency components.

The filter was designed according to commercial components values, with a best-fit solution designed to obtain a constant phase delay near the 60 Hz band, the program was created in Matlab™ employing the equations provided by [52]. The chosen filter uses the values described by Table 6.6, in

Table 6.7 the designed filter characteristics are given.

Table 6.6 Sallen-Key filter components

Component	Value	Tolerance
$R_1$	17.4 k $\Omega$	0.1%
$R_2$	239 k $\Omega$ (229 k $\Omega$ +10 k $\Omega$ )	0.1%
$R_3$	200 k $\Omega$	0.1%
$C_1$	0.1 $\mu F$	1.0%
$C_2$	0.01 $\mu F$	1.0%
$C_3$	0.0033 $\mu F$	1.0%

Table 6.7 Sallen-Key filter characteristics

Component	Value
$F_{C1}$	92.16 Hz
$F_{C2}$	126.24 Hz
$\xi$	.6025
$P_1$	-76.06 +100.75i Hz
$P_2$	92.16 Hz
$P_3$	-76.06 -100.75i Hz

This filter is characterized by its constant delay group near the 60 Hz band; this can be observed by Figure 6-20 and Figure 6-21, where the magnitude and group delay response is plotted. The results presented in Figure 6-20 indicate that the signal attenuation near the 60 Hz band is almost constant and remains in the 84-91% range, while in Figure 6-21 the phase delay remains within a  $3.13748 \pm 0.00250$  ms variation for input signals near the 60 Hz band.

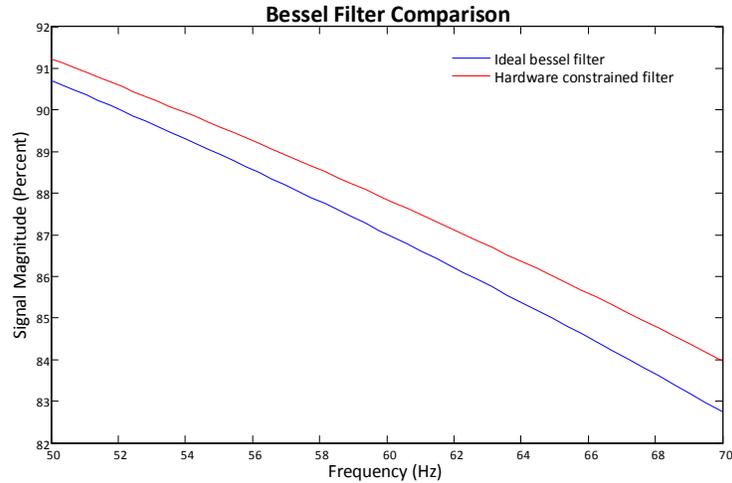


Figure 6-20. Ideal vs hardware constrained Bessel filter magnitude characteristics.

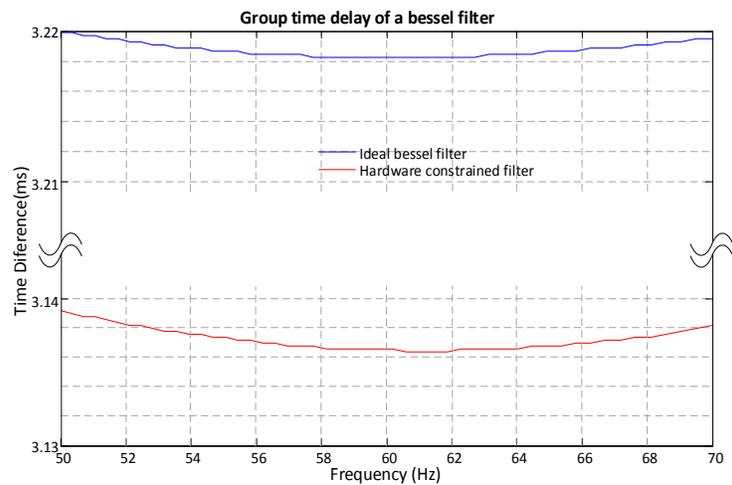


Figure 6-21. Ideal vs hardware constrained Bessel filter group time delay characteristics.

Finally, the transfer function for this filter was obtained by employing Eq. 2.5, achieving the response illustrated by Eq. 6.5.

$$G(s) = \frac{364341729.2}{s^3 + 1534.96s^2 + 1182696.97s + 364341729.22} \quad \text{Eq. 6.5}$$

A further in-depth analysis of this module is continued in section 6.5.11.1 of this thesis.

### 6.5.3.3 DC component Removal

Once the signal has been filtered, a DC component removal is used to create a symmetrical waveform, the filter is implemented by using a simple high pass filter designed to filter out frequencies below 1 Hz (DC signals), the employed circuit can be observed at Figure 6-22.

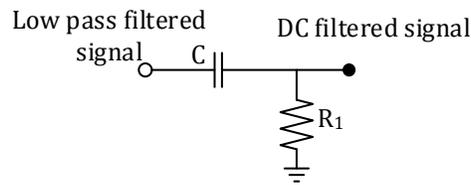


Figure 6-22 DC filter used on the frequency measurement circuit.

#### 6.5.3.4 Square wave generator

Once the signal has been DC filtered an OpAmp-Transistor pair is used to output a square wave (see Figure 6-23), the outputted waveform is designed to be compatible with 3.3 volts logical levels.

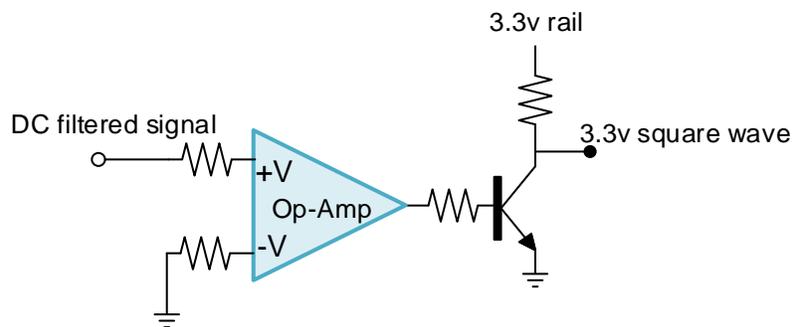


Figure 6-23 Sinusoidal waveform to square waveform conversion circuit.

#### 6.5.4 Dynamic signal sampling mechanism-Hardware Implementation

Phasor measurement units are required to report precise phasor measurements, in certain studies the phasor angle is perhaps the most important aspect and thus requires a high precision. This angle can be affected by a mismatch of the system frequency and sampling frequency, in Figure 6-24 a set of unit phasors with a  $30^\circ$  offset are sampled at a fixed frequency (60 Hz), causing errors on the estimated phasor angles, these angle variations render PMU units useless. Frequency mismatches can be mitigated by digitally resampling the signal (i.e. applying digital filters to the data window), or by dynamically adjusting the sampling frequency. In this thesis, a hardware solution to dynamically adjust the sampling frequency is proposed.

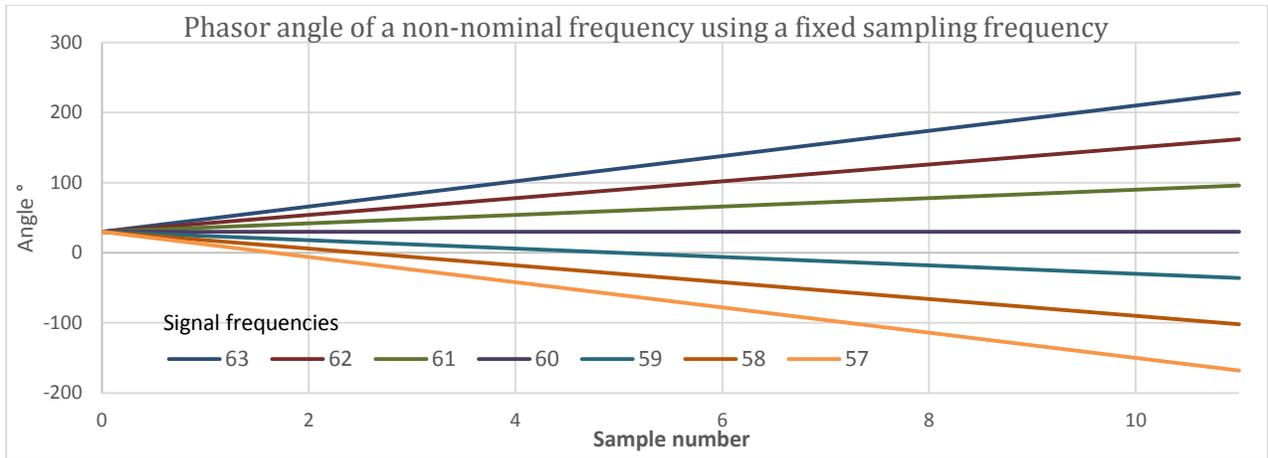


Figure 6-24 Angle variation by a real frequency and sampling frequency mismatch, different signals are sampled at 60 Hz.

The dynamic signal sampling is achieved by using a programmable PLL-VCO chip (controllable in software); this enables the ADC unit to sample signals according to the measured frequency via an open loop control. The PLL-VCO used is the LMK03033C unit, which has several clock divisors that enable to provide sampling frequencies in the range of 1.806-2.117 MHz (see Figure 6-25), by employing Eq. 6.2 the electrical frequencies that can be measured by the ADC-PLL driven-unit are in the range of 55.11 to 64.60 Hz.

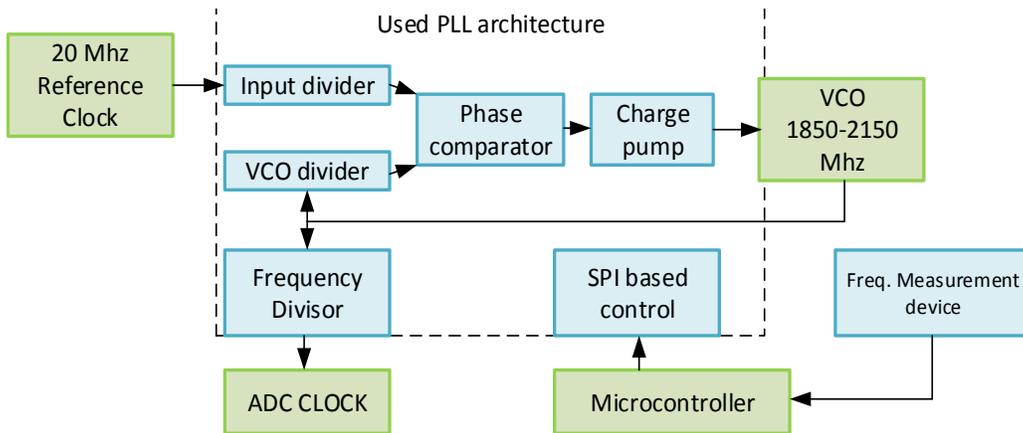


Figure 6-25 PLL driven dynamic clock reference used to drive the ADC unit.

The LMK03033C unit is controllable thru a SPI port that is attached to the metering microcontroller. Internally the PLL is composed of a series of registers that enable multiple frequency tracking mechanisms, the selected mode uses a fixed PLL\_R divisor and a software driven PLL\_N divisor as the frequency control mechanism. The LMK03033C frequency response can be replicated by employing the TI clock design tool as shown in Figure 6-26.

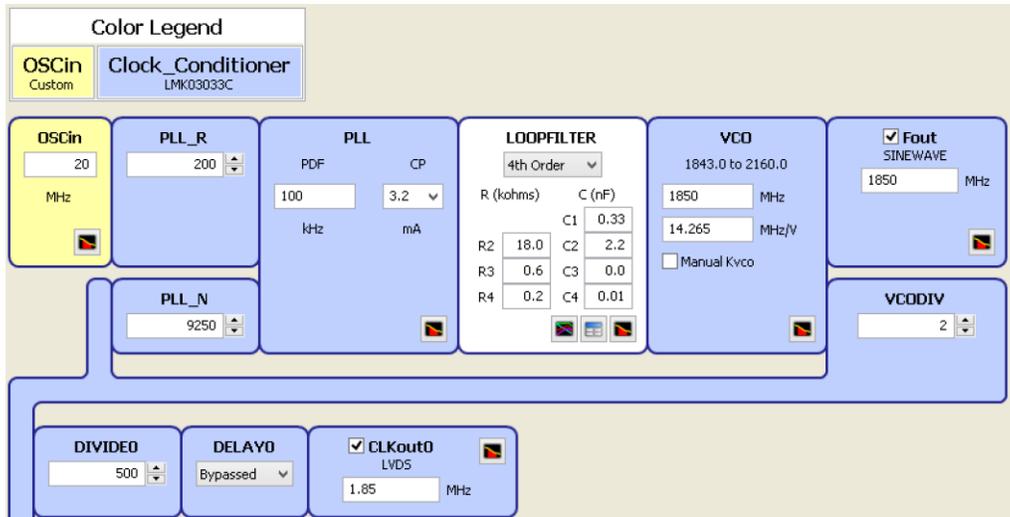


Figure 6-26 Internal PLL values used to simulate the PLL response under the clock design tool available at [137].

### 6.5.5 DFT implementation

Energy measurement is often defined in terms of frequency domain signals, in this section, the author describes the time-optimized implementation designed to transform discrete signal measurements into a time domain representation.

Traditional FFT is the preferred method for signal decomposition, since it is considered an optimized version of the DFT. Although this assertion is mostly true, in certain cases this rule can be broken according to the required number of samples, and hardware characteristics. In the next sections a highly optimized DFT algorithm is proposed that proves to be faster than an off the shelf solution.

#### 6.5.5.1 Analysis of the required operations needed by FFT/DFT implementations.

According to [138] the FFT complexity of a radix-2 algorithm requires  $\left(\frac{N}{2}\right)\log_2 N$  complex multiplications and  $N\log_2 N$  complex additions to compute the result vs the  $(O)N^2$  complexity of a traditional DFT implementation. However this complexity can be lowered for the DFT if the required frequency components is much lower than  $N$ . For example in electrical systems, even harmonics are less severe than odd harmonics, and can be omitted in certain cases. For the IEEE 1459-2000 compliance the author proposes to use 30 interest frequencies to compute the results, this frequencies correspond to all the odd harmonics up to the 51<sup>st</sup> harmonics (1, 3, 5, ..., 51) plus a limited set of even harmonics (2, 4, 6, 8, 10). In Table 6.8 the total number of operations required

to transform discrete time signals into frequency domain signals is estimated according to the sample size ( $N$ ).

Table 6.8 Comparison of the number of operations required to transform discrete time signals into frequency domain quantities by using DFT and FFT

Sample size	DFT ( F=30)		FFT	
	# Real multiplications	# Real additions	# Complex multiplications	# Complex additions
<b>N</b>	<b><math>2N * F</math></b>	<b><math>2(N - 1) * F</math></b>	<b><math>(N/2)\log_2 N</math></b>	<b><math>N\log_2 N</math></b>
16	960	900	32	64
32	1920	1860	80	160
64	3840	3780	192	384
128	7680	7620	448	896
256	15360	15300	1024	2048

Although the results in Table 6.8 clearly show the FFT advantage, these results are misleading in microcontroller environments, since most units do not handle complex numbers directly, and instead use subroutines to perform the complex multiplications in the real number field ( $\mathcal{R}$ ). Complex multiplications can be implemented by employing four multiplications and two additions, as it can be observed by Eq. 6.6 (some optimized versions reduce the total number of operations to five), while complex additions require two additions to complete. By adjusting the number of required operations (considering MCU only handle operations on the  $\mathcal{R}$  field), Table 6.9 was obtained.

$$(a + jb)(c + jd) = (ac - bd) + j(ad + bc) \quad \text{Eq. 6.6}$$

Table 6.9 Comparison of the number of operations required to transform discrete time signals into frequency domain quantities by using  $\mathcal{R}$  field operations

Sample size	DFT ( F=30)			FFT			#operation relationship
	#real multiplications	#real additions	Total # of operations	# Complex multiplications	# Complex additions	Total # of operations	
<b>N</b>	<b><math>2N * F</math></b>	<b><math>2(N - 1) * F</math></b>	<b><math>T_{DFT}</math></b>	<b><math>6 * (N/2)\log_2 N</math></b>	<b><math>(N)\log_2 N</math></b>	<b><math>T_{FFT}</math></b>	<b><math>T_{FFT}/T_{DFT}</math></b>
16	960	900	1860	192	128	320	0.17
32	1920	1860	3780	480	320	800	0.21
64	3840	3780	7620	1152	768	1920	0.25
128	7680	7620	15300	2688	1792	4480	0.29
256	15360	15300	30660	6144	4096	10240	0.33

Some microcontrollers (like the PIC32MZ) are able to handle multiply and accumulate operations (MAC, MADD) into single cycle operations, therefore lowering the total amount of operations; by using this hardware dependent optimization, a new table was constructed to reflect the minimum amount of operations required, the results are reported Table 6.10. As it can be observed from this table, certain operations can be removed by combining them; this leads to time optimizations on both the FFT and DFT algorithm (only those multiplication operations that are immediately followed by an addition are reducible).

Table 6.10 Comparison of the number of operations required to transform discrete time signals into frequency domain quantities by using  $\mathcal{R}$  field operations under a MAC enabled unit

Sample size	DFT ( F=30)		FFT			#operation relationship
	#real MADD operations	Total # of operations	# Complex multiplications (using MADD)	# Complex additions	Total # of operations	
<b>N</b>	<b><math>2N * F</math></b>	<b><math>T_{DFT}</math></b>	<b><math>4 * (N/2)\log_2 N</math></b>	<b><math>(N)\log_2 N</math></b>	<b><math>T_{FFT}</math></b>	<b><math>T_{FFT}/T_{DFT}</math></b>
16	960	960	128	128	256	0.26
32	1920	1920	320	320	640	0.33
64	3840	3840	768	768	1536	0.40
128	7680	7680	1792	1792	3584	0.46
256	15360	15360	4096	4096	8192	0.53

Although the results shown in Table 6.10 indicate that for a data window of size  $N = 128$  the FFT implementation offers an almost 2X advantage over the number of required operations, its internal construction can impose optimizing difficulties at the hardware level, and therefore the DFT method was selected for further optimization due to its simplicity.

#### 6.5.5.2 Optimization of a cross correlation function to compute the DFT

As mentioned by the chapter introduction code optimization at the hardware level is based on two main aspects:

- Memory use optimization (i.e. adequate cache use, amount of memory employed)
- CPU optimization (i.e. Pipelining techniques, Assembler language)

These hardware optimization techniques should be mixed with appropriate programming procedures that enable the programmer to keep a maintainable code (i.e. using *for* sequences instead of hardcoded multiplication vectors)

By using the before mentioned techniques an assembler based cross-correlation algorithm was developed according to the following principles

- First, a loop unrolling technique was employed that lowers the amount of control overhead by grouping common operations into a single block
- Indirect addressing was exploited to access the data vectors, with precomputed offsets (i.e. hardcoded addresses)
- A pipelined optimized code was achieved by checking for data collisions and verifying wait slots (see Table 6.4)

The final algorithm works on data chunks of eight 32-bit integers, by constantly multiplying the digitalized signal ( $S_i$ ) with a Lookup Table (LUT) containing the DFT factors associated with each harmonic ( $h$ ). The operation  $\{\sum_{i=1}^n S_i LUT_h(i)\}$  is done according to the procedure described by Figure 6-27. This optimized version performs on average a MAC instruction every 3.5 cycles, (considering the function call overheads), and allows to compute the correlation factor of signal containing 128 samples in 448 instructions/clock cycles (Numbers obtained by setting  $N = 128$ ).

Although the solution only implements the cross correlation function, it can be adapted to provide the correlation functionalities of a DFT algorithm by obtaining two correlation indexes, one for the sine function and the other for the cosine function, by using this concept the number of required operations to obtain a complete DFT decomposition can be approximated to 900 instructions.

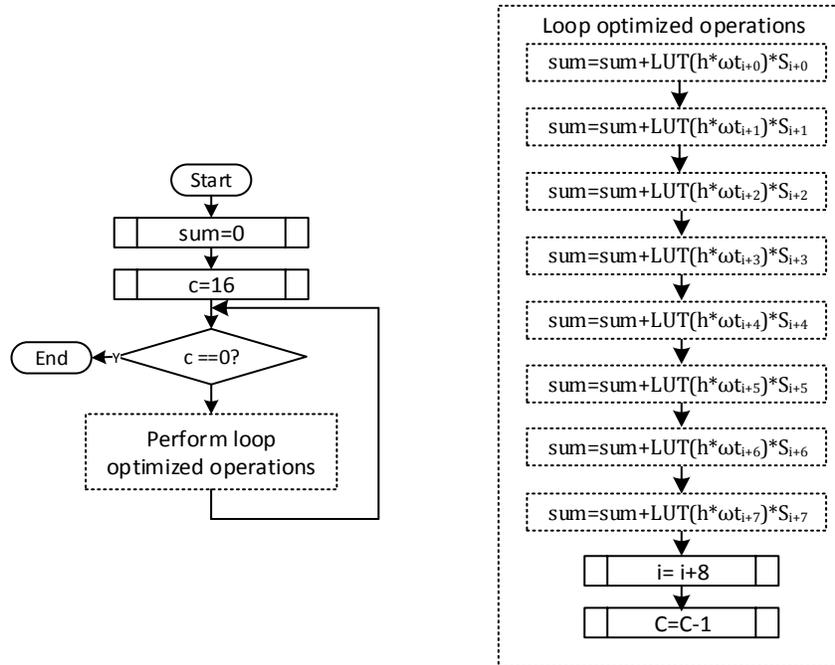


Figure 6-27 DFT calculation algorithm, featuring a loop unrolling optimization,  $N = 128$ .

### 6.5.5.3 DFT based signal decomposition-The square root problem

By using, the subroutine proposed on the previous section, and using fixed-point math techniques the signals begin digitalized by the ADC unit can almost be transformed into time domain signals, however to complete this procedure the cross-correlated signals must be first normalized. This process can be done according to the scaling function that depends on the ADC bit resolution, LUT characteristics and scaling circuits (see Eq. 6.7).

$$B_h = \sum_{i=1}^n S_i LUT_{\sin\_h}(i) * (Sf) \quad \text{Eq. 6.7}$$

$$C_h = \sum_{i=1}^n S_i LUT_{\cos\_h}(i) * (Sf)$$

With the  $B_h$  and  $C_h$  factors, the signal phasor characteristics can be calculated by applying the magnitude equation found in Table 2.2, which brings up the square root problem. The square root can be calculated by many methods, some well-known algorithms are the Newton and the Babylonian method. The Babylonian method is sometimes used on microcontroller courses due to

its simplicity and is illustrated by Eq. 6.8; it relies on successive division processes that are computationally expensive in most MCU environments.

$$X_{n...} = \sqrt{y} \tag{Eq. 6.8}$$

$$X_{n+1} = \frac{X_n + y/x_n}{2}$$

In this section an improved algorithm is presented, it is based on using multiplications instead of divisions, the general procedure for obtaining square roots up to 0xFFFE0001 (32 bits) is given in Figure 6-28. This algorithm works by constantly evaluating if the square of a proposed value ( $P_v$ ) is greater than the target value, the  $P_v$  value is constantly altered by clearing and setting bits in an orderable fashion (from the MSB to the LSB).

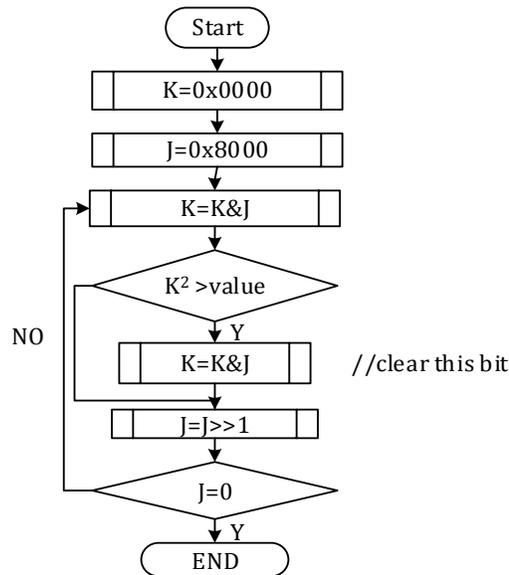


Figure 6-28 Fast Square root algorithm based on successive multiplications

Since the PIC32MZ unit offers a 32-bit-by-32-bit, multiplier it is feasible to develop a program that is able to solve square roots for numbers up to 64 bits. Although the usability of such a large square root capability can be considered useless, this capability can be used to add precision to results involving relatively small numbers, by employing Eq. 6.9. This equation enables to perform bit shifts that alter the operand value that result in additional bits that can be used to obtain decimal values (by performing inverse bit shifting).

$$x = \sqrt[n]{y} = \frac{\sqrt{y * 2^n}}{n/2} \quad \text{Eq. 6.9}$$

The proposed square root algorithm was tested for precision and total execution time, the results were validated by using a set of test vectors presented in Table 6.11. In Table 6.11, the exact square root values of some proposed values are given, these were calculated using an IEEE 754 quad-precision program developed in FORTRAN.

Table 6.11 Sample values for testing the proposed method, showing the exact values.

Test #	Hexadecimal (unsigned)	Decimal value	Bits	Exponential representation	Exact value (128-bits)
1	0xFFFFFFFFE0000001	18,446,744,065,119,600,000	64	1.8447E+19	4294967294.9999900
2	0X000FFFFFFFFFFFFF	281,474,976,710,655	48	2.8147E+14	16777215.9999999
3	0X0000000FFFFFFFFF	4,294,967,295	32	4.2950E+09	65535.9999924
4	0X000000000000FFFF	65535	16	6.5535E+04	255.9980469

In Table 6.12 the results of implementing the proposed algorithm in C language and assembler (ASM) are presented. In this particular case, the ASM implementation was hardware optimized by the same techniques explained in section 6.5.5.2. The results indicate that a higher precision is achieved than with the pure Babylonian method (written in C), with an additional 5X execution time improvement.

Additionally the number of cycles on the “insertion” method remain constant independently of the test value, this programming technique is known as “fixed-response time” [139] and is particularly useful for implementing functions that are resistant to timing attacks (e.g. AES implementations), in this case the functions were programmed with this technique in mind to determine a fixed DFT computation time.

Table 6.12 Time and precision comparison of the proposed method against the Babylonian method

Test value	Insertion (C)		Babylonian method (C)		Insertion method (ASM)		Insertion method with increased precision (ASM)		
	#Cycles	Result	#Cycles	Result	#Cycles	Result	#Cycles	Result	Relative error (%)
1	1644	4294967295	1402	4294967295	558	4294967295	574	4294967295	-2.33E-13
2	1718	16777216	2358	16777216	558	16777215	574	16777215.996100	2.32E-08
3	1750	65536	3174	65536	558	65535	574	65535.9999847	1.17E-08
4	1760	255	3792	255	558	255	574	255.9980316	5.95E-06
Average #Cycles.	1718		2681		558		574		

#### 6.5.5.4 DFT validation (Digital simulation)

The DFT algorithm validation was evaluated by performing the signal decomposition, of a current and voltage waveform with known harmonic contents [61]. These particular waveforms are composed of odd order harmonics that represent the effect of non-linear loads, a plot containing the signal waveform can be viewed at Figure 6-29.

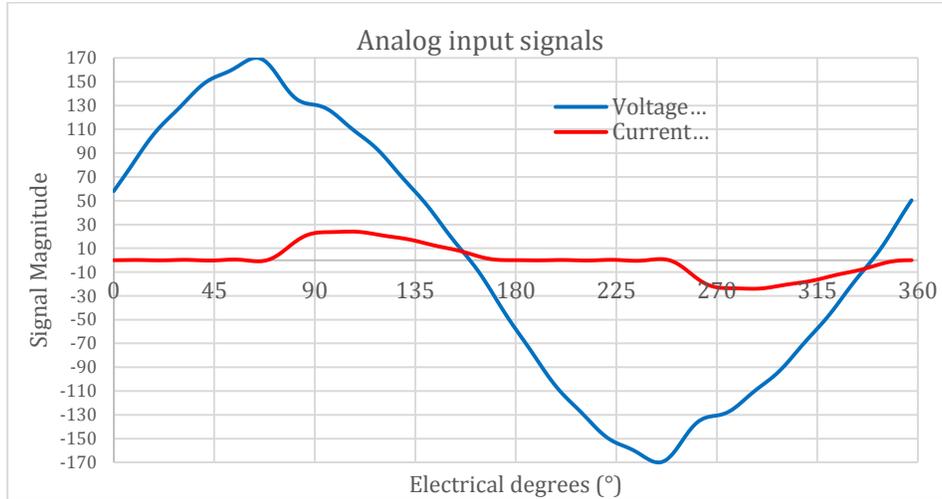


Figure 6-29 Sample current and voltage signals proposed by [61]

The discretized signal values were generated by simulating an ideal ADC unit that is configured as per Table 6.13, this virtual ADC unit was designed to test the effects of low level signals being digitalized, resembling the low test currents used to certify meter accuracy according to ANSI C12.10 standard.

Table 6.13 ADC voltage and current configurations employed by the digital simulation.

Voltage channel configuration		Current channel configuration	
$V_{max}$	1700 V	$I_{max}$	240 A
$V_{min}$	-1700.0129 V	$I_{min}$	-240.00183 A
ENOB	18	ENOB	18
Number of discrete steps	131071	Number of discrete steps	131071
Resolution	0.01297006 V/step	Resolution	0.00183106 A/step

With the ADC configuration given in Table 6.13, the test signals provided by IEEE were digitalized by setting the signal frequency at 60 Hz and obtaining 128 samples per cycle, the signal amplitude of both signals represent at the most 10% percent of its useful range (see Figure 6-30). This creates

a stress on the numerical stability of the proposed algorithms, since they are based on fixed-point math that can lead to truncation and thus has the capability of maximizing errors.

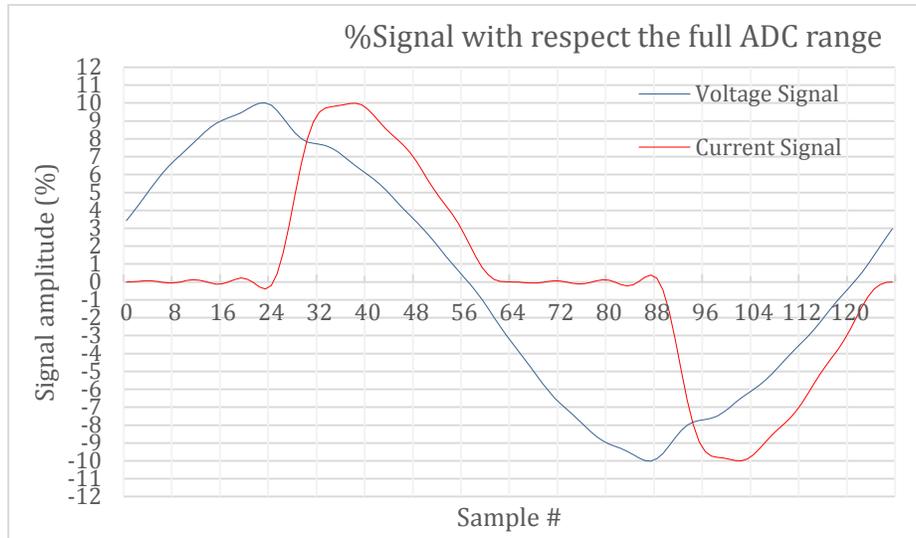


Figure 6-30 Discretized test signals given by [61], expressed on % respect the nominal ADC channel configuration.

In Table 6.14 and Table 6.15 the results for the magnitude of the voltage and current signals are reported, in this case the obtained harmonics are compared with those given by IEEE on [61]. As it can be observed from the results, both signals are correctly decomposed, the implemented DFT method show a higher relative error than those results obtained by using IEEE 758 floating point math. These errors can be associated with the effects of using fixed-point math vs the dynamic nature of floating point math.

Table 6.14 Harmonic contents (magnitude) obtained from the sample voltage signal

Harmonic	IEEE reported value	DFT (obtained)	FFT (obtained)	Relative error % (DFT)	Relative error% (FFT)
1	110.09	110.09015	110.09012	-0.0001363	-0.0001090
3	6.583382	6.58335	6.58321	0.0004861	0.0026127
5	1.915566	1.91563	1.91559	-0.0033409	-0.0012529
7	2.2018	2.20168	2.20153	0.0054504	0.0122642
9	1.266035	1.266032	1.26602	0.0002370	0.0011848
11	1.255026	1.25502	1.25502	0.0004781	0.0004781
13	0.935765	0.9352	0.93568	0.0604149	0.0090843
15	0.979801	0.97921	0.97935	0.0603548	0.0460510
17	0	0.00074	0.00074	*	*
19	0	0.00163	0.00163	*	*
21	0	0.00097	0.00097	*	*
23	0	0.00030	0.00029	*	*

Table 6.15 Harmonic contents (magnitude) obtained from the sample current signal.

Harmonic	IEEE reported value	DFT (obtained)	FFT (obtained)	Relative error % (DFT)	Relative error% (FFT)
1	11.17	11.17009	11.17008	-0.0008057	-0.0007162
3	5.71904	5.71911	5.71911	-0.0012240	-0.0012240
5	1.427526	1.42754	1.42754	-0.0009807	-0.0009807
7	1.392899	1.39299	1.39297	-0.0065327	-0.0050970
9	0.600946	0.60096	0.60096	-0.0023296	-0.0023296
11	0.615467	0.61548	0.61548	-0.0021122	-0.0021122
13	0.346270	0.34628	0.34625	-0.0028878	0.0057762
15	0.339568	0.33956	0.33953	0.0008835	0.0111919
17	0	0.00017	0.00017	*	*
19	0	0.00010	0.00010	*	*
21	0	0.00004	0.00004	*	*
23	0	0.00009	0.00009	*	*

The developed DFT algorithm was time benchmarked with the one given on [140]; the open source code was ported to the PIC32MZ compiler environment and optimized by altering the following parameters

- Transform the twiddle calculation procedure into a one that uses a LUT
- Elimination of recursive functions
- The program initialization was moved into a separate routine that is not timed
- The compiler optimization level was set to-O1 (basic optimization, limited by the compiler license)

The total number of clock cycles required to perform the signal decomposition process is given in Table 6.16, where a clear advantage can be seen on the developed DFT algorithm. If the program suggested by [140] were to be implemented on a production version, it would not be able to process all the readings, considering that a typical CPU has a 16.666 *ms* to perform energy measurements on 60 Hz signals.

Table 6.16 Execution times obtained by the FFT and DFT algorithms implemented on the PIC32MZ

	FFT (C)	DFT (ASM)
Clock cycles	287858 (average value)	59742*
Execution time @ 80 MHz	3.59823 <i>ms</i>	746.77 <i>us</i>
Total execution time (6 signals)	21.5894 <i>ms</i>	4.48065 <i>ms</i>

\*The number of cycles is fixed and includes the DFT correlation + DFT magnitude + DFT angle computation (LUT based)

### 6.5.6 IEEE 1459-2000 implementation

With the phasors obtained in the previous sections, power consumption calculations were performed according to IEEE 1459-2000 standard. The functions to calculate the IEEE quantities were performed using traditional floating-point math by a program written in C, the results are presented in Table 6.17. As it can be observed, the fundamental power measurements are well within the 0.5% accuracy levels required by the standard (compared against those reported on annex A of [61]).

Table 6.17 Comparative results of the IEEE sample calculations and those implemented by the author.

Unit	IEEE reported value	Calculated value	Relative error (%)
$v_1$	110.09	110.09015	0.00013
$i_1$	11.17	11.1709	0.00805
$v_h$	7.55	7.569234803	0.25476
$i_h$	6.15	6.144415772	-0.09080
V	110.35	110.348157	-0.00167
I	12.75	12.744787	-0.04088
$THD_V$	0.069	0.068754878	-0.35524
$THD_I$	0.549	0.550037667	0.18901
$P_1$	836.9288*	836.93671	0.00094
$P_H$	-40.08822	-40.10496536	0.04177
$P$	815.90595	796.8317447	-0.00110
$Q_1$	898.54179	900.9664573	0.26984
$S_1$	1229.7	1229.71688	0.00137

\*Corrected value from the one published on the standard

### 6.5.7 IEEE C37.118 implementation

Once the measurement software modules were validated an additional validation was performed on the phasor angles obtained by the proposed DFT algorithm, these results are expressed in terms of the absolute error and are shown in Table 6.18

The phasor angles are obtained by performing a single division operation between the sine and cosine components obtained from the DFT cross-correlation process ( $B_h/C_h$ ). The absolute value of the quotient is used as an indexer of a LUT table that contains a mapping function  $\{atan(\theta) = \circ\}$ , see Table 6.19. In certain cases, additional operations are performed to correct the angle quadrant.

Table 6.18 Harmonic contents (angle) obtained from the sample waveforms given in [61]

Harmonic (h)	$\angle\theta$ as reported by IEEE	$\angle\theta$ obtained by DFT	Absolute error (°)
1	23.8	23.91	0.11
3	-39.12	-39.20	-0.08
5	173.3	173.32	0.02
7	21.39	21.45	0.06
9	-120.1	-120.23	-0.13
11	87.24	87.36	0.12
13	-60.68	-60.82	-0.14
15	156.4	156.45	0.05

Table 6.19 Angle LUT used to compute the phasor angles

Position	Unsigned Q(16.16) (indexer)	Result
0	0x0000 0000	0.00°
1	0x0000 00CE	0.18°
2	0x0000 0190	0.35°
...	...	...
256	0x0000 FFFF	45.00°
...	...	....
509	0x006C 1A27	89.47°
510	0x00A3 B33F	89.65°
511	0x013E 4F10	89.82°

#### 6.5.7.1 GPS time signal characterization

GPS receiver units are often used by PMU units to provide a reference time, this synchronization process is often done by a pulse emitted every second (PPS) by the receiver unit, the receiver unit is then read by the MCU retrieving the timestamp that was valid when the pulse was emitted.

Although GPS receiver units offer great time precision capabilities these can be affected by adverse weather effects, signal interference caused by near buildings and the receiver quality. GPS signals are transmitted at a rate of 50 bits/s; and contrary to common belief, the time stamp is only transmitted every 6 seconds by each satellite [141], meaning that the receiver unit is responsible for creating a self-generated pulse, based on the received time stamps from other GPS satellites.

The employed receiver unit was the *EB – 5365RE* unit from Globalsat Corporation. This unit features a Satellite Based Augmentation System (SBAS) that enables to perform Ionospheric corrections by analyzing a special sequence transmitted along the main data stream. These

Ionospheric corrections are uploaded into the satellites by earth located stations that constantly monitor the atmospheric conditions; one of such stations is located on Mexico City [142]

In Figure 6-31 two cases are presented showing the time difference between two GPS receiver units, in this case the readings were obtained in Mexico city and offer a  $\pm 20$  ns time difference (which can be considered low), however this time difference can vary significantly depending on the climate, obstacles and availability of nearby SBAS stations. In Figure 6-32 the time difference for 502 sample time differences is shown, this particular graph was obtained by turning off the SBAS system (via internal register configurations) on a cloudy day.

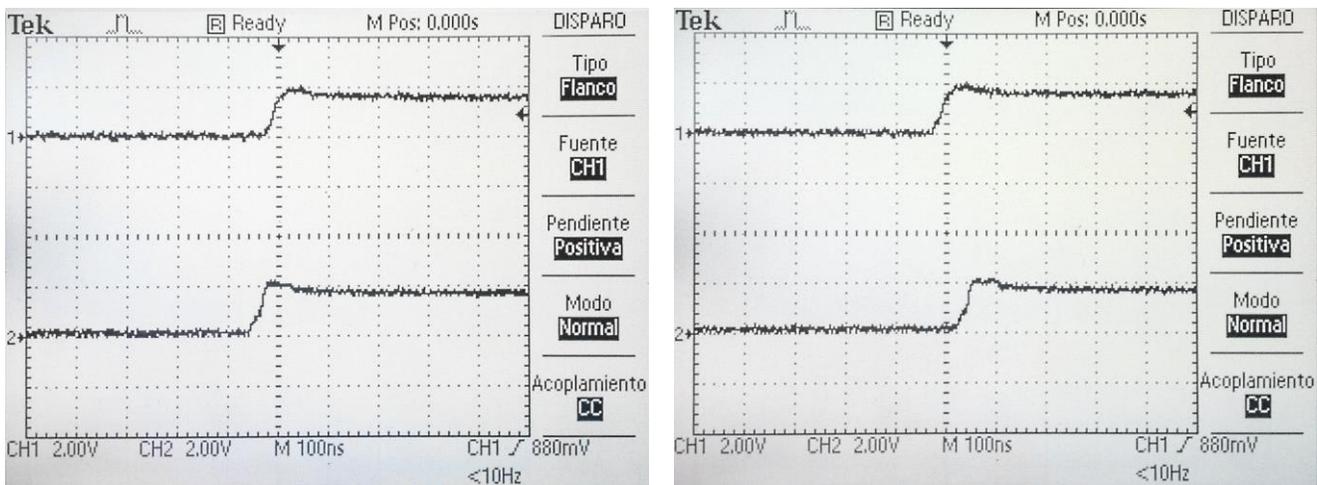


Figure 6-31 Time synchronization measurements obtained from two GPS units, showing time differences

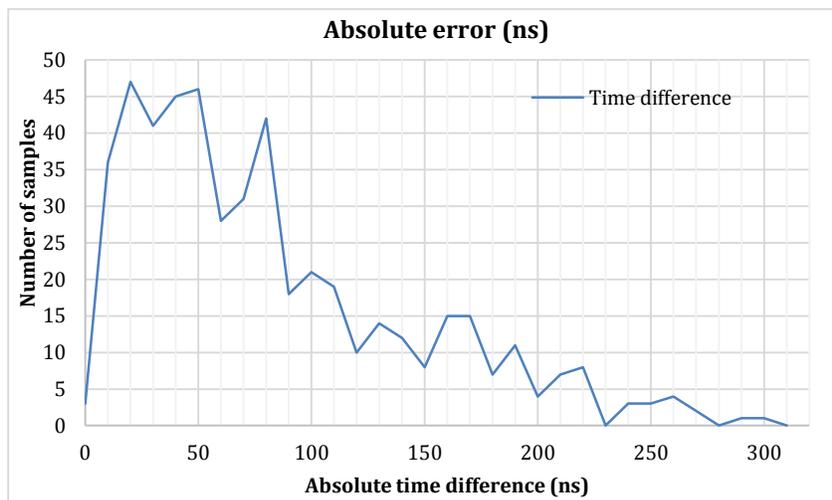


Figure 6-32 Time difference between two GPS units (absolute) organized through a histogram

The readings obtained in Figure 6-32 resemble those published by the national Coordination Office for Space-Based Positioning, Navigation and Timing organization (located on the US) [143]. The published readings for the absolute horizontal position error is shown in Figure 6-33.

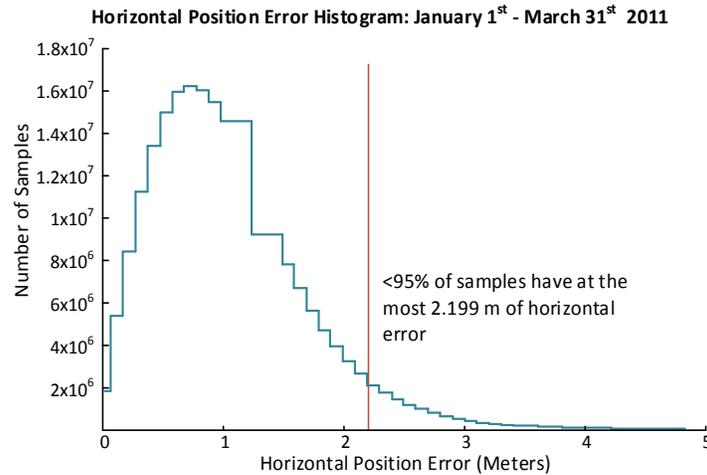


Figure 6-33 Absolute position error histogram reported by the US government, for a large number of samples [143]

With the readings obtained in Figure 6-32, Table 6.20 was created based on the observed average, minimum and maximum time differences, these time differences are used to estimate the approximate error differences in degrees if an electrical signal was time stamped with this timing error.

Table 6.20 Time stamp differences between two GPS units and their associated electrical angle error.

<b>Condition</b>	<b>Time difference (ns)</b>	<b>Error (°) 1<sup>th</sup> harmonic</b>	<b>Error (°) 3<sup>th</sup> harmonic</b>	<b>Error (°) 5<sup>th</sup> harmonic</b>	<b>Error (°) 49<sup>th</sup> harmonic</b>
Maximum	290	0.00630	0.01892	0.03153	0.30905
Minimum	0	0.00000	0.00000	0.00000	0.00000
Average	130	0.00280	0.00842	0.01404	0.13759

It is important to note that not all GPS units offer synchronous PPS signals, an example of such a chips is the Global Sat EB-365 or any module powered by SiRF v3.0 (Garmin™ and some Trimble™ models [144])

### 6.5.7.2 Phasor measurement unit testing procedure.

In Figure 6-34 a flowchart for the proposed PMU software architecture is presented, this flowchart presents the two asynchronous processes that must be executed in order to provide time-stamped measurements. The first process is responsible for transferring the ADC readings via a DMA enabled channel, registering the exact time at which the reading was obtained (via an input capture channel), when the cycle capture is complete ( $N = 128$ ) the measurements vector is tagged with temporal time information. The second process is responsible for assigning the correct time information based on the system frequency, measurement time tags and GPS time stamp time.

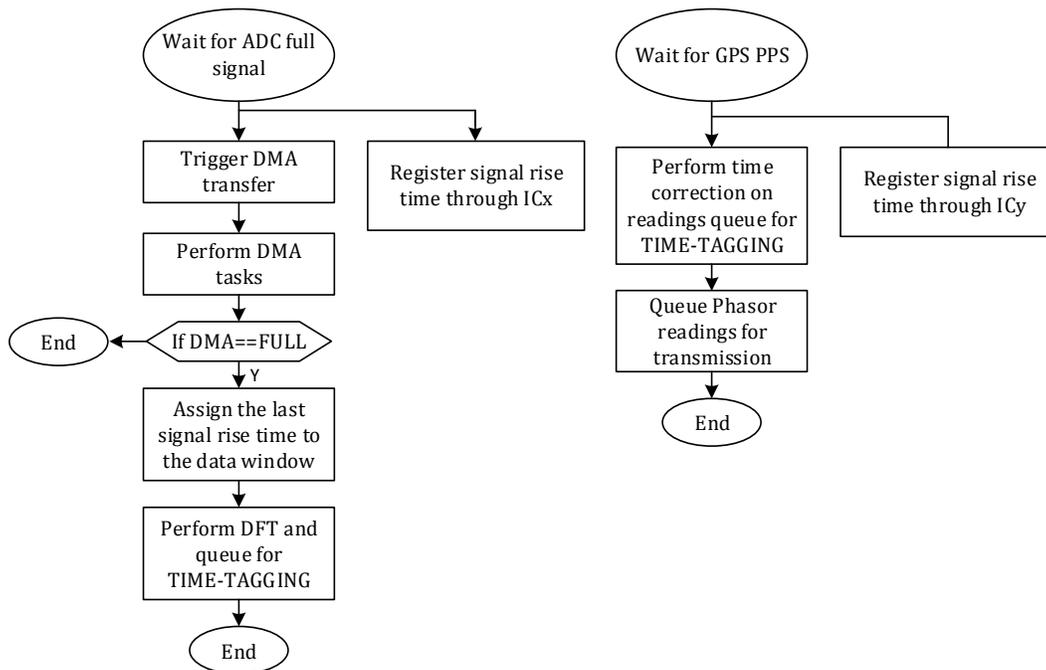


Figure 6-34 Time-stamped measurement flowchart

PMUs are often tested under dynamic operations, however for this particular project the phasor measurement unit was evaluated under steady state conditions. The PMU characteristics were evaluated by using two identical devices attached to the same physical power outlet, for this test the results were captured with the phasor and appropriate ADC data, in Figure 6-35 the signal used to evaluate the PMU characteristics is illustrated.

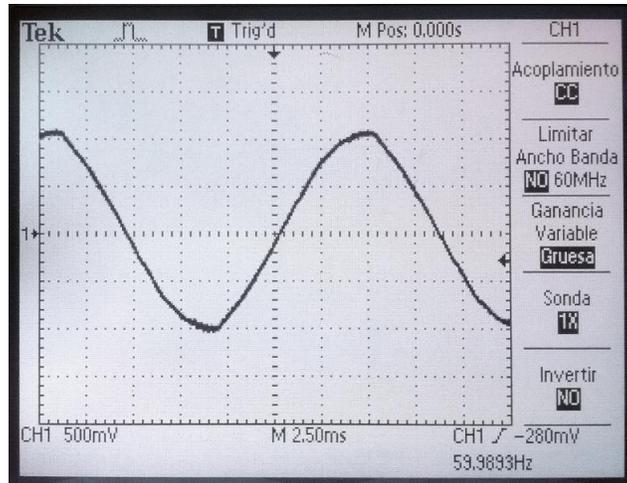


Figure 6-35 Sample voltage found at the university premises.

To evaluate the signal acquisition process the signal values acquired by the ADC of a PMU unit were overlapped with the image shown in Figure 6-35, creating Figure 6-36. Although the signal characteristics match, some errors can be observed due to a misalignment on the image.

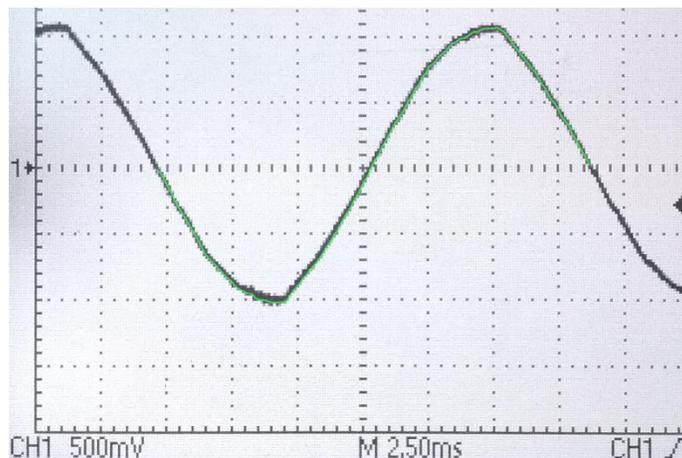


Figure 6-36 ADC measured signal (digitally added signal, based on the obtained readings).

Once the signal digitalization process was validated (graphically), the phasor results obtained from two PMU units (PMU #1 and PMU #2) were compared according to their magnitude and angle components; the results are shown in Table 6.21 (The reported results use a distinct *ATAN* function from the one reported in Table 6.19).

Table 6.21 Signal characteristics obtained from two PMU units that are connected to the same power outlet.

h	PMU unit #1		PMU unit #2		ABS(V <sub>1</sub> - V <sub>2</sub> )	$\frac{V_1 - V_2}{V_2}$	ABS(A <sub>1</sub> - A <sub>2</sub> )
	V <sub>1</sub>	A <sub>1</sub>	V <sub>2</sub>	A <sub>2</sub>			
1	127.28	-91.50	127.27	-91.75	0.0065	0.0051	0.25
3	2.6337	112.00	2.6379	111.50	0.0041	-0.1574	0.50
5	1.6970	-10.25	1.6941	-10.50	0.0029	0.1710	0.25
7	1.2253	167.50	1.2208	168.00	0.0044	0.3622	0.50
9	0.4495	33.75	0.4409	34.00	0.0087	1.9474	0.25
11	0.6256	-122.00	0.6260	-121.50	0.0005	-0.0767	0.50
13	0.4346	70.00	0.4441	69.75	0.0095	-2.1625	0.25
15	0.3809	48.50	0.3890	49.00	0.0081	-2.1015	0.50
17	0.4055	169.00	0.3985	169.00	0.0070	1.7314	0.00
19	0.1110	40.50	0.0997	40.50	0.0114	10.8031	0.00
25	0.0603	-163.25	0.0731	-163.25	0.0128	-19.2233	0.00
49	0.0623	83.00	0.0579	83.50	0.0044	7.2755	0.50

Where:

- A<sub>i</sub> Angle in electrical degrees
- V<sub>i</sub> Voltage in volts.
- ABS Absolute function

With the reported phasor values, a PC-based transformation was done to transform the phasor measurements into rectangular coordinates; by applying the Eq. 2.26 the Total Vector Error was determined. As it can be observed, the results are within the tolerance levels required by IEEE C37.118, and could be compliant with signals that represent up to 0.5% of the nominal signal value.

Table 6.22 Total Vector Error computed from the readings obtained in Table 6.21

h	X <sub>r</sub>	X <sub>I</sub>	$\widehat{X}_r$	$\widehat{X}_I$	TVE %	% Respect nominal (127 volts)
1	-3.33	-127.24	-3.89	-127.22	0.4363	100.22
3	-0.99	2.44	-0.97	2.45	0.8878	2.07
5	1.67	-0.30	1.67	-0.31	0.4683	1.34
7	-1.20	0.27	-1.19	0.25	0.9453	0.96
9	0.3737	0.2497	0.3655	0.2465	1.9614	0.35
11	-0.33	-0.53	-0.33	-0.53	0.8753	0.49
13	0.1486	0.4084	0.1537	0.4167	2.2300	0.34
15	0.2524	0.2853	0.2552	0.2936	2.3022	0.3
17	-0.3980	0.0774	-0.3912	0.0760	1.7263	0.32
19	0.0844	0.0721	0.0758	0.0647	10.1802	0.09
25	-0.0577	-0.0174	-0.0700	-0.0211	21.2272	0.05
49	0.0076	0.0618	0.0066	0.0575	7.1125	0.05

### 6.5.8 I/O control

Traditional programming of unsupported protocols is usually done by using “bit banging” techniques, which involve driving output pins via software. For devices requiring low data bandwidths, optimized algorithms can be done using interrupts, thus freeing up processing

resources [145]. Low-resolution b/w LCD devices can be considered an example of low bandwidth devices since their refresh rate is quite slow, and graphical data is in the order of kB [146]. Many modern microcontrollers include a hardware LCD driver, freeing up processor resources, in case of the PIC32MZ series this software-hardware module can be implemented by the use of the Parallel Master Port (PMP), the Extended Bus Interface (EBI), or by the DMA module. The DMA module was selected for its simplicity and nonexistence of software overhead during calls, the DMA unit, as mentioned earlier can be auto triggered by the internal timers. The LCD driver is based on using a series of PORTxCLR, PORTxSET register operations, and its architecture can be observed in Figure 6-37.

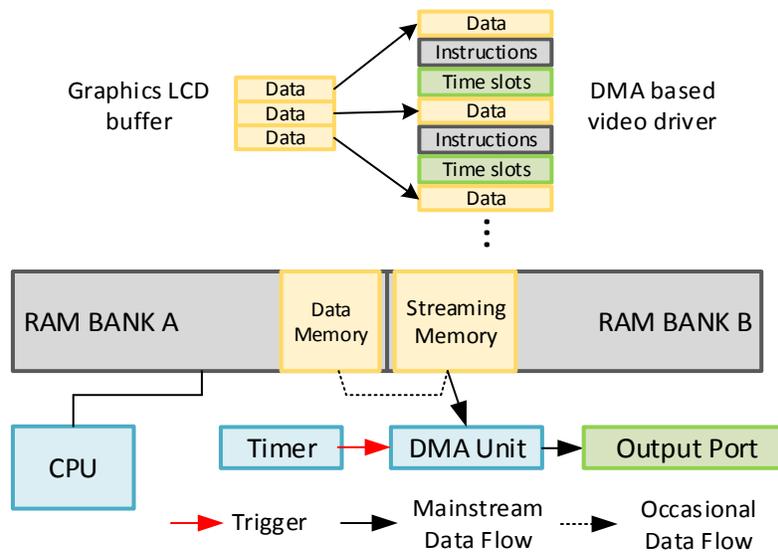


Figure 6-37 Streaming data to devices using a DMA Unit

DMA intensive operations can result in CPU stalling due to concurrent RAM access (bus arbitration schemes). In order to minimize these events the streaming memory was placed in an alternate RAM bank that enables the CPU to exclusively use the first 256 kB of RAM. By employing the aforementioned technique a fast graphics LCD-DMA driver was developed. This driver outlives most of the bit-banging process to the DMA unit, the LCD subroutine only transfers 16x64 bytes (1kB) to the DMA driver to accomplish a single screen refresh. This greatly reduces the processing time vs traditional interrupt driven solutions. In Figure 6-38 the result of implementing this solution is presented for the selected LCD module, which employs the ST7565 chipset, this particular LCD controller performs screen refreshes in horizontal pattern.



Figure 6-38 Sample LCD display by employing a DMA based video driver

### 6.5.9 *PUF generation and recovery algorithm*

The technique described in section 3.7.3 was implemented in the microcontroller firmware by using a two-part process. The first part of this process is called the “PUF generation” and it is executed during the MCU power-up sequence (booting). During the boot process a RAM data capture is stored into a predefined ROM space, until a sufficient amount of data is available for analysis (16 boot sequences). Once the RAM data has been captured a byte analysis is executed inside the device, this analysis performs the identification and subsequently stores the appropriate byte positions required by the PUF recovery algorithm. As a last step, the device credentials are generated and stored in encrypted form inside the FLASH unit along with verification codes that enable PUF validation. In an attempt to improve security, the PUF generation algorithm is erased from the device, replacing the code section with the appropriate PUF validation subroutine (preventing device firmware reset attacks).

The second part of this process is executed after the PUF locations have been determined, in this case, the firmware acquires and validates the ID according to the stored byte positions (PUF). A successful validation results in the decryption of the stored credentials, and subsequent transfer to RAM, the two processes are illustrated in Figure 6-39

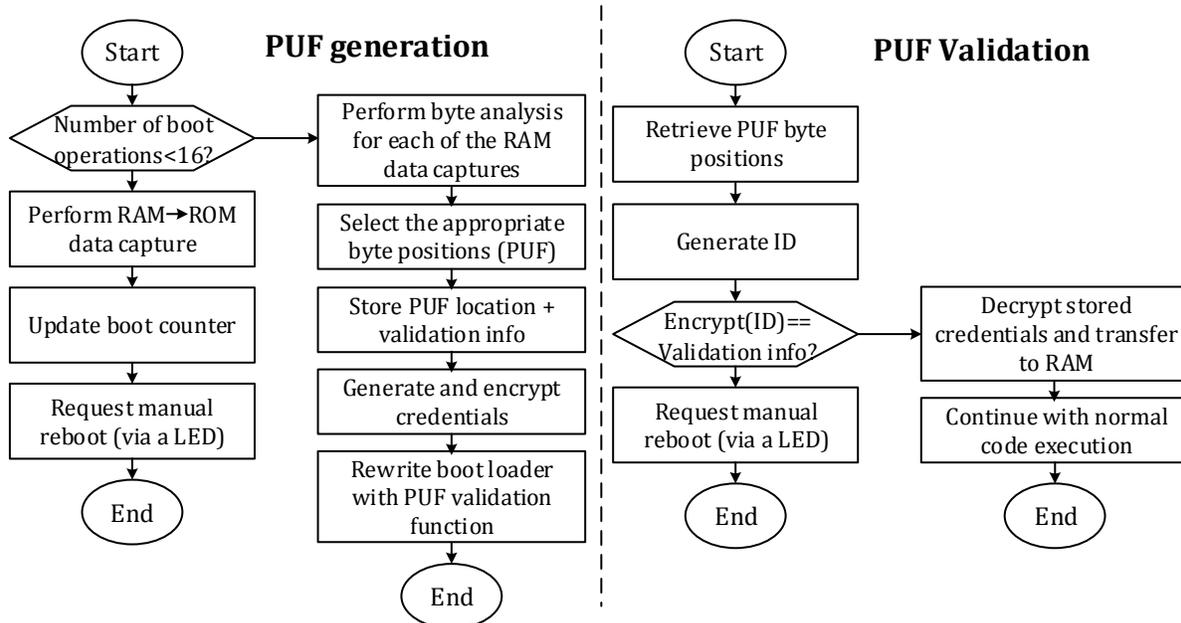


Figure 6-39 PUF generation algorithm (Actual Firmware implementation)

Based on a limited testing done to the PUF recovery algorithm where 50 boot sequences were analyzed, no manual reboot requests were observed. However, since PUF recovery errors are likely to occur on production hardware the LED output should be connected to a power switch that effectively performs a hard reset of the unit.

#### 6.5.10 AES implementation

As mentioned earlier in Chapter 3 an AES implementation was performed to be included in the TLS protocol suite. This implementation was written in assembly and features the following characteristics:

- Support for 128-bit encryption and decryption
- Constant time execution due to equal branch conditions timing and cache unit disable
- Adequate use of the pipeline to evade interlocks
- Interrupt disable in a per block basis to keep a constant time under all operating conditions
- Round 10 includes a time delay to compensate for the MixColumns missing operation.

The block diagram for this particular implementation is given by Figure 6-40.

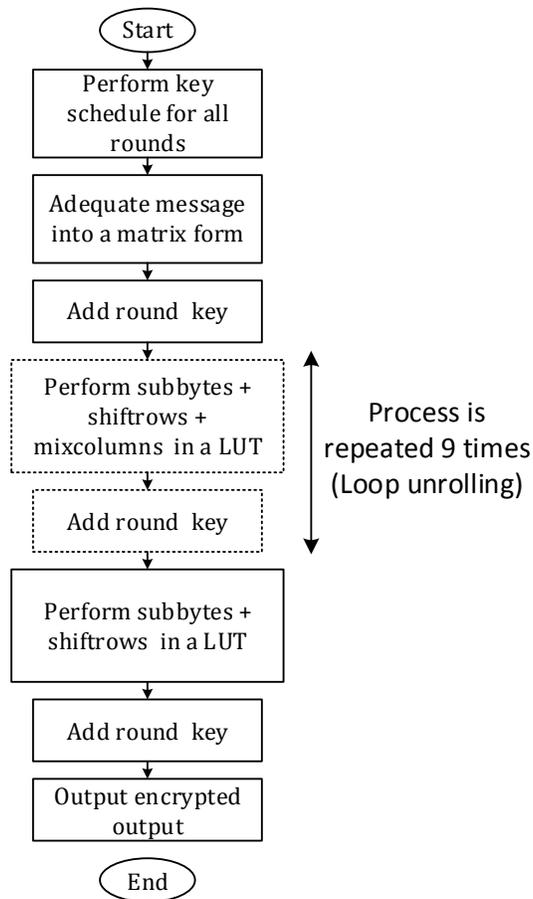


Figure 6-40 Proposed AES implementation flow chart algorithm.

Each of the execution blocks was timed according to the number of operations needed by the CPU, in Table 6.23 the execution times are shown according to each of algorithm steps. Similarly in Table 6.24 the results of bulk encryption are shown for the proposed algorithm vs a one implemented in C, from this table it can be seen that the proposed algorithm matches the C implementation even though the cache unit is disabled (slower execution speeds)

Table 6.23 Constant execution times achieved by the developed algorithm

	No of CPU operations	Time to execute at 80 MHz
Setup time	500	6.250 us
Key expansion time	1450	18.125 us
Round operation (LUT)	1650	20.625 us
Clear up time	200	2.500 us
Total time required for 128 bits	18650	305.625 us

Table 6.24 Block encryption execution time for the developed algorithm.

Test condition	Data size	Execution time	MB/s
AES-128-CyaSSL [147] (Cache enabled)	25 kB	0.04700	0.52 MB/s
AES-128-ASM-ECB (Cache disabled)	25 kB	0.04287	0.5694 MB/s
AES-128-ASM-CBC (Cache disabled)	25 kB	0.04650	0c.5250 MB/s

Although the implementation was designed to be resistant against timing attacks it is susceptible to power analysis attacks as it can be observed in Figure 6-41. This side channel attack could be attributed to rises in power during intensive SRAM access (LUTs).

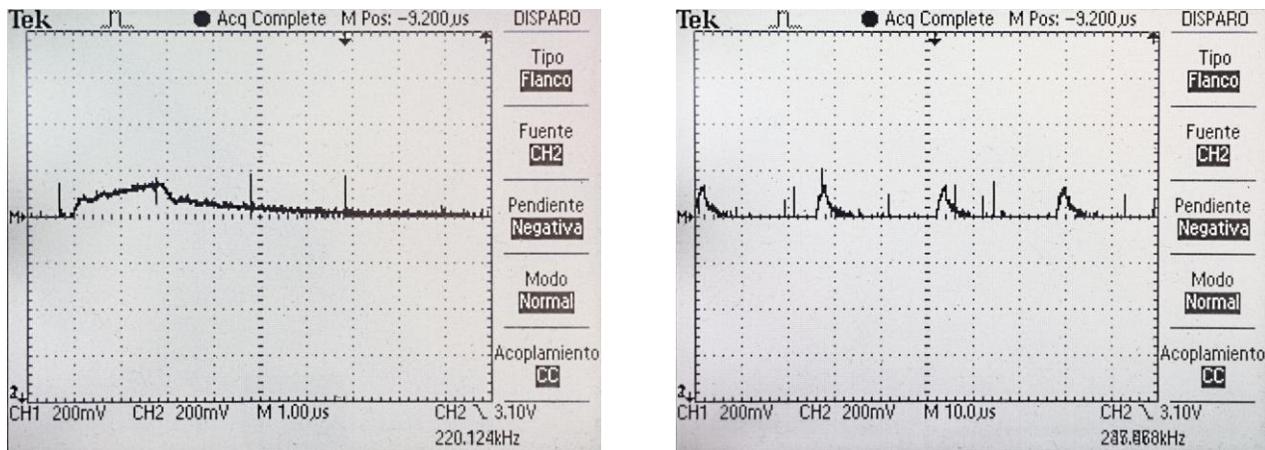


Figure 6-41 The proposed AES implementation features a constant execution time, but could be susceptible to power analysis attacks.

### 6.5.11 *Dynamic frequency measurement-Signal Validation*

In this section, a hardware/software validation is presented for the hardware modules that enable energy measurement during dynamic frequency conditions; these measurements were mostly done with the help of an oscilloscope that does not contain accuracy specifications and it is only used as a visualization tool.

#### 6.5.11.1 *Frequency measurement circuit*

Although system frequency can be considered constant under normal system operations due to tight frequency controls, its variability can affect the measurement capabilities of DFT based algorithms. For the Mexican market this variability is given by frequency tolerance band that is in

between 59.8 and 60.2 Hz during normal system operations. In Figure 6-42 a sample of frequency variability is given for a typical evening in a weekday.

To address these issues IEEE C37.118 proposes the use of a frequency sweep higher than 1 Hz over second to test dynamic response for P class units with a maximum measurement error of .01 Hz. In Figure 6-43 a proposed frequency sweep is proposed as the dynamic test for the frequency measurement circuit, this particular test pattern exhibits higher than required values to stress the Bessel filter response.

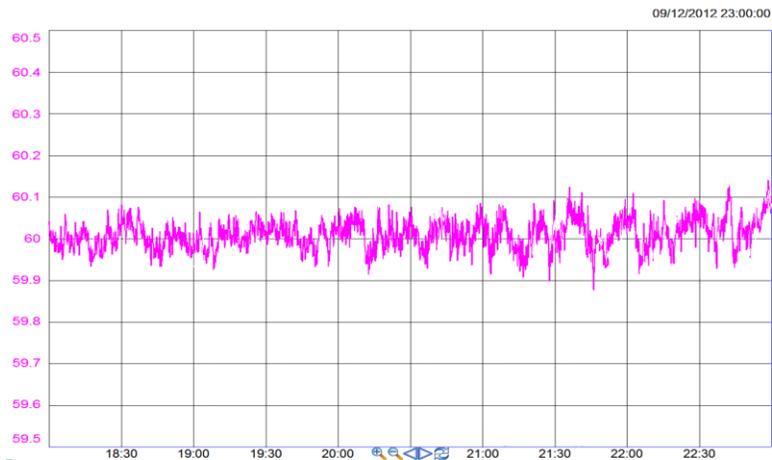


Figure 6-42 Typical frequency ranges and variations for the Mexican Electrical Interconnected Grid, obtained from

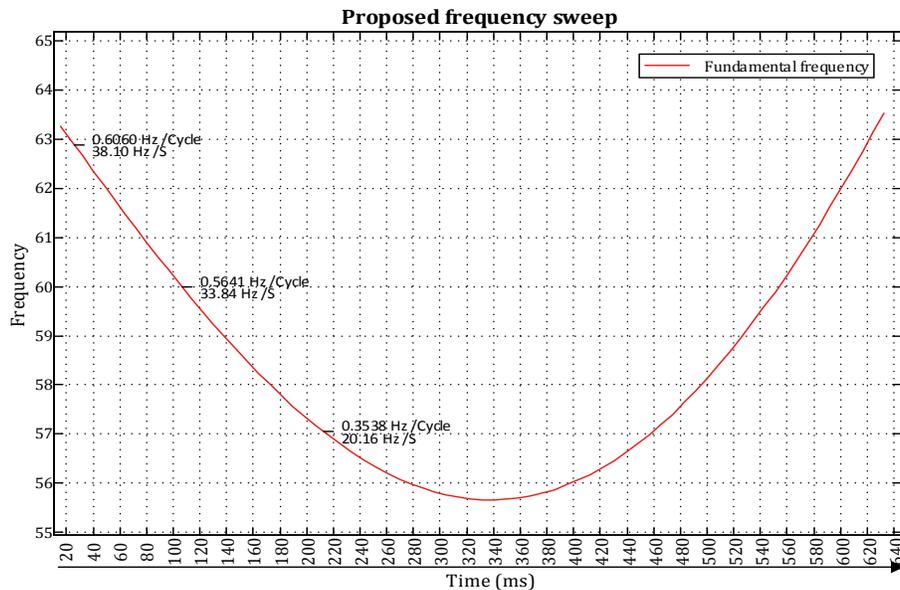


Figure 6-43 Frequency sweep applied to third order Bessel and Butterworth filter

In Figure 6-44 the proposed Bessel filter frequency response is presented, showing lesser than .01 Hz error over the entire range (57-63 Hz) under dynamic conditions, the results are compared with those provided by a typical Butterworth filter.

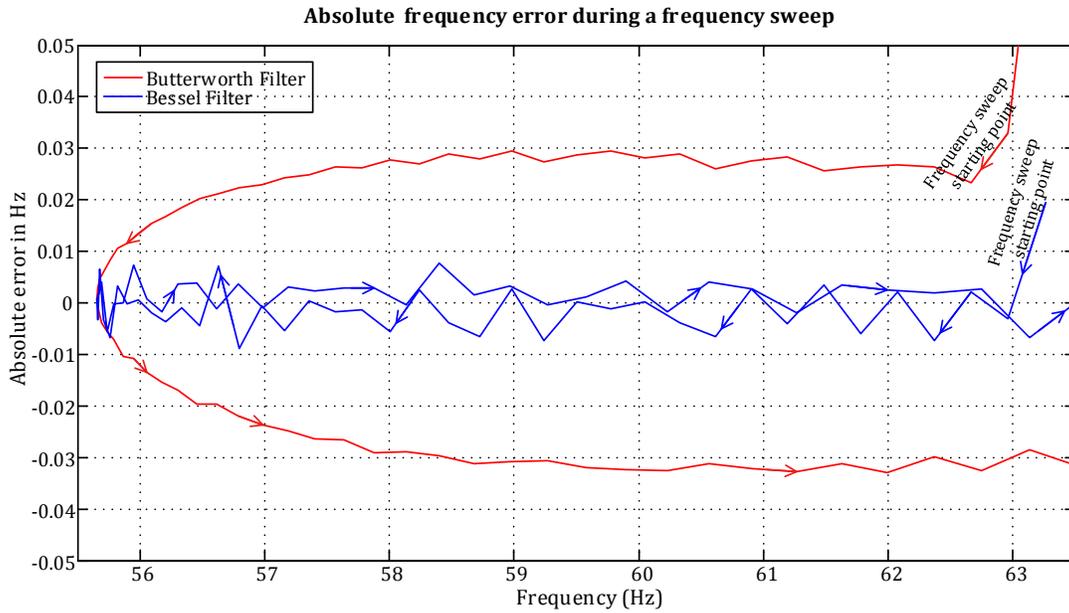


Figure 6-44 Frequency measurement errors recorded during the frequency sweep used to study the Bessel filter response.

Similarly, in Figure 6-45 the filter Bessel filter signal response is compared with a Butterworth filter where a higher damping factor can be observed for the Bessel filter, under the presence of high harmonics.

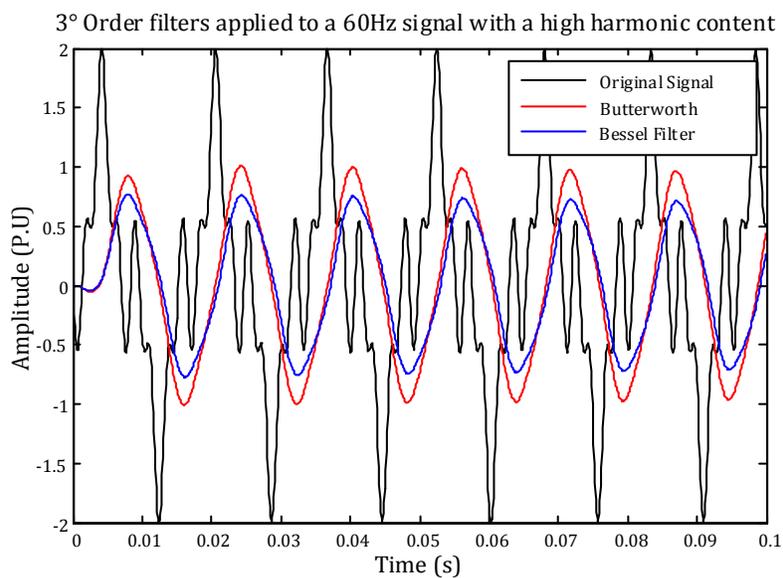


Figure 6-45 Signal response characteristics of the proposed Bessel filter.

Although filters are designed under ideal conditions it is helpful to know the effect of component variability. For the proposed components given in section 6.5.3.2, Figure 6-46 shows the group delay characteristics of the proposed filter. In this figure it can be observed that a constant phase delay remains, although the average time delay can be higher or lower of the idealized value.

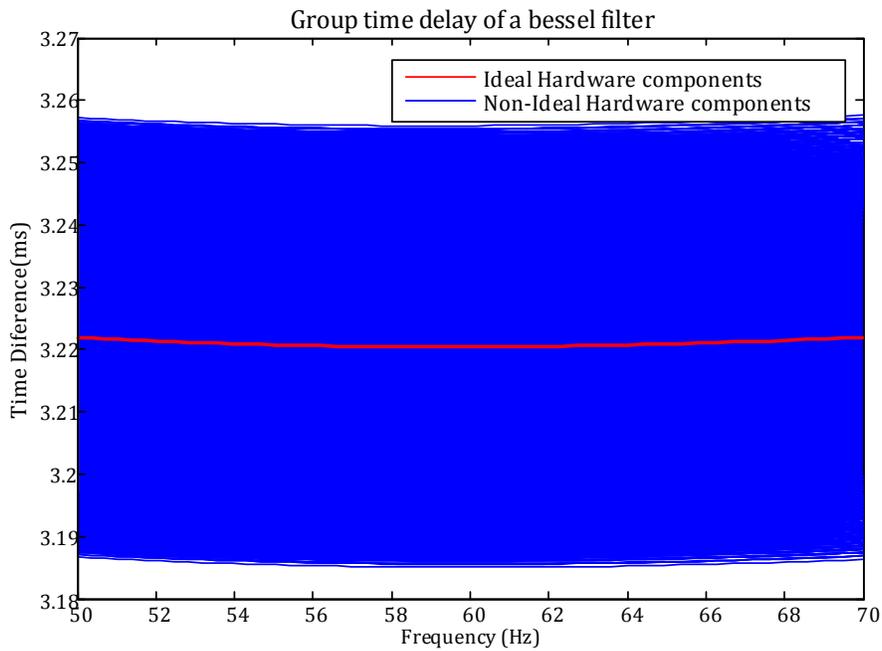
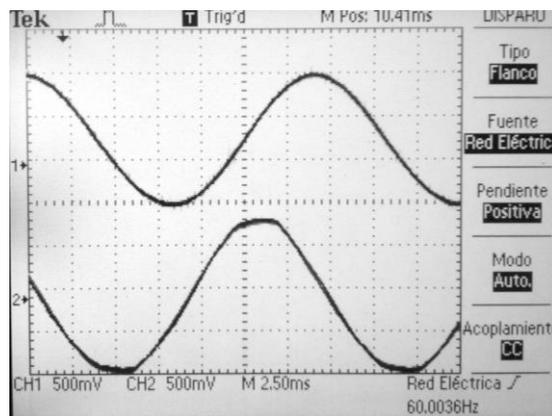
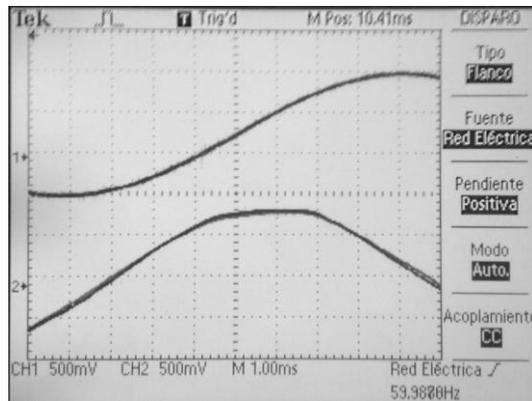


Figure 6-46 Group delay characteristics of the proposed filter under non-ideal component values.

In Figure 6-47 the actual filter response (channel 1) is shown on an oscilloscope screen given a voltage signal distorted by the presence of higher harmonics (channel 2). As it can be seen, a clearer and smoother signal is achieved after the filtering stage, but a 3.2 ms delay is appended.

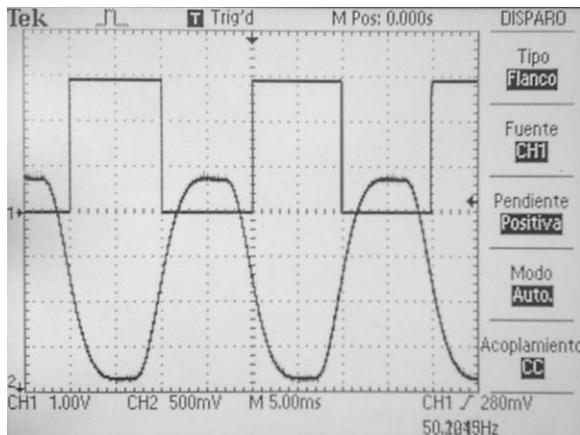


A) Input Frequency (bottom) vs Filtered Signal (S-K)

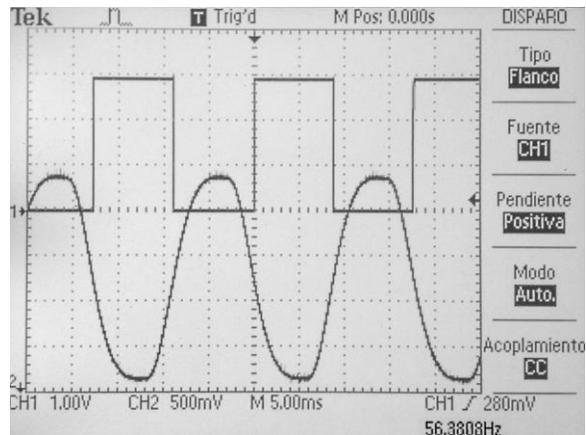


B) Input Frequency (bottom) vs Filtered Signal (S-K) (Zoomed)  
 Figure 6-47 Actual filter response characteristics of the proposed Bessel filter.

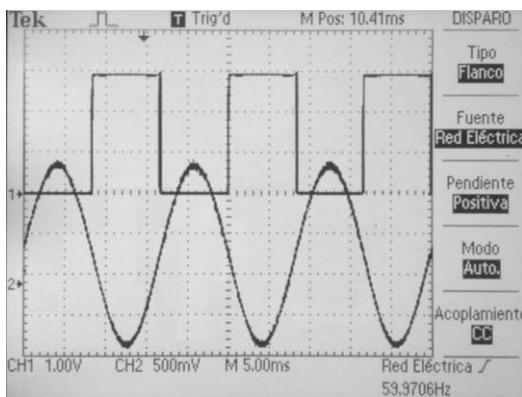
In Figure 6-48 the frequency module output vs the input signal is given for four input frequencies in the range of 50-63 Hz. Each of the images contains the input signal as generated by a sinusoidal frequency generator and the square wave signal outputted by the proposed frequency module.



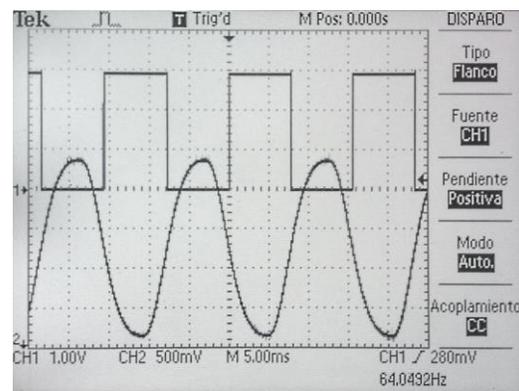
A) 50.2049 Hz



B) 56.3808 Hz



C) 59.9706 Hz



D) 64.0492 Hz

Figure 6-48 Sample frequency measurements as outputted by the frequency measurement hardware.

Finally, in Figure 6-49 three frequencies near the 60 Hz band are shown under the same oscilloscope screen by digitally adding sinusoidal signals oscillating at several frequencies.

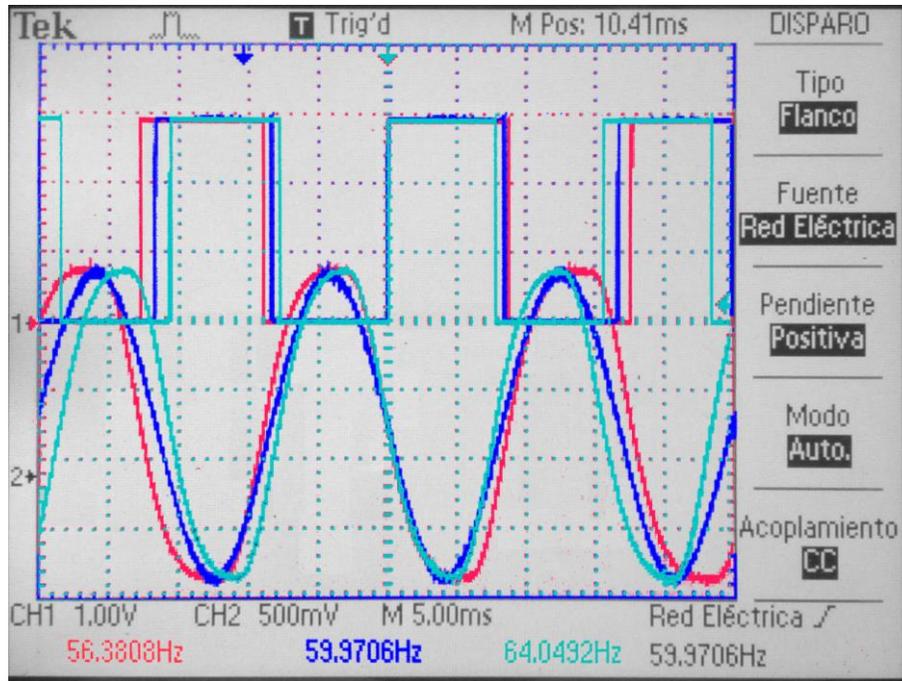


Figure 6-49 Simultaneous waveform captures at different frequencies, that shows correct filter performance.

#### 6.5.11.2 PLL validation

As mentioned in section 6.5.4 a digital PLL is used to dynamically adjust the ADC sampling frequency, in this section the PLL frequency control mechanisms are presented. The LMK03033C unit although digital in nature is susceptible to frequency deviations due to system noise, temperature effects and source clock deviations and thus a loopback mechanism must exist in order to provide a precise ADC clock reference. This loopback control is done by monitoring the ADC sample pulse (emitted at 1/256 the input clock frequency) and comparing it with a target PLL frequency dictated by the Frequency acquisition system (see Figure 6-50).

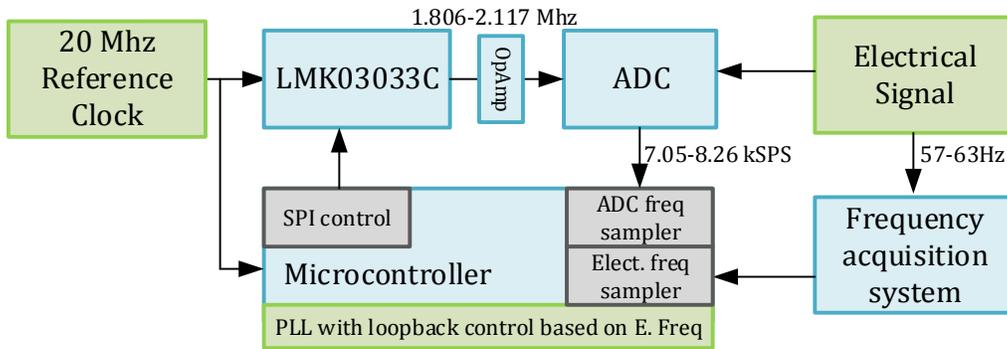


Figure 6-50 PLL loopback control.

The LMK03033C unit outputs a LDVS output that uses 2.4 volts to represent 1's and 1.2 volts to represent the 0's level, and thus this signal must be amplified to be used as the ADC input clock (which uses conventional 3.3 voltage levels). This is done thru the use of an Op-Amp circuit configured as a voltage comparator (see Figure 6-50 ) and DC filtering circuits.

In Figure 6-51 the PLL clock output is shown before applying the loopback control, as it can be seen an error exists between the output signal frequency and the expected frequency (controlled by the N\_Divisor). In Table 6.25 the relative error ( $R_e$ ) and absolute error ( $A_e$ ) of these frequency errors is determined, these type of frequency mismatches can cause significant ADC sample errors (equivalent to 0.1 Hz offset at 60 Hz), which will result in violations to the limits imposed by the IEEE C37.118.

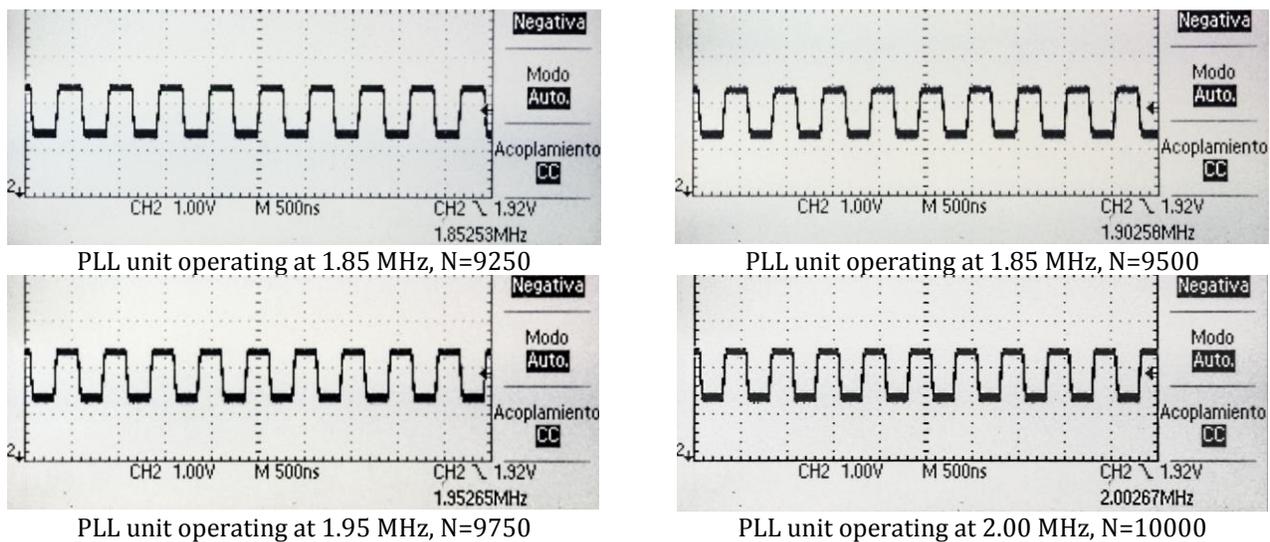


Figure 6-51 PLL output before applying frequency control

Table 6.25 PLL generated frequency characteristics.

	PLL Freq	ADC Freq						
Measured Freq.	1.85253	56.53473	1.90258	58.06213	1.95265	59.59015	2.00267	61.11664
Target Freq.	1.85000	56.45752	1.90000	57.98340	1.95000	59.50928	2.00000	61.03516
	$A_e$	-0.07721	$A_e$	-0.07874	$A_e$	-0.08087	$A_e$	-0.08148
	$R_e$	0.13676	$R_e$	0.13579	$R_e$	0.13590	$R_e$	0.13350

To address these type of errors a loopback control is implemented by monitoring the ADC\_CLOCK signal available on the ADS131E08 chip, the effects of this control can be observed in Figure 6-52. As it can be observed the clock deviations can be considered as zero.

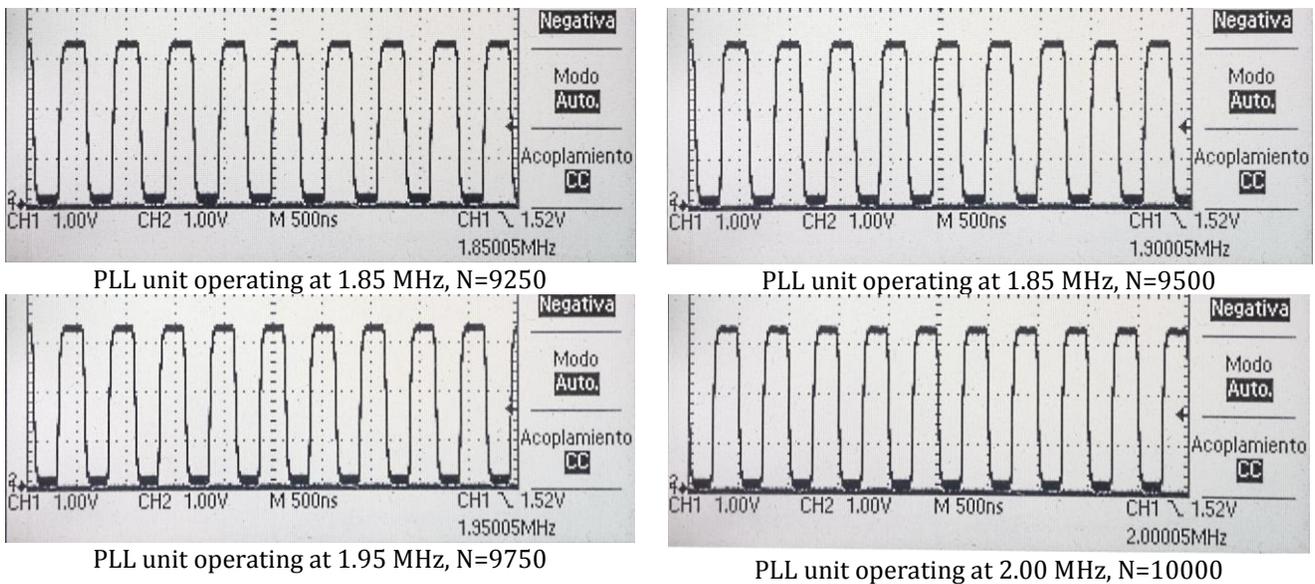


Figure 6-52 Amplified PLL output after applying frequency control

In Figure 6-53 the ADC\_CLOCK output (channel 1) vs the PLL output signal (channel 2) is shown, as per the ADC datasheet the ADC\_CLOCK signal is used to indicate that a conversion has been completed and must last for four PLL clock cycles. With the ADC\_CLOCK output frequency Table 6.26 was used to determine the actual sampling frequency, this value was compared with the intended sampling frequency and the results indicate that the frequency mismatch is at acceptable levels (0.001 Hz).

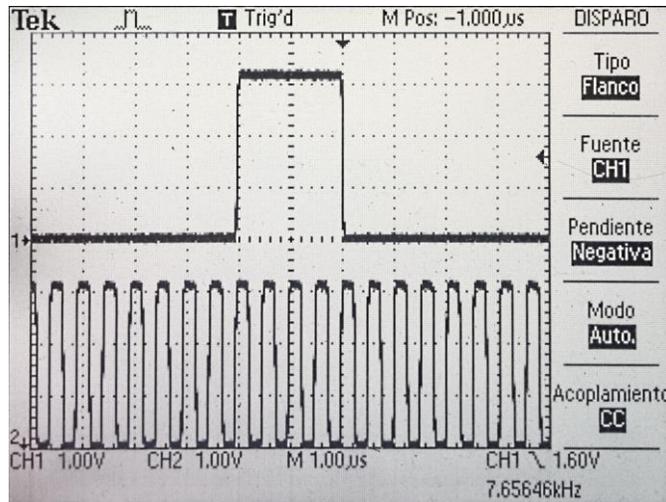


Figure 6-53 ADC\_CLOCK output compared with the PLL clock signal

Table 6.26 PLL generated frequency characteristics (with loopback control).

	PLL output (MHz)	ADC conversions (kSPS)	Actual sampling frequency (Hz)
Target Frequency	1.960000	7.65625	59.8144531
Achieved Frequency	1.960054	7.65646	59.8160938
		Abs Error (Hz)	0.00164063
		Relative Error (%)	0.00274286

### 6.5.12 *Microcontroller operating system.*

Based on the design requirements illustrated by Figure 6-1 two different firmware requirements were identified for each of the MPUs. First of all, the metering MPU requires to perform a set of repetitive tasks with hard time limits and thus it is well suited for firmware-based software that handles tasks based on a cooperative scheduling system. Whereas the communications MPU must handle dynamic communications that call for a multitasking system.

In microcontrollers, real-time OS are often used to introduce multitasking capabilities, often in the form of preemptive scheduler. A preemptive scheduler forcefully switches tasks based on system interruptions and task priorities thus creating a system capable of handling various events at the same time. In order to enable these task-switching mechanisms real-time OS's use additional memory resources to store program states and introduce switching times, an in depth discussion of embedded OS architectures can be found at [148].

Based on the requirements and supported architectures, FreeRTOS was chosen to provide multitasking capabilities to the communications microcontroller, particularly due to the fact that the selected TCP/IP stack supports FreeRTOS as its state-switching machine. FreeRTOS is an open source operating system designed to use a small footprint while at the same time provides advanced task switching mechanism like interlocking, semaphores, dynamic memory reservation and software based timers [149].

FreeRTOS is highly configurable and thus for this particular project it is hooked to the timer1 available in the PIC32MZ, with a tick of a 1/1000<sup>th</sup> of a second, which can be interrupted dynamically according to the communication needs, for example in radio based operations the driver cannot be interrupted until an acknowledge frame is received.

#### **6.5.13 *Smart meter communications platform.***

Smart metering units often provide data services that enable them to communicate with the utilities data servers. The communication used can range from proprietary protocols to standardized suites mostly based on the TCP/IP protocol. In this section the TCP/IP based communications supported by the proposed metering unit is presented. The unit enables to transmit HTTP traffic under two network interfaces (Ethernet and 802.15.4g) enabling to act as an end user device (client smart meter) and data collector (Ethernet to 802.15.4g bridge).

In Figure 6-54 a sample screen is shown for an Elster Alpha3 smart meter as accessed by CFE employees. The interface shows the energy registered by the data concentrators and stored on the utilities data servers, this data is often pulled from the Alpha3 meters via TCP/IP requests over proprietary Power Line Communications hardware.

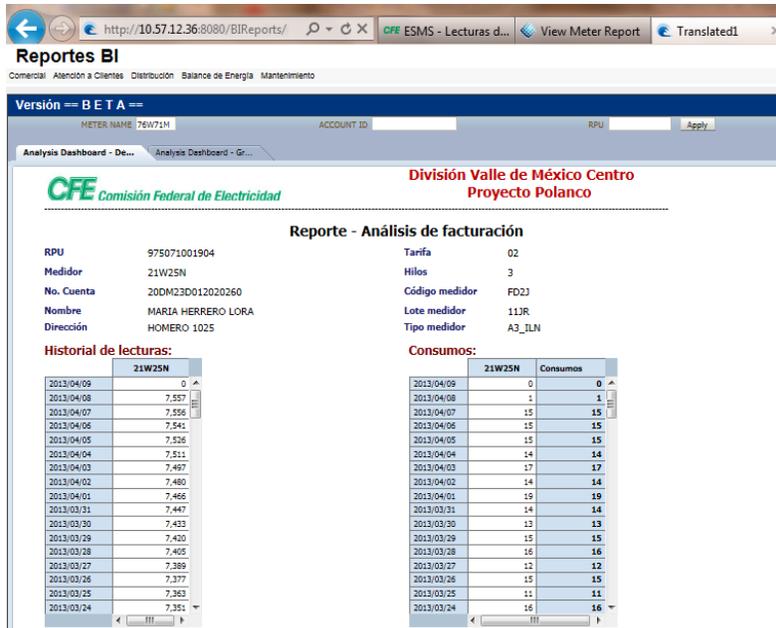


Figure 6-54 Web services provided by meters in CFE Polanco AMI project [150]

In Figure 6-55 the welcome screen for the developed meter is shown, in this case the HTTP response is served by the smart metering unit connected to a local area network. The transmitted content uses server-side includes to minimize traffic transferred thru the network.

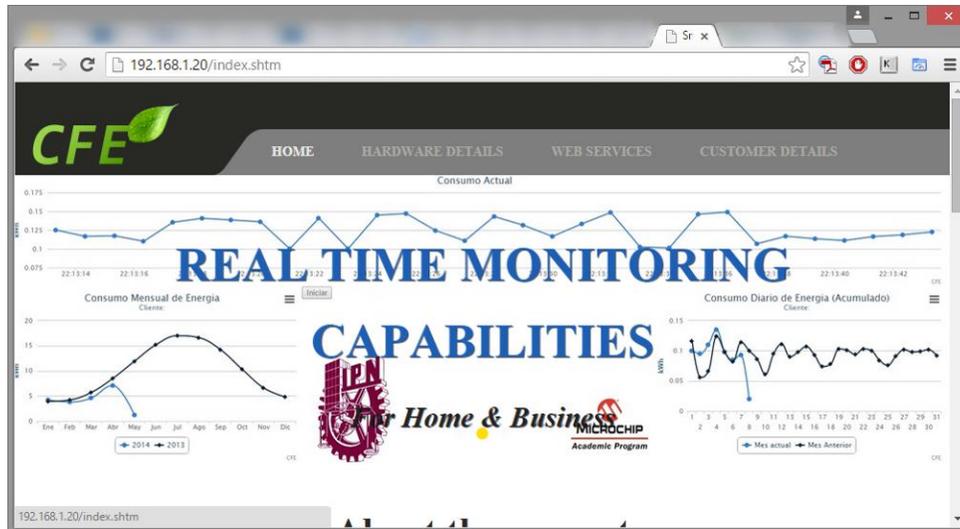


Figure 6-55 Web services provided by the developed meter.

By employing server-side includes (SSI), the amount of data being transferred across the smart metering network is reduced by assembling the data content at the end user browser (see Figure 6-56). This enables to embed metering data into resource intensive HTML content while reducing data traffic. The resource intensive content is dynamically loaded by using a technology known as

cross-site scripting, which can introduce security problems, and thus appropriate measures must be taken to avoid attacks [151].

Typical meter data request dataflow	Data payload requirements (typical values)	Data carrier
HTTP request to meter	256 bytes	Metering Infrastructure
Retrieve metering information with HTTP content and ajax	1 Kb	Metering Infrastructure
Retrieve HTTP content stored on remote servers	10 Kb	LAN/internet
Assemble HTML content + readings in browser	100+ Kb	LAN/internet

Figure 6-56 Data transfer requirements for the proposed metering architecture.

### 6.5.13.1 Embedded HTTP server tests.

The proposed meter uses a HTTP server that is based on the CycloneTCP implementation available at [152]. To enable dynamic HTTP testing the optical port is used to provide data logging capabilities to the smart meter, in Figure 6-57 a sample log is provided (captured by a serial adapter connected to a PC)

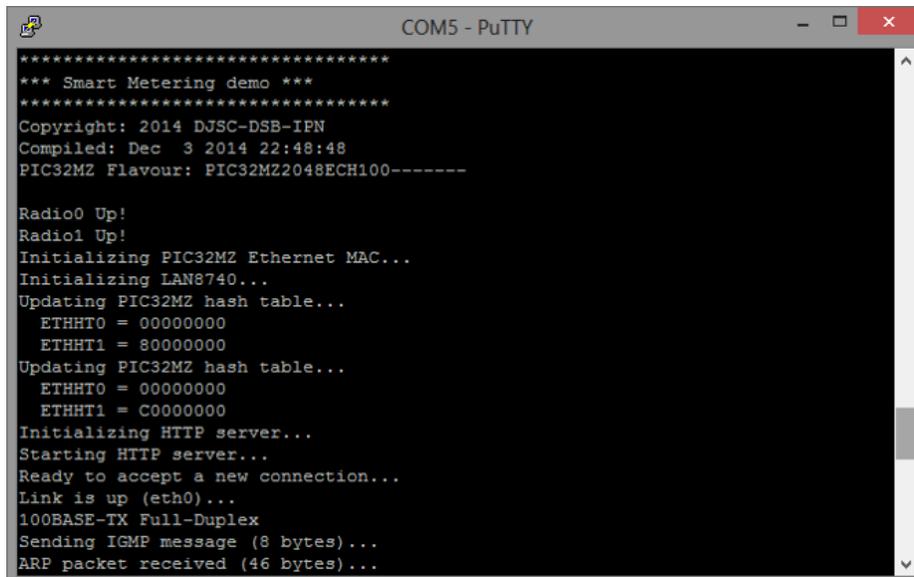


Figure 6-57 HTML Server log transmitted over the optical port installed on the developed metering unit.

HTTP servers are often tested with benchmark tools to estimate responsiveness under real life situations. These tests often measure the number of simultaneous connections supported, response speed, and overall availability. For the developed unit the test were carried by using an open source

tool called siege [153], which simulates a number of connected users to a HTTP servers under an ample set configurations. The results for these tests are reported by Figure 6-58.

```

C:\Windows\system32\CMD.exe
HTTP/1.1 200 0.09 secs: 755 bytes ==> GET /index.shtml
HTTP/1.1 200 0.09 secs: 755 bytes ==> GET /index.shtml
HTTP/1.1 200 0.09 secs: 755 bytes ==> GET /index.shtml

Lifting the server siege... done.

Transactions:      84 hits
Availability:     79.25 %
Elapsed time:     59.07 secs
Data transferred: 0.06 MB
Response time:   0.11 secs
Transaction rate: 1.42 trans/sec
Throughput:      0.00 MB/sec
Concurrency:     0.15
Successful transactions: 84
Failed transactions: 22
Longest transaction: 0.63
Shortest transaction: 0.09

FILE: siege.log
You can disable this annoying message by editing
the C:\siege-windows\etc\siegerc file; change
the directive 'show-logfile' to false.

C:\siege-windows>

```

Figure 6-58 HTML Server connection reliability results by using an open source benchmark tool.

The tests were done under the network illustrated by Figure 6-59, where the relative low availability can be attributed to the wireless links used to transfer the TCP/IP traffic. From Figure 6-58 it can also be observed that the amount of data being transfer is less than 1 kB per request, minimizing network congestion problems.

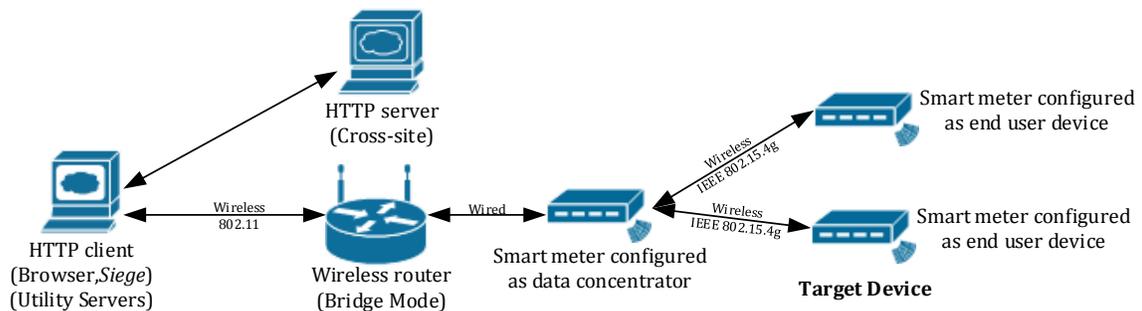


Figure 6-59 Network architecture used to determine connectivity to an end user device from a HTTP client.

In Figure 6-60 some sample HTTP content as served by the metering unit is shown, in this case only the dynamic HTML content is served by the unit, while additional graphical and style information resides in an alternate server.

## System information

```

DEVICE DETAILS
● METER HARDWARE: BOARD REV2
NETWORK INTERFACE
● MAC ADDRESS: 00-1E-C0-B0-0C-1E
● IPV4 ADDRESS: 192.168.1.20
● SUBNET MASK: 255.255.255.0
● DEFAULT GATEWAY: 192.168.1.254
● PRIMARY DNS: 8.8.8.8
● SECONDARY DNS: 8.8.4.4
CONNECTION
● REMOTE ADDRESS: 192.168.1.104
● REMOTE PORT: 60698
● SERVER ADDRESS: 192.168.1.20
● SERVER PORT: 80
DEBUGGING
● DOCUMENT URI: /PROPERTIES.SHTM
● QUERY STRING:
● LOAD COUNTER: 1 TIME
● GPS POS: 19.4978961,-99.1417412
    
```

A) Network details of the metering unit



HOME PAGE > PAGES > WEB SERVICES

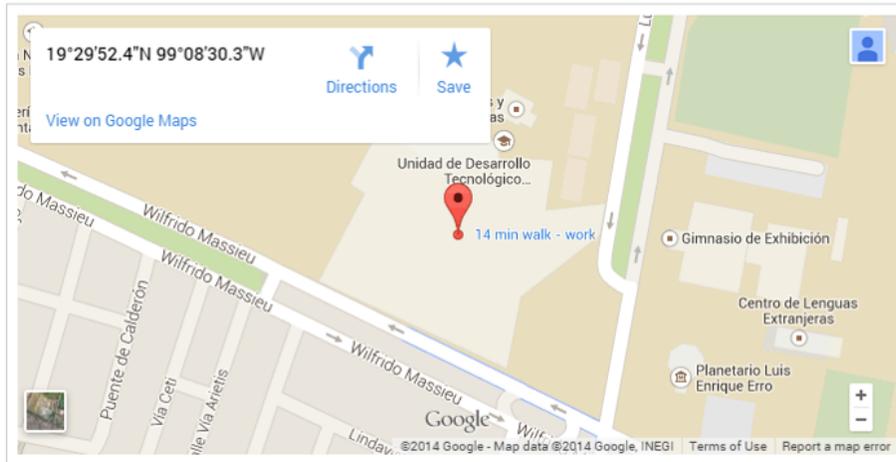
## Available services

- REMOTE DISCONNECTION
- REMOTE CONNECTION
- VIEW CONSUMPTION
- VIEW POWER QUALITY
- METRICS
- REAL TIME WAVEFORM

B) Web services provided by the metering unit

## Customer location

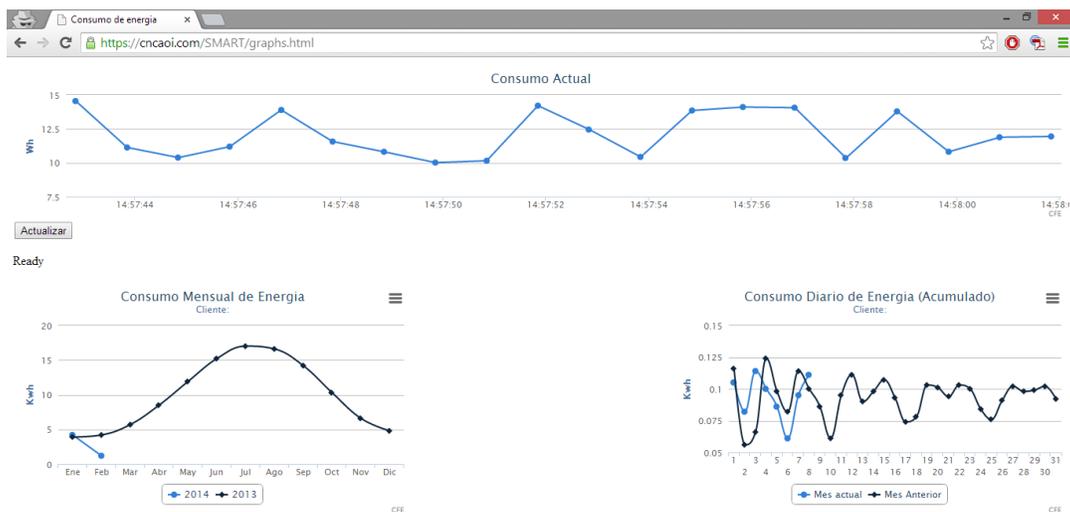
9.4978961,-99.1417412



### Customer Address (GPS)

Wilfrido Massieu 308, U. Prof.  
Adolfo López Mateos, Ciudad de  
México, D.F., Mexico

C) Customer location based on GPS sensor data



Consumer load seen from the utilities data server after been parsed from the web services data.  
Figure 6-60. Sample HTTP content provided by the developed unit.

### 6.5.13.2 Assembled Hardware Modules

In this section, the final version of the assembled hardware is presented. In Figure 6-61 the main smart meter PCB board can be observed, in this view the ADC module, GPS, radio and Ethernet port are shown mounted. The relevant electrical diagrams and PCB layouts can be found in Annex A.

In Figure 6-62 the LCD screen and optical port can be observed, while in Figure 6-63 the final metering unit as it looks on an ANSI 1S receptacle is shown.



Figure 6-61 Rear view of the metering unit configured as a data concentrator (with Ethernet module)

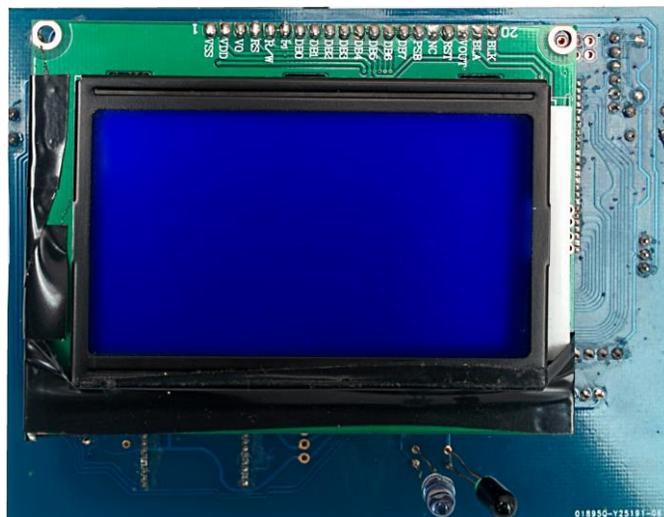


Figure 6-62 Front view of the designed metering unit with LCD and optical port mounted.



Figure 6-63 Front view of the designed metering unit mounted inside an ANSI 1S sized container.



## **CHAPTER 7**

### **7. CONCLUSIONS AND FUTURE WORK**

#### **7.1 Conclusions**

In this thesis a smart meter was designed, with several components being specifically designed and tested in depth. With respect to the measurements standards compliance, the results indicate that the accuracy levels are fulfilled for the steady state IEEE C37.118 domain and the complete IEEE 1459-2000 standard. Since no standard apparatuses were available for testing accuracy, the reported results were evaluated by comparing two hardware units developed by the author.

The compliance of the prototype with IEEE C37.118 standard was evaluated by analyzing the time-stamped differences between two units at the input and output processes. The input process was evaluated by analyzing the timing differences provided by the GPS receiver units, while the output processes were evaluated in terms of the TVE (as indicated by the standard).

The IEEE 1459-2000 standard compliance enables the meter to fulfill the revenue meter characteristics, by accounting each of the energy quantities that represent losses to the utility service. In this way, the utility can bill users in terms of multiple variables that can be monetized by using conversion variables that can be demand or time driven.

The high resolution Delta-Sigma ADC employed in conjunction with a hardware based frequency measurement mechanism enables to obtain correct phasor measurements under off nominal frequency conditions. These characteristics, plus appropriate PGA settings enable the unit to measure signals that vary in amplitude and frequency.

An extensive use of software-based optimization techniques were used to create a meter firmware that uses minimal amounts of computing resources, for example the DFT technique proves to be 80% faster than a traditional FFT solution implemented in C. These optimizations were designed to enable future-proof upgradeability by reserving computing resources for future in field-upgradability.

In this work, the concept of hardware optimized programs is featured as an optimizing solution, although it often requires deep knowledge of the hardware architecture, it enables to maximize the processor resources, and in certain cases to develop constant timing algorithms, which can be useful in cryptography.

A wireless transceiver unit was custom designed/driven to provide secure wireless communications, the current implementation enables to transmit real time consumption data to a central server via a HTTP protocol, this data could be further exploited to provide other services such as fault location and automated circuit reconfiguration capabilities.

During the thesis a secure communication meter was developed. It employs standardized security suites with additional custom blocks to provide a complete security solution. In this respect two key aspects must be mentioned. The implementation of a PUF function that enables to generate unique ID per meter that is used to secure the credential storage in the unit, and the implementation of a “cache-attack” resistant and “timing attack” resistant AES implementation.

Finally, an energy theft algorithm is proposed based on the characteristics of the previously developed meter. This algorithm uses the current data information to perform time-synched current balance operations. The simulation results indicate that this algorithm could be applicable to the smart meter architecture, but the actual field results are not available for reporting at this time.

The energy theft algorithm is capable of precisely identifying users that possess a malfunctioning/altered unit and determining the amount of energy being stolen. The algorithm was tested under numerous cases that involve a variant number of dishonest users; the overall algorithm response improves as the percent of dishonest users is decreased. Although this algorithm uses heuristic methods, it could be enhanced by using other methods like SVM's or artificial intelligence.

The result of this work is a smart meter unit that is targeted for commercial deployment. In the economic aspect the unit is designed to cost less than 200 USD when mass-produced by using fully

proprietary circuit layouts, and raw electronic components. The solution also uses proprietary software solutions or at least parts of code that are in the public domain and are licensed for commercial use.

## 7.2 Contributions

- An IEEE 1459-2000 compliant metering device was built. The unit complies with the ANSI C12 0.5 precision class. The physical design takes into consideration the size restrictions of ANSI C12 1S meter base. Most of the size reductions come from the use of surface mount components.
- A 24-bit Delta-Sigma fully differential ADC unit was employed as the analog front end, this enables the device to perform high quality measurements when compared against traditional SAR-based ADCs.
- An architecture-optimized DFT algorithm was developed, achieving speeds that surpass open source FFT implementations.
- The IEEE C37.188 “M class” PMU functionality was incorporated into the design by integrating a GPS unit that provides accurate time stamps and a PLL-driven dynamic frequency acquisition system.
- An IEEE 802.15.4g communication interface was developed, it employs a software driven RF module designed to be field upgradable, as per NIST 7628 requirements.
- A custom driver was built to serve as a bridge between the RF module and an open source TCP/IP stack that provides most of the Neighborhood Area Network capabilities.
- A dual-purpose unit that can serve as a smart meter or data concentrator was achieved by using an optional Ethernet port that serves as a wired to wireless data bridge, reducing utilities hardware\installation costs.
- Basic web service data fetching mechanisms were incorporated into the smart meter by incorporating a HTTP server into the firmware, this enables to deploy AMI technologies by retrieving data from individual devices located at the user premises.
- A software based AES subroutine was developed, it features execution times similar to commercial solutions but it is resistant to timing and cache based attacks by employing LUTs and assembly code programming.

- A Physical Unclonable function algorithm was developed, it uses the noise from the SRAM unit as its source of entropy and it is based on a bit counting mechanism that selects stable memory locations in order to generate a unique ID.
- TLS security was implemented thru the use of an open source library and integration with a custom AES function resistant to timing attacks.
- The MCU programming techniques include a mixture of C language and assembly that results in significant execution time reductions (AES and DFT).
- Two types of firmware architectures were used to deploy the microcontrollers used in the smart meter, the metering MCU uses a traditional loop based firmware, whereas the communication mCU employs a real time operating system (FreeRTOS)
- A real-time energy theft detection algorithm is proposed, it relies on the harmonic decomposition to perform time-synchronized Kirchoff-based current balancing between the energy meter located at the user premises and a central observer agent. The developed algorithm is able to pinpoint altered meters by analyzing consumption patterns across several days

### 7.3 Future work ideas

- To test the developed smart meter for accuracy levels with respect a calibrated standard to ensure the measurements precision.
- To test other forms of side-channel attacks to evaluate and improve the security modules.
- To publically challenge the information security community to find vulnerabilities on the proposed meter architecture, and mitigate those that are highly possible.
- To Include an RFID reader to enable prepay or post pay methods, often supported by meters deployed on areas where customers neglect to pay the energy bills, and complete wireless coverage is impossible to achieve.
- To create a proprietary smart meter case that is standards compliant and enables to deploy the meter commercially.
- To improve the circuit designs to fulfill commercial equipment tests designed to evaluate durability, accuracy, or other commercial aspects.
- To improve the energy theft detection algorithm by using automated detect thresholds, based on artificial neural networks, classifiers or SVM's techniques.

## REFERENCES

- [1] Department of Energy, "Smart Grid," [Online]. Available: <http://energy.gov/oe/technology-development/smart-grid>. [Accessed 2013 January 28].
- [2] J. Ekanayake, N. Jenkins, K. Liyanage, J. Wu and A. Yokoyama, *Smart Grid: Technology and Applications*, Wiley, 2012.
- [3] C. J. Bandim, J. Alves, A. V. J. Pinto, F. C. Souza, M. R. B. Loureiro, C. Magalhaes and F. Galvez-Durand, "Identification of energy theft and tampered meters using a central observer meter: a mathematical approach.," in *Transmission and Distribution Conference and Exposition 2003 IEEE PES*, 2003.
- [4] Diario Oficial De la Federacion, "Ley de la industria eléctrica," 11 08 2014. [Online]. Available: [http://www.dof.gob.mx/nota\\_detalle.php?codigo=5355986&fecha=11/08/2014](http://www.dof.gob.mx/nota_detalle.php?codigo=5355986&fecha=11/08/2014). [Accessed 03 11 2014].
- [5] K. Seger, *Revenue Protection: Combating Utility Theft & Fraud*, Tulsa, Oklahoma : PennWell Corp, 2005.
- [6] M. Córdoba, "Roba Pepsico luz durante ¡4 Años!," *El Reforma*, p. 1, 05 July 2012.
- [7] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis and R. Cepeda, "Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures," in *Smart Grid Communications (SmartGridComm), First IEEE International Conference on*, 2010.
- [8] X. Li, X. Liang, R. Lu, X. Shen, X. Lin and H. Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges," *Communications Magazine, IEEE*, vol. 50, no. 8, 2012.
- [9] K. Yilin Mo, K. Brancik, D. Dickinson, H. Lee, A. Perrig and B. Sinopoli, "Cyber-Physical Security of a Smart Grid Infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, 2012.
- [10] IEEE, «IEEE Standard for Local and metropolitan area networks—Low-Rate Wireless Personal Area Networks-Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks,» IEEE-IEEE LAN/MAN Standards Committee, New York, NY, 2012.
- [11] Shaw, William T.; Cyber security Consulting, "SCADA System Vulnerabilities to Cyber Attack," *Electric Energy T&D*, vol. 8, no. 6, pp. 62-68, 2004.
- [12] W. H. Kersting, *Distribution System Modeling and Analysis*, William H. Kersting: CRC Press, 2002.
- [13] EPRI, "Smart Grid Enterprise Architecture Interest Group," Electric Power Research Institute, Inc, 2013. [Online]. [Accessed 29 January 2014].
- [14] EPRI, "EPRI-Smart Grid Resource Center," Electric Power Research Institute, Inc, 2011. [Online]. Available: <http://smartgrid.epri.com/>. [Accessed 28 January 2014].
- [15] U.S. Department of Commerce, National Institute of Standards and Technology, "NIST Framework and Roadmap for Smart Grid Interoperability Standards," NIST, 2010.
- [16] Department of Energy, "What the smart frid means to America's future.," U.S. Department of Energy, 2008.
- [17] EPRI, "EPRI smart grid demonstration initiative-Fifth Year Update," Electric Power Research Institute, Palo Alto, California, 2013.
- [18] EIFER EDF R&D, "PREMIO The Smart Grid Demonstration Project supported by EDF," EUROPEAN INSTITUTE FOR ENERGY RESEARCH, 2011.
- [19] E. S. G. D. Project, "Électricité de France Smart Grid Host Site Evaluation Report After Six Months," Electric Power Research Institute, Inc, Palo Alto, California, 2011.
- [20] ENIAC Joint Undertaking, "Energy to Smart Grid," 2011. [Online]. Available: <http://www.e2sg-project.eu/>. [Accessed 30 January 2014].
- [21] ENIAC Joint Undertaking , "E2SG Project profile," [www.eniac.eu](http://www.eniac.eu), Brussels, Belgium, 2011.

- [22] P. J. Panneerselvam, Design of smart grid interfaces: Focusing of smart TVs, Karlskrona, Sweeden: School of Computing Blekinge Institute of Technology, 2013.
- [23] SENER, "Primer informe de labores 2012-2013," 12 November 2013. [Online]. Available: [http://sener.gob.mx/res/PE\\_y\\_DT/pub/Informe%20Labores%20SENER%202013.pdf](http://sener.gob.mx/res/PE_y_DT/pub/Informe%20Labores%20SENER%202013.pdf). [Accessed 5 November 2014].
- [24] A. López, "Ponen en queretaro 600 'antidiablitos'," *El Reforma*, p. 7, 1 June 2013.
- [25] Ochoa, Cesar; S & C, "Proyecto Red Inteligente basado en inteligencia distribuida en Cozumel," in *Seminario de Redes Inteligentes 2014*, Cuernavaca, Morelos, 2014.
- [26] Comision Federal de Electricidad, Division valle de mexico centro, «Beneficios del proyecto AMI polanco,» CFE, Morelia, Michoacan, 2013.
- [27] D. Nikolaev Nikovski, Z. Wang, A. Esenther, H. Sun, K. Sugiura, T. Muso and K. Tsuru, "Detection, Smart Meter Data Analysis for Power Theft," in *Machine Learning and Data Mining in Pattern Recognition.*, Springer Berlin Heidelberg, 2013, pp. 379-389.
- [28] CFE-División de Distribución Baja California, "Infraestructura avanzada de medicion," in *Reunión regional en Mexicali*, Mexicali, 2012.
- [29] Padilla, Polo; Eriquez, Harper; Raul, Cortés, "Desarrollo de un sistema de Medición de variables eléctricas para un sistema de medición de baja tension tipo industrial," IPN, Mexico DF, 2006.
- [30] B. Hernandez and R. Cortés, "Diseño e implentacion de un medidor sincrono normalizado con el estandar IEEE C37.118," IPN, Mexico City, 2009.
- [31] A. Valdiosera, "Diseño de medidor inteligente e implementación de sistema de comunicacion bidireccional," IPN, Mexico City, 2013.
- [32] K. S. Devendra and A. P.-L. S. Inc., "Tamper detection apparatus for electrical meters". US Patent CA 2638449 C, 8 January 2013.
- [33] M. V. Krishna Rao and S. H. Miller, "Revenue improvement from intelligent metering systems," in *Metering and Tariffs for Energy Supply, 1999. Ninth International Conference on*, IEEE, 1999.
- [34] Sanchez Rojo, Ruben; Protecsa Ing., "Proyecto piloto de redes inteligentes en México," in *Seminario de Redes Inteligentes*, Cuernava, Morelos, 2014.
- [35] SecureState; Spencer J. McIntyre, "Termineter- smart meter testing framework," [Online]. Available: [ps://github.com/securestate/termineter](https://github.com/securestate/termineter). [Accessed 1 July 2014].
- [36] Depuru, Soma; Shekara Sreenadh Reddy, Wang Lingfeng; Devabhaktuni, Vijay, «Support vector machine based data classification for detection of electricity theft.,» de *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*, 2011.
- [37] A. Statnikov, D. Hardin, I. Guyon and C. F. Aliferis, "A Gentle Introduction to Support Vector Machines in Biomedicine," in *AMIA*, 2009.
- [38] John Shawe-Taylor; Nello Cristianini, Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000.
- [39] S. Salinas, M. Li and P. Li, "Privacy-Preserving Energy Theft Detection in Smart Grids," in *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Seoul, Korea, 2012.
- [40] H. Khurana, M. Hadley, N. Lu and D. Frincke, "Smart-Grid Security Issues," *Security & Privacy, IEEE*, vol. 8, no. 1, 2010.
- [41] B. Bjelajac and R. Salazar, "Deploying Smart Meters for Compliance with Future Standards," European Telecommunications Standards Institute, 2011. [Online]. Available:

[http://docbox.etsi.org/workshop/2011/201104\\_SMARTGRIDS/04\\_INFORMATION\\_DATAMNGT/LANDYSa ndGYR\\_bjelajac\\_deployingSmartMetersforfutureStandards.pdf](http://docbox.etsi.org/workshop/2011/201104_SMARTGRIDS/04_INFORMATION_DATAMNGT/LANDYSa ndGYR_bjelajac_deployingSmartMetersforfutureStandards.pdf). [Accessed 6 March 2014].

- [42] Federal Energy Regulatory Commission, "Assessment of Demand Response and Advanced Metering," Department of Energy, Washington D.C, 2006.
- [43] Research, Pike, "Private Wireless Utility Field Area Networks," Pike Research, Washington D.C, 2012.
- [44] Research, Pike, "Power Line Communications for Smart Grids," Pike Research, Washington, D.C, 2012.
- [45] M. Integrated., "Smart meters solutions guide," Smart Grids Solutions Guide, 2014.
- [46] W. Kester, "ADC Architectures III: Sigma-Delta ADC Basics," Analog Device, MT-022, 2009.
- [47] M. E. v. Valkenburg, Analog Filter Design, CBS Publishing, 1982.
- [48] M. Balch, Complete Digital Design. A Comprehensive Guide To Digital Electronics and Components, McGraw-Hill Higher Education, 2003.
- [49] P. A. Mohan, VLSI Analog Filters, Active RC, OTA-C, and SC, London: Birkhauser-Springer, 2012.
- [50] J. Bishop, B. Trump and S. Mark, "FilterPro MFB and Sallen-Key Low-Pass Filter Design Program," Texas Instruments, 2001.
- [51] R. Sallen and E. L. Key, "A Practical Method of Designing RC Active Filters," *Circuit Theory-IRE TRANSACTIONS-CIRC*, pp. 74-86, 1957.
- [52] OKAWA Electric Design, "3rd order Sallen-Key Low-pass Filter Design Tool," OKAWA Electric Design, 2009. [Online]. Available: <http://sim.okawa-denshi.jp/en/Sallenkey3Lowkeisan.htm>. [Accessed 5 January 2014].
- [53] Christopher, Paul; Motorola, "Design a third-order Sallen-Key filters with one opamp," EDN Network, 2011.
- [54] M. Fortunato and T. Instruments, "Circuit Sensitivity, With Emphasis On Analog Filters," in *TI Developer Conference*, Dallas,Tx, 2007.
- [55] D. Irvine and D. Harle, "Calculation of the frequency spectrum," in *Data communications and networks: an engineering approach*, John Wiley and Sons, pp. 21-23.
- [56] I.-W. Commons, "Square Wave Fourier Series," 13 March 2008. [Online]. Available: [http://upload.wikimedia.org/wikipedia/commons/7/7e/Square\\_Wave\\_Fourier\\_Series.svg](http://upload.wikimedia.org/wikipedia/commons/7/7e/Square_Wave_Fourier_Series.svg). [Accessed 3 June 2014].
- [57] Faculty of Mathematical Studies, *Mathematics for part I engineering*, School of Mathematics-University of Southampton, 2001.
- [58] G. J. Wakileh, Power Systems Harmonics: Fundamentals, Analysis and Filter Design, Springer, 2001.
- [59] H. E. Mazin, E. E. Nino, W. Xu and J. Yong, "A Study on the Harmonic Contributions of Residential Loads.," *IEEE TRANSACTIONS ON POWER DELIVERY*, vol. 26, no. 3, pp. 1592-1600, 2011.
- [60] W. H. Hayt, Engineering Circuit Analysis, New York: McGraw Hill, 2012.
- [61] IEEE Power System Instrumentation and Measurements Committee, "IEEE Trial-Use Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions," IEEE-SA Standards Board, New York, NY 10016-5997, USA, 2000.
- [62] IEEE Standards Association, «IEEE Standard C37.118-2010, IEEE Standard for Synchrophasor Measurements for Power Systems,» IEEE Power & Energy Society, 2010.
- [63] H. F. Gaines, Cryptanalysis: A study of ciphers and their solution, Courier Dover Publications, 1956.
- [64] Y. L. Jonathan Katz, Introduction to Modern Cryptography: Principles and Protocols, Boca Raton, FL: CRC Press, 2008.

- [65] Bureau of Industry and Security, "Export Administration Regulation (EAR)," U.S. Department of Commerce, 2013. [Online]. Available: <http://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear>. [Accessed 2014 March 2].
- [66] A. J. Menezes, P. C. v. Oorschot and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.
- [67] C. Paar y J. Pelzl, «Understanding Cryptography-A Textbook for Students and Practitioners,» Springer, 2010.
- [68] Bohm, Christoph; Hofer, Maximilian; Pribyl, Wolfgang, "A Microcontroller SRAM-PUF," in *5th International Conference on Network and System Security (NSS)*, 2011.
- [69] Microsoft, "TechNet-Encryption," Microsoft TechNet, 2014. [Online]. Available: <http://technet.microsoft.com/en-us/library/cc962028.aspx>. [Accessed 2 March 2014].
- [70] National Institute of Standards and Technology-NIST,, "Announcing the Advanced Encryption Standard (AES)," Federal Information Processing Standards Publications, 26 November 2001. [Online]. Available: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>. [Accessed 3 March 2014].
- [71] W. Diffie and M. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, 1976.
- [72] D. Boneh, "Coursera-Crypto I-Using-block ciphers," 06 February 2012. [Online]. Available: <http://spark-university.s3.amazonaws.com/stanford-crypto/slides/04-using-block-v2-annotated.pdf>. [Accessed 23 September 2013].
- [73] R. Spreitzer y T. Plos, «Cache access pattern attack on disaligned AES T-Tables,» de *COSADE*, Paris, 2013.
- [74] D. A. Osvik, A. Shamir and E. Tromer, "Cache Attacks and Countermeasures: the Case of AES," in *Topics in Cryptology—CT-RSA*, Springer Berlin Heidelberg, 2006, pp. 1-20.
- [75] S. Law, "Frequently Asked Questions (and answers) about reverse engineering," ChillingEffects.org, [Online]. Available: <https://chillingeffects.org/reverse/faq.cgi>. [Accessed 23 June 2014].
- [76] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," *Security & Privacy, IEEE*, vol. 9, no. 3, 2011.
- [77] Searle, Justin-Inguardians Inc., «Attacking and defending the grid: Pulling back the curtains to reveal the front battle lines of smart grid security,» de *DEFCON 19*, Las Vegas, 2010.
- [78] A. Zonenberg, "Silicon Exposed-Microchip PIC32MZ process vs PIC32MX," 24 March 2014. [Online]. Available: <http://siliconexposed.blogspot.mx/2014/03/microchip-pic32mz-process-comparison-to.html>. [Accessed 14 September 2014].
- [79] Committee on National Security Systems, "National Information Assurance Glossary (CNSS Instruction No. 4009)," US National Security Systems, 2010.
- [80] U.S. Department of Commerce, National Institute of Standards and Technology, "Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cyber Security Strategy, Architecture, and High-Level Requirements," NIST, 2010.
- [81] J. Froehlich, E. Larson, S. Gupta, G. Cohn, M. S. Reynolds and S. N. Patel, "Disaggregated end-use energy sensing for the smart grid," *IEEE Pervasive Computing*, vol. 10, no. 1, 2011.
- [82] U.S. Department of Commerce, National Institute of Standards and Technology, "Guidelines for Smart Grid Cyber Security: Vol. 2, Privacy and the Smart Grid," NIST, 2010.
- [83] M. Newborough and P. Augood, "Demand-side management opportunities for the UK domestic sector," *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 146, no. 3, 1999.
- [84] N. Ferguson, B. Schneier and T. Kohno, Cryptography Engineering: Design Principles and Practical Applications, Wiley, 2010.

- [85] J. Bringer, H. Chabanne and T. Icart, "On Physical Obscuration of Cryptographic Algorithms," in *INDOCRYPT 2009, 10th International Conference on Cryptology in India*, New Delhi, 2009.
- [86] Guajardo, Jorge, Boris Škorić, Pim Tuyls, Sandeep S. Kumar, Thijs Bel, Antoon HM Blom, and Geert-Jan Schrijen. , «Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions,» *Information Systems Frontiers*, vol. 11, nº 1, pp. 19-41, 2009.
- [87] U. o. M. School of Mathematics and P. Symonds, "MATH32031: Coding Theory -Hamming distance," 24 September 2007. [Online]. Available: <http://www.maths.manchester.ac.uk/~pas/code/notes/part2.pdf>. [Accessed 12 April 2014].
- [88] Biebighauser, Dan -University of Minnesota - Twin Cities, "Testing Random Number Generators," 2000. [Online]. Available: <http://www.math.umn.edu/~garrett/students/reu/pRNGs.pdf>. [Accessed 6 May 2014].
- [89] R. Jain, "Testing Random-Number Generators," 2008. [Online]. Available: [http://www.cse.wustl.edu/~jain/cse567-08/ftp/k\\_27trg.pdf](http://www.cse.wustl.edu/~jain/cse567-08/ftp/k_27trg.pdf). [Accessed 7 June 2014].
- [90] RANDOM.ORG, "RANDOM.ORG -Analysis," [Online]. Available: <http://www.random.org/analysis/>. [Accessed 6 June 2014].
- [91] Fall, Kevin R.; Stevens, W. Richard, *TCP/IP Illustrated, Volume 1: The Protocols*, Ann Arbor, Michigan: Addison-Wesley Professional, 2011.
- [92] H. D. Young and R. A. Freedman, "Chapter 32-ELECTROMAGNETIC WAVES," in *Sears and Zemansky's UNIVERSITY PHYSICS, WITH MODERN PHYSICS*, Boston, Addison-Weasley, 2010, pp. 1051-1074.
- [93] Du, Ke-lin; Swamy, M. N. S., *Wireless Communication Systems: From RF Subsystems to 4G Enabling Technologies*, Cambridge: CAMBRIDGE UNIVERSITY PRESS, 2010.
- [94] D. D. Coleman y D. A. Westcott, *Certified Wireless Network Administrator -Study Guide*, Indianapolis, Indiana: Wiley Publishing, Inc, 2006.
- [95] D. H. Johnson, "Fundamentals of Electrical Engineering-Basics of digital Comunciations," 18 August 2012. [Online]. Available: [https://d396qusza40orc.cloudfront.net/eefun/lecture\\_slides/Channels\\_I.pdf](https://d396qusza40orc.cloudfront.net/eefun/lecture_slides/Channels_I.pdf). [Accessed 6 August 2013].
- [96] K. S. Panchal, *IMPLEMENTING PHYSICAL LAYER (PHY) OF IEEE 802.15.4G STANDARD WITH DIRECT SEQUENCE SPREAD SPECTRUM (DSSS) USING OFFSET QUADRATURE PHASE SHIFT KEYING (O-QPSK)*, San Diego State University, 2012.
- [97] K.-H. Chang, B. Mason y E. Solutions, «The IEEE 802.15.4g Standard for Smart Metering Utility Networks,» de *IEEE SmartGridComm 2012 Symposium*, Tainan City, Taiwan, 2012.
- [98] J. Ltd, «Co-existence of IEEE 802.15.4 at 2.4 GHz Application Note,» Jennic Ltd an NXP Subsidiary, Sheffield, 2008.
- [99] D. H. Johnson, "Fundamentals of Electrical Engineering-Digital Comunication Receivers," 20 August 2012. [Online]. Available: [https://d396qusza40orc.cloudfront.net/eefun/lecture\\_slides/Digital\\_Comm\\_II.pdf](https://d396qusza40orc.cloudfront.net/eefun/lecture_slides/Digital_Comm_II.pdf). [Accessed 3 August 2014].
- [100] Microchip Technology, "MRF24XA- Low-Power, 2.4 GHz ISM-Band IEEE 802.15.4™ RF Transceiver with Extended Proprietary Features," Microchip-Datasheet, Chandler, Arizona, 2013.
- [101] IEEE-LAN/MAN Standards Committee-IEEE Computer Society, «IEEE Std 802.15.4™-2011-IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),» IEEE Standards Association, New York, NY, 2011.
- [102] Gutierrez, Jose A. , «IEEE Std. 802.15.4- Enabling Pervasive Wireless Sensor Networks,» de *Eaton-Innovation Center*, 2005.

- [103] Microchip Technology, "MRF24XA PICtail™/PICtail Plus Daughter Board User's Guide," Microchip-Documentation, Chandler, Arizona, 2013.
- [104] ANSI-StandardsPortal, "Standards Developing Organizations," American National Standards Institute, 2013. [Online]. Available: [http://www.standardsportal.org/usa\\_en/resources/sdo.aspx](http://www.standardsportal.org/usa_en/resources/sdo.aspx). [Accessed 25 February 2014].
- [105] B. o. E. Geology and I. University of Texas at Austin, "Guide to Electric Power In Mexico," Center for Energy Economics, Bureau of Economic Geology, the University of Texas at Austin, Austin, Tx, 2006.
- [106] CFE, "Estadísticas, Clientes," 08 January 14. [Online]. Available: [http://www.cfe.gob.mx/ConoceCFE/1\\_AcercadeCFE/Estadisticas/Paginas/Clientes.aspx](http://www.cfe.gob.mx/ConoceCFE/1_AcercadeCFE/Estadisticas/Paginas/Clientes.aspx). [Accessed 23 January 2014].
- [107] Solicitud de Informacion publica, "Sistema Infomex Gobierno Federal," 1816400188711, 22 November 2011. [Online]. Available: <https://www.infomex.org.mx/gobiernofederal/moduloPublico/moduloPublico.action>.
- [108] E. O. Reza, "Conferencia magistral de clausura- "Retos de la CFE dentro del marco de la reforma electrica", " in *RVP-AI 2014*, Acapulco, Gro, 2014.
- [109] Solicitud de Informacion publica, "Sistema Infomex Gobierno Federal," 1816400050713, 29 April 2013. [Online]. Available: <https://www.infomex.org.mx/gobiernofederal/moduloPublico/moduloPublico.action>.
- [110] Solicitud de Informacion publica, "Sistema Infomex Gobierno Federal," 1816400010712, 13 February 2012. [Online]. Available: <https://www.infomex.org.mx/gobiernofederal/moduloPublico/moduloPublico.action>.
- [111] TydentBrooks Security Products Group, "Enviro NL-1 Padlock Seal," [Online]. Available: <http://www.tydenbrooks.com/Products/Indicative-Seals/Wire-Seals/Enviro-NL-1-Padlock-Seal.aspx>. [Accessed 2 September 2014].
- [112] TydentBrooks Security Products Group, "Cable Seal Selector," 21 February 2014. [Online]. Available: <http://www.tydenbrooks.com/Products/Cable-Seals.aspx>; <https://www.youtube.com/watch?v=QiXHJbumK4g>. [Accessed 3 September 2014].
- [113] G. Sreenivasan, Power Theft: Educates and sensitises people about the menace of power theft, New Delhi: PHI Learning Private Limited, 2011.
- [114] American National Standards Institute, Inc., "ANSI C12.1-2008 American National Standard for Electric Meters, Code for Electricity Metering," National Electrical Manufacturers Association, Rosslyn, VA, USA, 2008.
- [115] International Electrotechnical Commission, "Electricity metering equipment (AC)-General requirements, tests and test conditions," INTERNATIONAL STANDARD, Geneva, Switzerland, 2003.
- [116] Solicitud de Informacion publica, "Sistema INFOMEX Gobierno Federal," 1816400116214, 17 June 2014. [Online]. Available: <https://www.infomex.org.mx/gobiernofederal/moduloPublico/moduloPublico.action>.
- [117] D. Hall, "An rtl-sdr receiver for Itron ERT compatible smart meters operating in the 900MHz ISM band.," Github, February 2014. [Online]. Available: <https://github.com/bemasher/rtlamr>. [Accessed 16 June 2014].
- [118] krebsonsecurity.com, "FBI: Smart Meter Hacks Likely to Spread," 12 April 2012. [Online]. Available: <http://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/>. [Accessed 29 January 2014].
- [119] IOActive, «Secuting the Smart Grid-"To act without delay",» de *InfoSecurity Europe 2010*, London, 2010.
- [120] S. D., N. W. S. and A.-C. LIEW, "Neural-network-based signature recognition for harmonic source identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398-405, 2006.

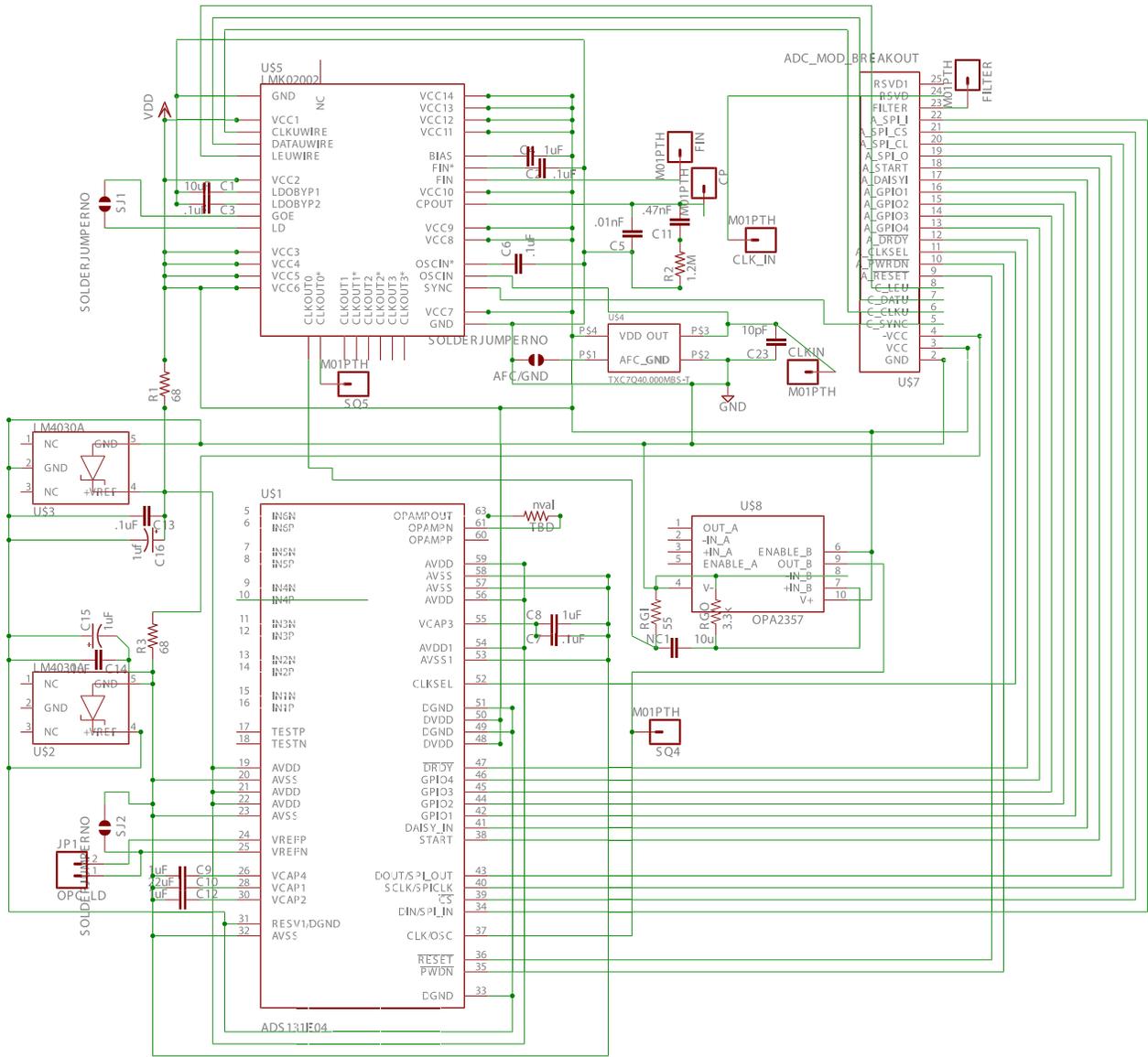
- [121] G. Cornuejols and M. Trick, "Quantitative Methods for the Management Sciences-Class Notes," 24 August 1998. [Online]. Available: <http://mat.gsia.cmu.edu/classes/QUANT/NOTES/chap7.pdf>. [Accessed 8 September 2014].
- [122] CFE-CONACYT-IPN, *Registros de consumo del alimentador OAP4030 en el año 2013, del Proyecto CFE-2006-C05-48545*, Oaxaca, Mexico, 2013.
- [123] A. Priyadharshini, N. Devarajan, A. Umasarany y R. Anitt, «Survey of Harmonics in Non Linear Loads,» *International Journal of Recent Technology and*, vol. 1, n° 1, 2012.
- [124] NEMA Smart Grid Standards Publication, "Requirements for Smart Meter Upgradeability," National Electrical Manufacturers Association, Rosslyn, Virginia, 2009.
- [125] J. Horgan, G. Keogh and J. Waldron, "Analysis of Factors Afecting Assembly Language Programming Ability.," 1999.
- [126] J. Hennessy, N. Jouppi, S. Przybylski, C. Rowen, T. Gross, F. Baskett and J. Gill, "MIPS: A Microprocessor Architecture," in *MICRO 15 Proceedings of the 15th annual workshop on Microprogramming*, Piscataway, NJ, 1982.
- [127] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design*, Morgan Kaufmann Publishers, 2009.
- [128] Intel, "What features are unique to Intel® Core™2 Duo mobile processors?," 22 June 2006. [Online]. Available: <http://www.intel.com/support/processors/mobile/core2duo/sb/CS-023242.htm>. [Accessed 13 May 2014].
- [129] Microchip, "PIC32MZ series manual, section 50. "CPU for Devices with microAptiv™ Core", " Datasheet, Chandler, 2014.
- [130] R. Levy, «Memory issues in power-aware design of embedded systems: An overview,» de *Second International Workshop on Compiler and Architecture Support for Embedded Systems*, 1999.
- [131] Microchip Technology, "PIC32MZ Embedded Connectivity (EC) Family Datasheet," Rev C-Preliminary, Chandler, Arizona, 2014.
- [132] Maxim IC, "Understanding SAR ADCs: Their Architecture and Comparison with Other ADCs," Dallas Semiconductor-Maxim IC, 2001.
- [133] E. Worthman, "It's All IP In An SoC," *Semiconductor Engineering*, 5 June 2014. [Online]. Available: <http://semiengineering.com/its-all-ip-in-an-soc/>. [Accessed 22 June 2014].
- [134] Texas Instruments, "Analog Front-End for Power Monitoring, Control, and Protection-ADS13E0x," Datasheet, 2012.
- [135] Texas Instruments, «How to Select the Right Voltage Reference,» Application Note, 2012.
- [136] Texas Instruments, «LM4030 SOT-23 Ultra-High Precision Shunt Voltage Reference,» Datasheet, 2008.
- [137] Texas Instruments, "Clock Design Tool - Loop Filter & Device Configuration + Simulation," 02 February 2012. [Online]. Available: <http://www.ti.com/tool/clockdesigntool>. [Accessed 6 November 2014].
- [138] J. G. Proakis and D. K. Manolakis, "Chapter 8: Efficient Computaiton Of The Dft: Fast Fourier Transform Algorithms," in *Digital Signal Processing*, Pearson- Higher Education, 2007.
- [139] S. Schinzel, «An Efficient Mitigation Method for Timing Side on the Web,» de *Second International Workshop on Constructive Side-Channel Analysis and Secure Design*, 2011.
- [140] Colaborative Work, "Literateprograms.org," 17 November 2008. [Online]. Available: [http://en.literateprograms.org/Cooley-Tukey\\_FFT\\_algorithm\\_\(C\)](http://en.literateprograms.org/Cooley-Tukey_FFT_algorithm_(C)). [Accessed 2013 May 2013].
- [141] GPS NAVSTAR, "Global positioning system standard positioning service signal specification," U.S. Department of Homeland Security, 1995.

- [142] Federal Aviation Agency/William J. Hughes Technical Center, "Wide-area augmentation system performance analysis report," Federal Aviation Administration, Atlantic City, 2008.
- [143] National Coordination Office for Space-Based Positioning, Navigation, and Timing, "GPS Accuracy-Official U.S. Government information about the Global Positioning System (GPS) and related topics," 18 September 2013. [Online]. Available: <http://www.gps.gov/systems/gps/performance/accuracy/>. [Accessed 30 September 2013].
- [144] U.S Forest Service, "Head-To-Head Comparison of Four SiRF-Based GPS Receivers," 8 March 2009. [Online]. Available: <http://www.fs.fed.us/database/gps/documents/SiRFComp.pdf>. [Accessed 15 August 2013].
- [145] M. Predko, "Programming and Customizing the PIC Microcontroller," McGraw Hill/Tab Electronics, 2007.
- [146] Hunter, Melissa; Freescale Semiconductor, "Understanding LCD Memory and Bus Bandwidth Requirements," Freescale Semiconductor, 2008.
- [147] wolfSSL Inc, "CyaSSL Microchip PIC32 Support," 2014. [Online]. Available: <http://www.yassl.com/yaSSL/cyassl-pic32.html>. [Accessed 5 September 2014].
- [148] D. E. Simon, An Embedded Software Primer, Addison-Wesley Professional, 1998.
- [149] R. Barry, FreeRTOS Tutorial Book, Microchip PIC32 Edition, FreeRTOS Tutorial Books.
- [150] Solicitud de Informacion publica, "Sistema Infomex Gobierno Federal- Solicitud 1816400076313," 1 April 2013. [Online]. Available: <https://www.infomex.org.mx/gobiernofederal/moduloPublico/moduloPublico.action>. [Accessed 4 November 2014].
- [151] The Open Web Application Security Project, "XSS (Cross Site Scripting) Prevention Cheat Sheet," 12 04 2014. [Online]. Available: [https://www.owasp.org/index.php/XSS\\_\(Cross\\_Site\\_Scripting\)\\_Prevention\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet). [Accessed 20 10 2014].
- [152] CycloneTCP, "Oryx-Embedded-Download Source code," 19 March 2014. [Online]. Available: <http://www.oryx-embedded.com/download.html>. [Accessed 2014 25 March].
- [153] J. Dog, "Siege 3.0.3," 23 July 2013. [Online]. Available: <https://github.com/tail/siege>. [Accessed 13 October 2014].
- [154] Joint Committee for Guides in Metrology, «International vocabulary of metrology — Basic and general concepts and associated terms,» Bureau International des Poids et Mesures, Paris, France, 2008.
- [155] W. Kester, "Fundamentals of Sampled Data Systems," in *The Data Conversion Handbook*, Analog Devices, 2004, pp. 39-52.
- [156] Maxim Integrated, "Clock Jitter and phase Noise Conversion," Application Note 3359, 2004.
- [157] N. Semiconductor, "Clock Conditioner Owner's Manual," Application note, 2006.
- [158] P. Savicky, "A strong nonrandom pattern in Matlab default random number generator.," Manuscript available from <http://www2.cs.cas.cz/~savicky/papers/rand2006.pdf>, 2006.
- [159] J. MOSER, "Moserware, jeff moser's software development adventures.," 22 September 2009. [Online]. Available: <http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html>. [Accessed 1 June 2013].
- [160] G. Bertoni, L. Breveglieri, P. Fragneto, M. Macchetti and S. Marchesin, "Efficient software implementation of AES on 32-bit platforms," in *Cryptographic Hardware and Embedded Systems-CHES*, Springer Berlin Heidelberg, 2002, pp. 159-171.

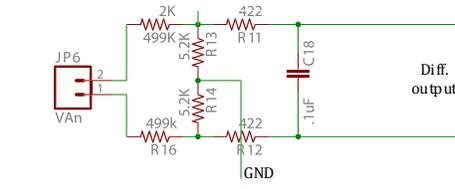
- [161] National Institute of standards and Technology, Morris Dworkin, «Recommendation for Block Cipher Modes of Operation- Methods and Techniques,» Computer security Division, Gaithersburg, MD , 2001.
- [162] WhiteTimberwolf- Wikipedia Commons, "Block cipher mode of operation (Schematics)," 1 June 2013. [Online]. Available: [http://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation). [Accessed 2014 January 1].
- [163] M. Kantarcioglu, "Introduction to Cryptography- Class Notes," 2012. [Online]. Available: <http://www.utdallas.edu/~mxk055100/courses/crypto12s.htm>. [Accessed 2 February 2014].
- [164] NIST-Computer Security Division, "Block Cipher Modes," 3 June 2014. [Online]. Available: [http://csrc.nist.gov/groups/ST/toolkit/BCM/current\\_modes.html](http://csrc.nist.gov/groups/ST/toolkit/BCM/current_modes.html). [Accessed 31 March 2014].
- [165] I. Mironov, «Hash functions: Theory, attacks, and applications,» Microsoft Research, Silicon Valley, 2005.
- [166] R. C. Merkle, "One Way Hash Functions and DES," in *Advances in Cryptology — CRYPTO' 89 Proceedings*, Springer New York, 1990.
- [167] A. Selvakumar and C. Ganadhas, "The Evaluation Report of SHA-256 Crypt Analysis Hash Function," in *International Conference on ICCSN '09. I*, 2009.
- [168] A. Kak, «Lecture 15: Hashing for Message Authentication-Lecture Notes on “Computer and Network Security”,» Purdue University, 2014.
- [169] R. C. Merkle, Secrecy, authentication, and public key systems-PHD Thesis, STANFORD, CA: Stanford electronics laboratories-Stanford University, 1979.
- [170] D. Boneh, "Coursera-Crypto I-Collision Resistance," 06 February 2012. [Online]. Available: <http://spark-university.s3.amazonaws.com/stanford-crypto/slides/06.3-collision-resistance-the-merkle-damgard-paradigm.pdf>. [Accessed 23 September 2013].
- [171] T. Bartkewitz, "Building Hash Functions from Block Ciphers, Their Security and Implementation Properties," Seminararbeit-Ruhr-University Bochum, 2009.
- [172] Surachit- Wikipedia Commons, "MD5 block diagram," 31 December 2006. [Online]. Available: <http://upload.wikimedia.org/wikipedia/commons/d/d8/MD5.svg>. [Accessed 2 April 2014].
- [173] A. Sotirov, «Analyzing the MD5 collision in Flame.,» Slides Available at <http://www.trailofbits.com/resources/flame-md5.pdf> , Presentation at SummerCon, 2012.
- [174] V. Klíma, «Finding MD5 Collisions – a Toy For a Notebook,» de *IACR Cryptology ePrint Archive 2005*, Prague, 2005.
- [175] X. Wang y H. Yu, «How to Break MD5 and Other Hash Functions,» EUROCRYPT 2005, 2005.
- [176] National Institute of Standards and Technology, «Secure Hash Standard (SHS)-FIPS PUB 180-4,» FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Gaithersburg, 2012.
- [177] H2g2bob-Wikipedia Commons, "SHA-1 Block Diagram," 20 January 2007. [Online]. Available: <http://upload.wikimedia.org/wikipedia/commons/e/e2/SHA-1.svg>. [Accessed 2 April 2014].
- [178] X. Wang, H. Yu and Y. L. Yin, "Efficient Collision Search Attacks on SHA-0," in *EUROCRYPT*, 2005.
- [179] J. Walker, "NIST-sponsored hash function mailing list-"RE: SHA-3 Selection Announcement", " 5 October 2012. [Online]. Available: <http://cio.nist.gov/esd/maildir/lists/hash-forum/msg02565.html>; Access via: [http://csrc.nist.gov/groups/ST/hash/email\\_list.html](http://csrc.nist.gov/groups/ST/hash/email_list.html); User: hash-forum; Passwd: competition. [Accessed 17 May 2014].
- [180] NIST-Computer security division, "NIST's policy on hash functions," 31 March 2014. [Online]. Available: <http://csrc.nist.gov/groups/ST/hash/policy.html>. [Accessed 1 June 2014].
- [181] kockmeyer-Wikipedia commons, "SHA-2 Block diagram," 22 March 2007. [Online]. Available: <http://upload.wikimedia.org/wikipedia/commons/7/7d/SHA-2.svg>. [Accessed 21 May 2014].

- [182] H. Krawczyk, M. Bellare and R. Canetti, " HMAC: Keyed-Hashing for Message Authentication," RFC2104 , <https://www.ietf.org/rfc/rfc2104.txt>, February 1997.
- [183] Federal Information Processing Standards Publication, «The Keyed-Hash Message Authentication Code (HMAC),» National Institute of Standards and Technology, Gaithersburg,, 2006.
- [184] M. Bellare, "New Proofs for NMAC and HMAC: Security Without Collision-Resistance," in *Advances in Cryptology - CRYPTO 2006*, Santa Barbara, California, 2006.
- [185] P. Cheng and R. Glenn, " Test Cases for HMAC-MD5 and HMAC-SHA-1," Request for Comments: 2202 , available at: <http://tools.ietf.org/html/rfc2202>, 1997.
- [186] S. Kelly y S. Frankel, «Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec,» Request for Comments: 4868 , Available at: <http://tools.ietf.org/html/rfc4868>, 2007.
- [187] D. Boneh, "Coursera-Crypto I-Message Integrity," 06 February 2012. [Online]. Available: <http://spark-university.s3.amazonaws.com/stanford-crypto/slides/05.3-integrity-cbc-mac-and-nmac.pdf>. [Accessed 5 October 2013].
- [188] J. G. Proakis and M. Salehi, *Digital Communications*, New York: McGraw-Hill, 2008.
- [189] Hewlett Packard, «Digital Modulation in Communications Systems – An introduction,» Application Note 1298; Hewlett Packard.
- [190] S. K. Kaul, «QPSK, OQPSK, CPM probability of error for AWGN and flat fading channels,» *Wireless Communication Technologies*, 2005.
- [191] I. International Meta Systems, "RISC architecture computer configured for emulation of the instruction set of a target computer". USA Patent US 5574927 A, 6 December 1996.

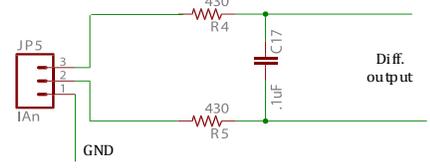
# A. SMART METER CIRCUITS



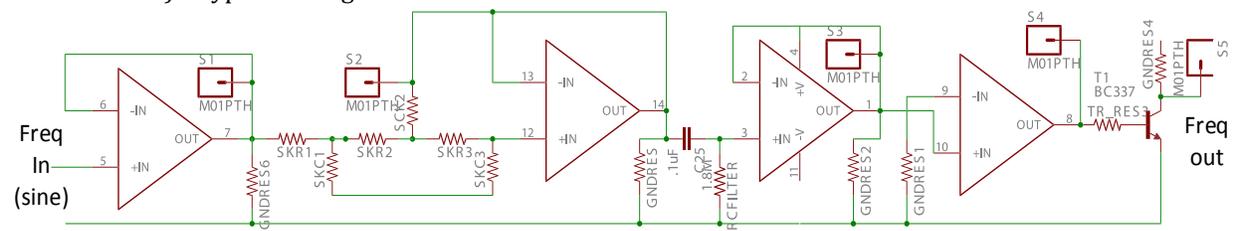
A) Main ADC board



B) Typical voltage divisor



C) Typical current divisor



D) Frequency measurement circuit

Figure A-1 Proposed ADC circuit (Electrical Diagram).

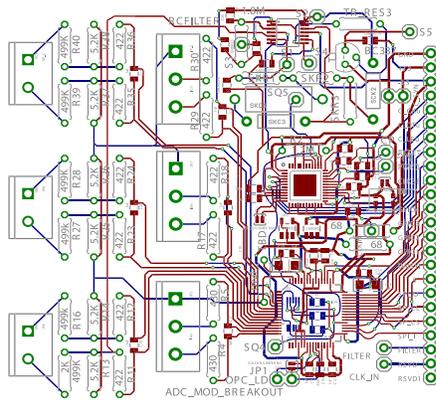


Figure A-2 Proposed ADC circuit (PCB layout).



Figure A-3 Proposed ADC circuit (manufactured PCB).

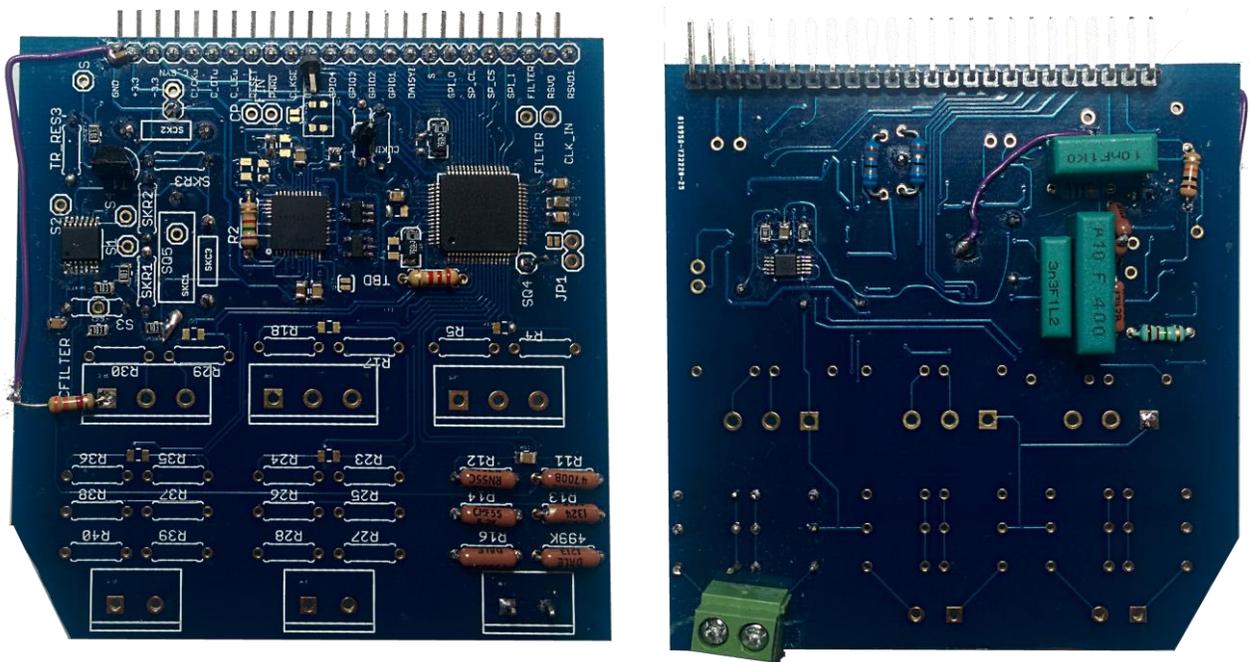


Figure A-4 Proposed ADC circuit (mounted PCB).

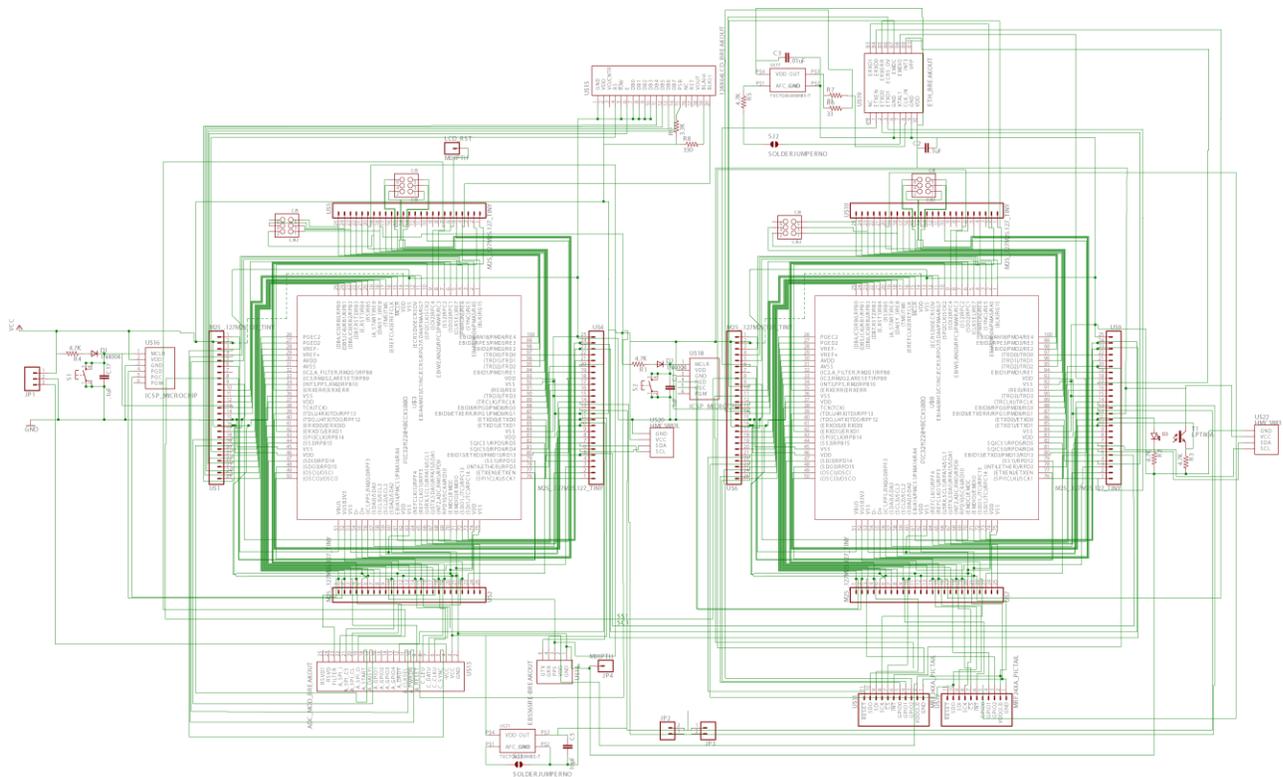


Figure A-5 Proposed CPU circuit (Electrical diagram).

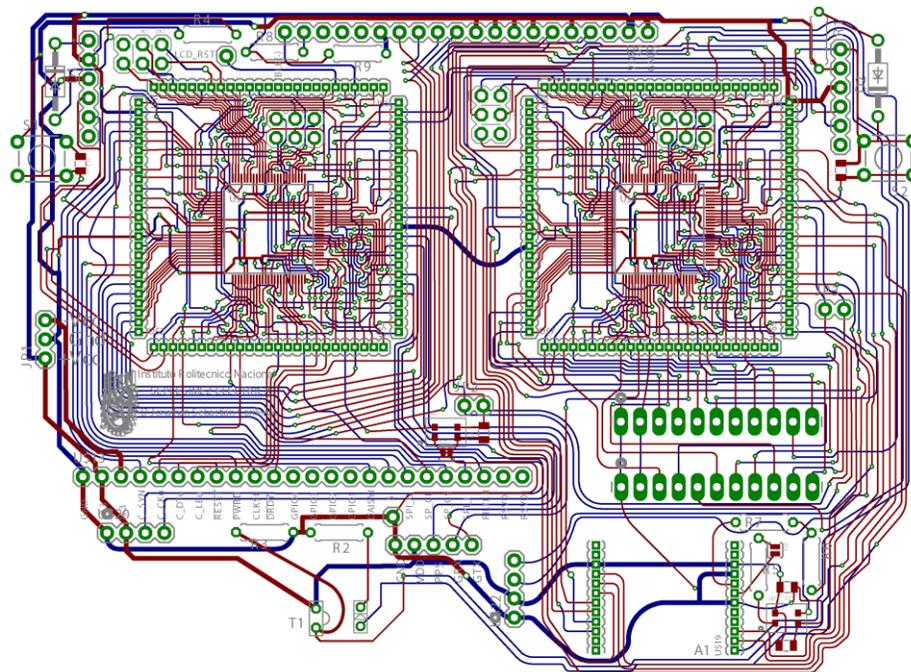


Figure A-6 Proposed CPU circuit (PCB layout).

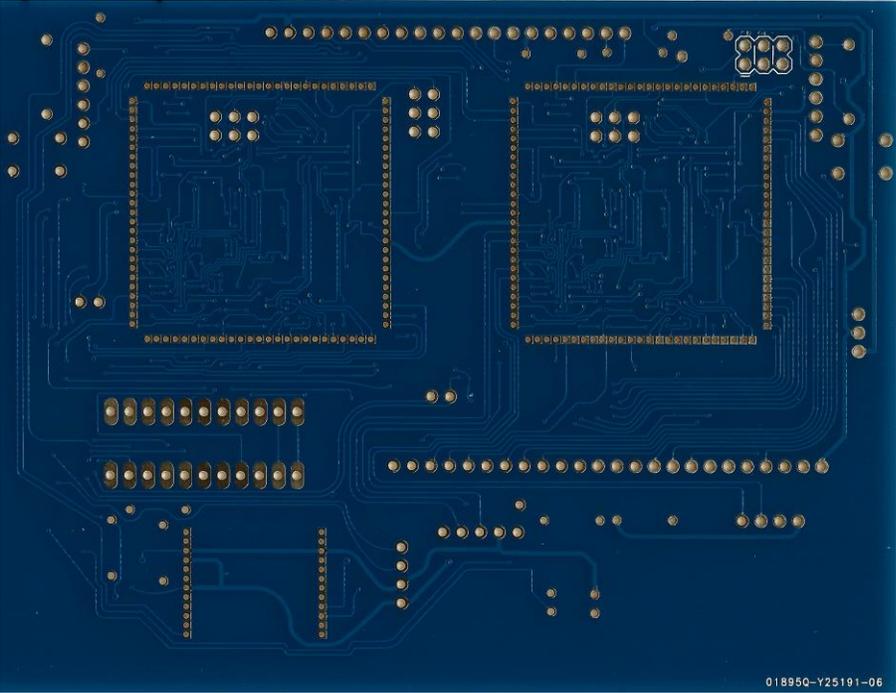
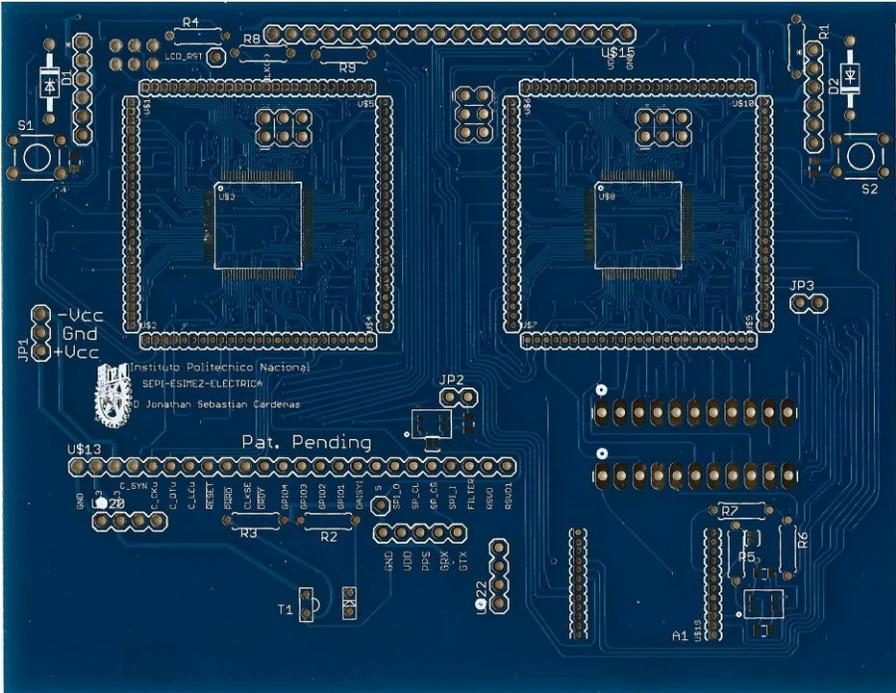


Figure A-7 Proposed CPU circuit (manufactured PCB).

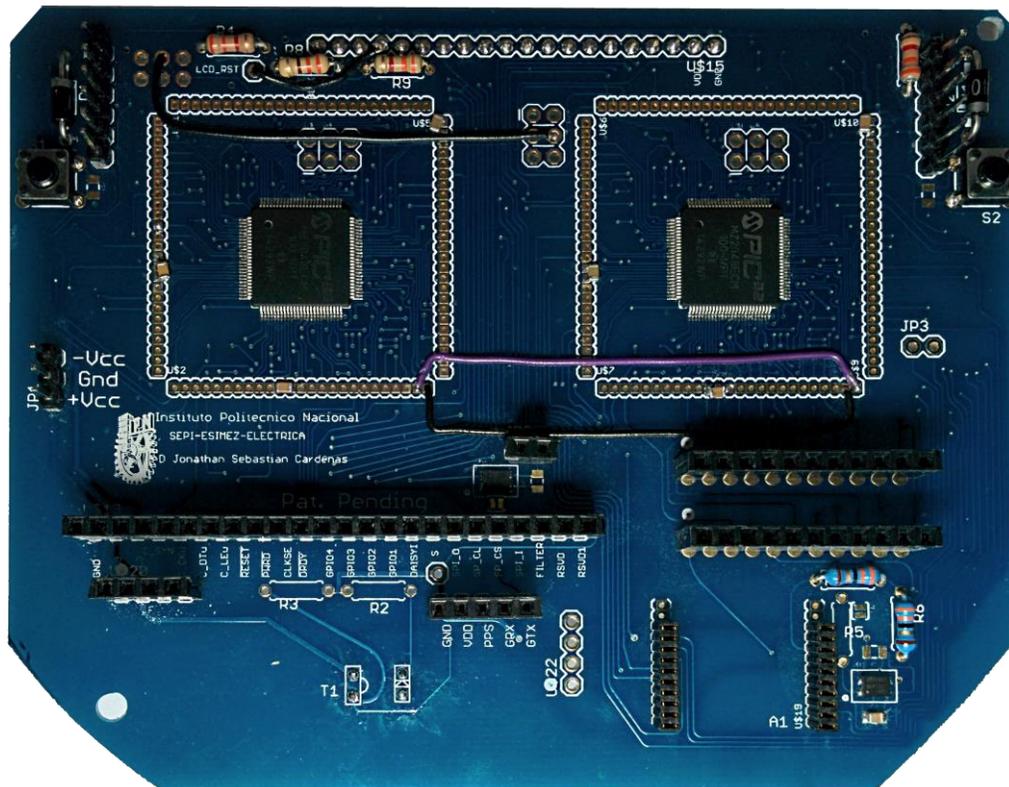
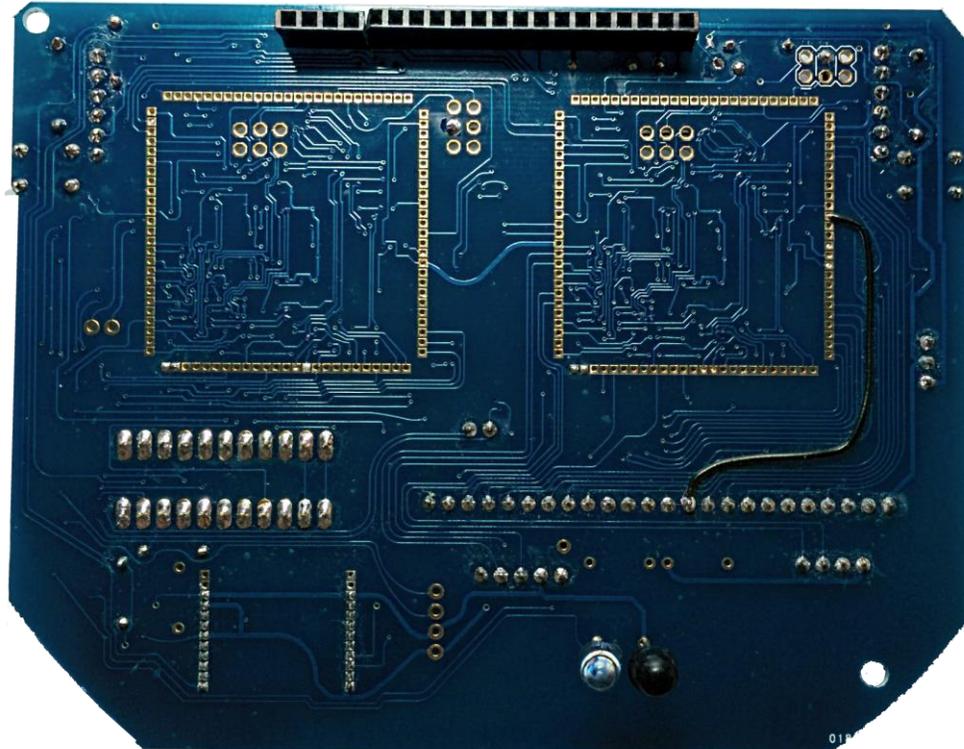


Figure A-8 Proposed CPU circuit with components mounted.

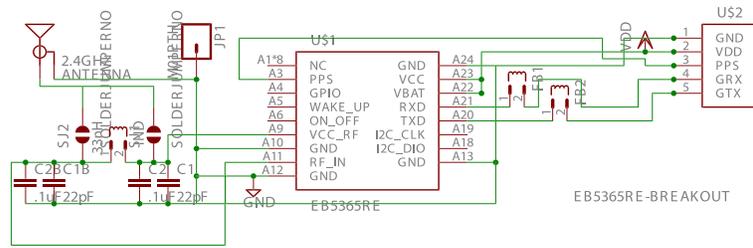


Figure A-9 Proposed GPS circuit (Electrical diagram).

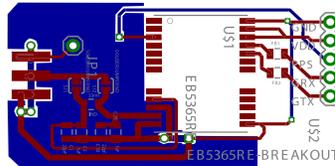


Figure A-10 Proposed GPS circuit (PCB layout).

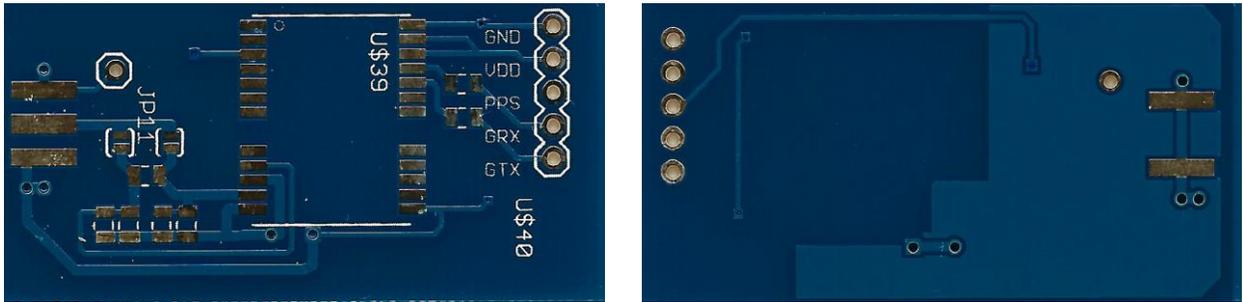


Figure A-11 Proposed GPS circuit (manufactured PCB).



Figure A-12 Proposed GPS circuit (Mounted components).

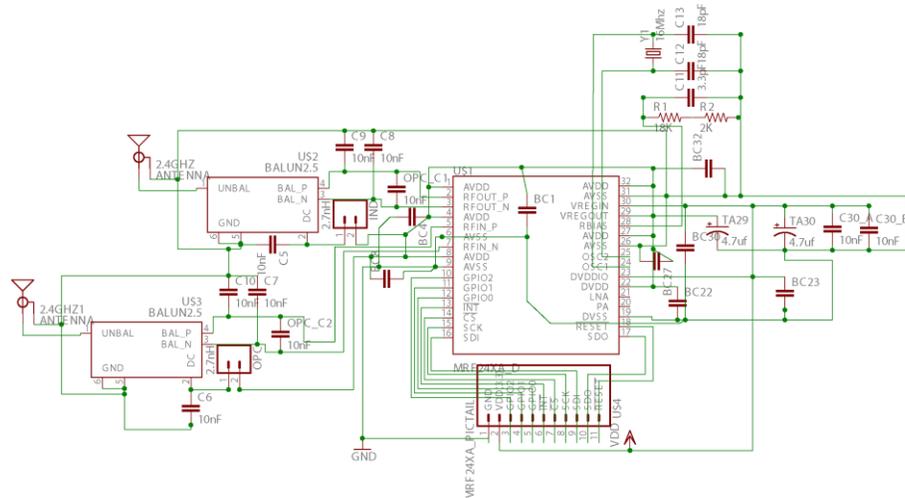


Figure A-13 Proposed GPS circuit (Electrical diagram).

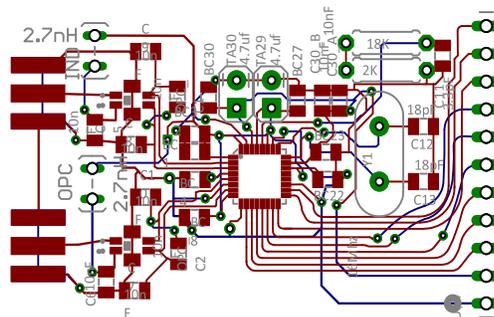


Figure A-14 Proposed GPS circuit (PCB layout).

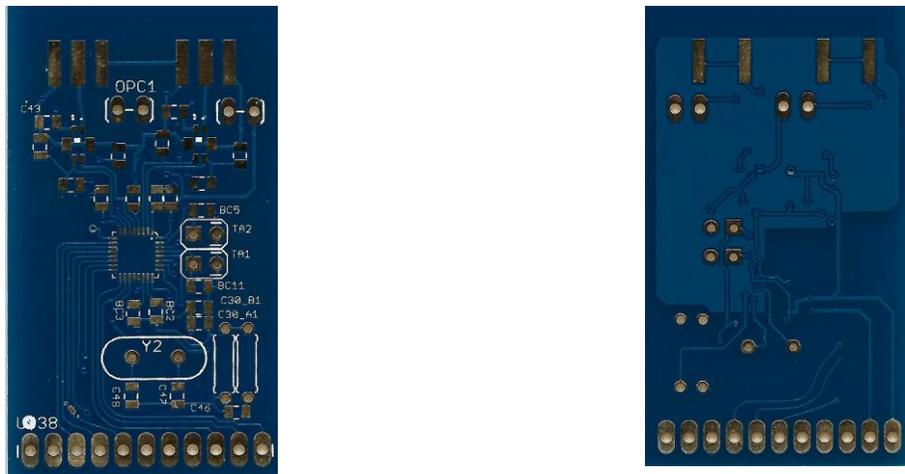


Figure A-15 Proposed GPS circuit (manufactured PCB).

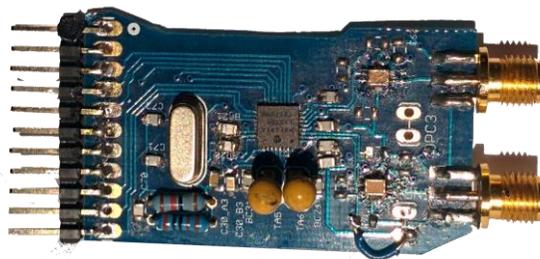


Figure A-16 Proposed GPS circuit (manufactured PCB).

Table A.1. Used pins in the communications microcontroller

Pin #	(Used As) Full Pin Name
1	AN23/AERXERR/RG15
2	EBIA5/AN34/PMA5/RA5
3	EBID5/AN17/RPE5/PMD5/RE5
4	EBID6/AN16/PMD6/RE6
5	EBID7/AN15/PMD7/RE7
6	<b>(RADIO0)(SDO2)</b> EBIA6/AN22/RPC1/PMA6/RC1
7	<b>(RADIO0)(SS2)</b> EBIA12/AN21/RPC2/PMA12/RC2
8	EBIWE/AN20/RPC3/PMWR/RC3
9	<b>(RADIO0)(SDI2)</b> EBIOE/AN19/RPC4/PMRD/RC4
10X	<b>(RADIO0)(SPI2CLK)</b>
11X	EBIA4/AN13/C1INC/ECRS/RPG7/SDA4/PMA4/RG7
12	<b>(ECRSDV)</b> EBIA3/AN12/C2IND/ERXDV/ECRSDV/....
13	VSS
14	VDD
15	MCLR
16	<b>(EREFCLK)</b> EBIA2/AN11/C2INC/ERXCLK/EREFCLK/
17	<b>(TMS) TMS</b> /EBIA16/AN24/RA0
18	AN25/AERXD0/RPE8/RE8
19	AN26/AERXD1/RPE9/RE9
20	AN45/C1INA/RPB5/RB5
21	<b>(RADIO0_RESET)</b> AN4/C1INB/RB4
22	AN3/C2INA/RPB3/RB3
23	<b>(OGPIO2)(IC6)</b> AN2/C2INB/RPB2/RB2
24	<b>(OGPIO0)(IC4)</b> PGEC1/AN1/RPB1/RB1
25	<b>(OGPIO1)(IC5)</b> PGED1/AN0/RPB0/RB0
26	<b>PGEC2</b> /AN46/RPB6/RB6
27	<b>PGED2</b> /AN47/RPB7/RB7
28	VREF-/CVREF-/AN27/AERXD2/RA9
29	VREF+/CVREF+/AN28/AERXD3/RA10
30	AVDD
31	AVSS
32	<b>(IC2)</b> EBIA10/AN48/RPB8/PMA10/RB8
33	<b>(IC3)</b> EBIA7/AN49/RPB9/PMA7/RB9
34	<b>(INT3)GPI1</b> EBIA13/CVREFOUT/AN5/RPB10/PMA13
35	<b>(ERXERR)</b> AN6/ERXERR/AETXERR/RB11
36	VSS
37	VDD
38	<b>(TCK) TCK</b> /EBIA19/AN29/RA1
39	<b>(U4TX,TDI) TDI</b> /EBIA18/AN30/RPF13/SCK5/RF13
40	<b>(U4RX,TDO) TDO</b> /EBIA17/AN31/RPF12/RF12
41	<b>(ERXD0)</b> EBIA11/AN7/ERXD0/AECRS/PMA11/RB12
42	<b>(ERDX1)</b> AN8/ERXD1/AECOL/RB13
43X	<b>(SPI3CLK)</b>
44X	<b>(SS3)</b>
45	VSS
46	VDD
47	<b>(SDI3)</b> AN32/AETXD0/RPD14/RD14
48	<b>(SDO3)</b> AN33/AETXD1/RPD15/SCK6/RD15
49	OSCI/CLKI/RC12

50	OSCO/CLKO/RC15
51	VBUS
52	VUSB3V3
53	VSS
54	D-
55	D+
56	<b>(IC1) USBID/RPF3/RF3</b>
57	EBIRDY3/RPF2/SDA3/RF2
58	EBIRDY2/RPF8/SCL3/RF8
59	EBICS0/SCL2/RA2
60	EBIRDY1/SDA2/RA3
61	EBIA14/PMCS1/PMA14/RA4
62	VDD
63	VSS
64	<b>(REFCLKI1)</b> EBIA9/RPF4/SDA5/PMA9/RF4
65	<b>(REFCLKO1)</b> EBIA8/RPF5/SCL5/PMA8/RF5
66	<b>(USRX) AETXCLK/RPA14/SCL1/RA14</b>
67	<b>(USTX) AETXEN/RPA15/SDA1/RA15</b>
68	<b>(RADIO0)(INT2) GPI2</b> EBIA15/RPD9/PMCS2/RD9
69	RPD10/SCK4/RD10
70	<b>(EMDC) EMDC</b> /AEMDC/RPD11/RD11
71	<b>(EMDIO) EMDIO</b> /AEMDIO/RPD0/RTCC/INT0/RD0
72	<b>(SDO1) SOSCI/RPC13/RC13</b>
73	<b>(SDI1) SOSCO/RPC14/T1CK/RC14</b>
74	VDD
75	VSS
76	<b>(SPI1CLK) RPD1/SCK1/RD1</b>
77	<b>(ETXEN) EBID14/ETXEN/RPD2/PMD14/RD2</b>
78	<b>(INT4) EBID15/ETXCLK/RPD3/PMD15/RD3 (ETHERN)</b>
79X	<b>(SS1) EBID12/ETXD2/RPD12/PMD12/RD12</b>
80X	EBID13/ETXD3/PMD13/RD13
81	SQICS0/RPD4/RD4
82	SQICS1/RPD5/RD5
83	VDD
84	VSS
85	<b>(ETXD1) EBID11/ETXD1/RPF0/PMD11/RFO</b>
86	<b>(ETXD0) EBID10/ETXD0/RPF1/PMD10/RF1</b>
87X	EBID9/ETXERR/RPG1/PMD9/RG1
88	EBID8/RPG0/PMD8/RG0
89	<b>(TRCLK) TRCLK/SQICLK/RA6</b>
90	<b>(TRD3) TRD3/SQID3/RA7</b>
91	<b>(GP0)BID0/PMD0/RE0</b>
92	VSS
93	VDD
94	<b>(GP1)EBID1/PMD1/RE1</b>
95	<b>(TRD2) TRD2/SQID2/RG14</b>
96	<b>(TRD1) TRD1/SQID1/RG12</b>
97	<b>(TRD0) TRD0/SQID0/RG13</b>
98	<b>(GP2)EBID2/PMD2/RE2</b>
99	<b>(GP3)EBID3/RPE3/PMD3/RE3</b>
100	<b>(GP4)EBID4/AN18/PMD4/RE4</b>

Legend:

SPI CHANNEL 1	SPI CHANNEL 2	SPI CHANNEL 3	INPUT CAPTURE CHANNELS	ETHERNET PINS	INTERRUPTS
POWER	UART CHANNEL 4	UART/I2C CHANNEL	DEBUGGER PINS	GENERAL PURPOSE PINS	

Table A.2.Used pins in the metering microcontroller.

Pin #	(Used As) Full Pin Name
1	<b>(BACKLIGHT)</b> AN23/AERXERR/RG15
2	EBIA5/AN34/PMA5/RA5
3	<b>(FREQ_SYNC)</b> EBID5/AN17/RPE5/PMD5/RE5
4	<b>(ADC_PWRDN)</b> EBID6/AN16/PMD6/RE6
5	<b>(ADC_CLKSEL)</b> EBID7/AN15/PMD7/RE7
6	<b>(SDO2)</b> EBIA6/AN22/ <b>RPC1</b> /PMA6/RC1
7	<b>(SS2)</b> EBIA12/AN21/ <b>RPC2</b> /PMA12/RC2
8	<b>(LCD_RESET)</b> EBIWE/AN20/RPC3/PMWR/RC3
9	<b>(SDI2)</b> EBIOE/AN19/ <b>RPC4</b> /PMRD/RC4
10	<b>(SPI2CLK)</b> AN14/C1IND/ECOL/RPG6/ <b>SCK2</b> /RG6
11	<b>(IC4)</b> EBIA4/AN13/C1INC/ECRS/RPG7/SDA4/PMA4/RG7
12	<b>(ECRSDV)</b> EBIA3/AN12/C2IND/ERXDV/ <b>ECRSDV</b> /....
13	VSS
14	VDD
15	MCLR
16	<b>(EREFCLK)</b> EBIA2/AN11/C2INC/ERXCLK/ <b>EREFCLK</b> /
17	<b>(TMS)</b> TMS/EBIA16/AN24/RA0
18	<b>(ADC_DAI5Y)</b> AN25/AERXD0/RPE8/RE8
19	<b>(ADC_START)</b> AN26/AERXD1/RPE9/RE9
20	<b>(E)</b> AN45/C1INA/RPB5/RB5
21	<b>(RS)</b> AN4/C1INB/RB4
22	<b>(DB7)</b> AN3/C2INA/RPB3/RB3
23	<b>(DB6)</b> AN2/C2INB/RPB2/RB2
24	<b>(DB5)</b> PGEC1/AN1/RPB1/RB1
25	<b>(DB4)</b> PGED1/AN0/RPB0/RB0
26	<b>PGEC2</b> /AN46/RPB6/RB6
27	<b>PGED2</b> /AN47/RPB7/RB7
28	VREF-/CVREF-/AN27/AERXD2/RA9
29	VREF+/CVREF+/AN28/AERXD3/RA10
30	AVDD
31	AVSS
32	<b>(ADC_FILTER)(IC2)</b> EBIA10/AN48/ <b>RPB8</b> /PMA10/RB8
33	<b>(ADC_RESET) (IC3)</b> EBIA7/AN49/ <b>RPB9</b> /PMA7/RB9
34	<b>(INT3) (PPS)</b> EBIA13/CVREFOUT/AN5/ <b>RPB10</b> /A13
35	<b>(ERXERR)</b> AN6/ <b>ERXERR</b> /AETXERR/RB11
36	VSS
37	VDD
38	<b>(TCK)</b> TCK/EBIA19/AN29/RA1
39	<b>(U4TX,TDI)</b> TDI/EBIA18/AN30/ <b>RPF13</b> /SCK5/RF13
40	<b>(U4RX,TDO)</b> TDO/EBIA17/AN31/ <b>RPF12</b> /RF12
41	<b>(ERXD0)</b> EBIA11/AN7/ <b>ERXD0</b> /AECRS/PMA11/RB12
42	<b>(ERDX1)</b> AN8/ <b>ERXD1</b> /AECOL/RB13
43	<b>(SPI3CLK)</b>
44	<b>(SS3)</b>
45	VSS
46	VDD
47	<b>(SDI3)</b> AN32/AETXD0/ <b>RPD14</b> /RD14
48	<b>(SDO3)</b> AN33/AETXD1/ <b>RPD15</b> /SCK6/RD15
49	OSCI/CLKI/RC12

50	OSCO/CLKO/RC15
51	VBUS
52	VUSB3V3
53	VSS
54	D-
55	D+
56	<b>(IC1)</b> USBID/ <b>RPF3</b> /RF3
57	EBIRDY3/RPF2/SDA3/RF2
58	EBIRDY2/RPF8/SCL3/RF8
59	EBICS0/SCL2/RA2
60	EBIRDY1/SDA2/RA3
61	EBIA14/PMCS1/PMA14/RA4
62	VDD
63	VSS
64	<b>(REFCLK1)</b> EBIA9/ <b>RPF4</b> /SDA5/PMA9/RF4
65	<b>(REFCLK01)</b> EBIA8/ <b>RPF5</b> /SCL5/PMA8/RF5
66	<b>(U5RX)(SCL1)</b> AETXCLK/ <b>RPA14</b> /SCL1/RA14
67	<b>(USTX)(SDA1)</b> AETXEN/ <b>RPA15</b> /SDA1/RA15
68	<b>(ADC) (INT2)</b> EBIA15/ <b>RPD9</b> /PMCS2/PMA15/RD9
69	RPD10/SCK4/RD10
70	<b>(EMDC) EMDC</b> /AEMDC/RPD11/RD11
71	<b>(EMDIO) EMDIO</b> /AEMDIO/RPD0/RTCC/INT0/RD0
72	<b>(SDO1)</b> SOSCI/ <b>RPC13</b> /RC13
73	<b>(SDI1)</b> SOSCO/ <b>RPC14</b> /T1CK/RC14
74	VDD
75	VSS
76	<b>(SPI1CLK)</b> RPD1/ <b>SCK1</b> /RD1
77	<b>(ETXEN)</b> EBID14/ <b>ETXEN</b> /RPD2/PMD14/RD2
78	<b>(INT4)</b> EBID15/ <b>ETXCLK</b> / <b>RPD3</b> /PMD15/RD3 (ETHERN)
79	<b>(SS1)</b> EBID12/ <b>ETXD2</b> / <b>RPD12</b> /PMD12/RD12
80	EBID13/ <b>ETXD3</b> /PMD13/RD13
81	SQICS0/RPD4/RD4
82	SQICS1/RPD5/RD5
83	VDD
84	VSS
85	<b>(ETXD1)</b> EBID11/ <b>ETXD1</b> /RPF0/PMD11/RF0
86	<b>(ETXD0)</b> EBID10/ <b>ETXD0</b> /RPF1/PMD10/RF1
87	EBID9/ <b>ETXERR</b> /RPG1/PMD9/RG1
88	EBID8/RPG0/PMD8/RG0
89	<b>(TRCLK) TRCLK</b> /SQICLK/RA6
90	<b>(TRD3) TRD3</b> /SQID3/RA7
91	EBID0/PMD0/RE0
92	VSS
93	VDD
94	EBID1/PMD1/RE1
95	<b>(TRD2) TRD2</b> /SQID2/RG14
96	<b>(TRD1) TRD1</b> /SQID1/RG12
97	<b>(TRD0) TRD0</b> /SQID0/RG13
98	EBID2/PMD2/RE2
99	(SPI1 CS)EBID3/RPE3/PMD3/RE3
100	EBID4/AN18/PMD4/RE4

Legend:

SPI CHANNEL 1	SPI CHANNEL 2	SPI CHANNEL 3	INPUT CAPTURE CHANNELS	ETHERNET PINS	INTERRUPTS
POWER	UART CHANNEL 4	UART/I2C CHANNEL	DEBUGGER PINS	GENERAL PURPOSE PINS	LCD PINS
ADC CTRL PINS					



## B. ADC terms used on this work

**Precision:** defined as the relative closeness between repeated measurements, which do not have to do with the true value of measurand. Precision is expressed in terms similar to the standard deviation or  $\pm$  quantities [154].

**Accuracy:** defined as the closeness between the measured quantity and the true value of the measurement [154].

**Sampling Rate:** Number of samples taken per second, these samples affect the bandwidth limit, a sampling rate of at least two times the input frequency is needed to prevent aliasing effect, and higher speed sampling raises costs and usually introduces noise.

**Resolution:** The number of output levels than an input signal can be quantized, often given in powers of two ( $2^n$ ), resolution is often referred to the full-scale signal amplitude and is only representative of the output digits and not the precision. Resolution also establishes the signal step size for a given input magnitude, i.e.  $S_{step} = \frac{Max_{Magnitude}}{2^{resolution}}$

**Quantization error:** Ideally, a measured sample lies in between two consecutive output values, having a theoretical maximum error of a  $\frac{1}{2}$  digit due to rounding effects, yet this error level can increase due to internal noise and output threshold levels.

**Signal to Noise Ratio (SNR):** Due to the sampling nature of ADCs, noise is introduced to the measurements at every conversion step, the SNR is defined as “the ratio of the Root Mean Square (RMS) signal amplitude relative to the mean value of the root-sum square of all other spectral components, including harmonics” [155]. This also defines the other spectral components as the ones introduced by the conversion process; it is similar to the Total Harmonic Distortion (THD) factor found in power quality studies.

**Effective Bit Resolution:** Since the SNR introduces noise into the measurements, the ideal resolution losses precision, causing certain digits to remain with useful data (most significant), while causing random errors into the less significant. These useful digits are called the effective bit resolution (since most ADCs operate in binary code), meaning they hold the high precision digits, where only the quantization errors are present.

**Non-Linearity:** In theory, the magnitude between two consecutive values interleaved by the signal step value must differ by “1” digit at the output conversion, during the entire ADC range; yet most ADCs have certain offsets, or “nonlinear” characteristics that create deviations from the real and

reported signal values. These deviations must be determined, and mathematically canceled in order to obtain a closer approximation to the real measurand; in Figure B-1 the effects of a nonlinear response can be observed.

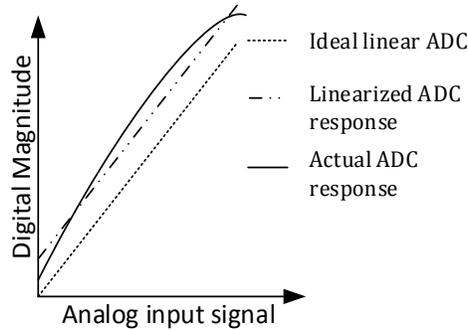


Figure B-1 Non-Linear ADC responses, adapted from [155].

**Bandwidth:** The actual bandwidth of the ADC is characterized primarily by its sampling rate and to a lesser extent by how it handles errors such as aliasing. Higher bandwidth ADCs allow to oversample signals and to perform digital noise rejection filters.

**Dynamic range:** The dynamic range of the ADC is influenced by many factors, including the resolution (the number of output levels it can quantize a signal to), linearity and accuracy (how well the quantization levels match the true analog signal) and clock jitter (timing errors that introduce additional noise by sampling at varying times, see section C.1).

## C. Phase Lock Devices

Digital interfaces are the responsible for executing the processes done in a computing machine, most digital concepts are beyond the scope of this work, but a selected list of topics are needed to correctly justify the design of a smart meter. The next sections explain some key aspects about digital communication and data processing typically found in an embedded device.

### C.1. Clocks

Most processes require being split into small steps to be able to be executed in the digital world; these steps are known as computing states, in this case, clocks are used to provide the triggering mechanism that enables a machine to reach a new state according to the current machine state. Some of this states need to be synchronized with other devices states, thus allowing different devices to share information, or processes data simultaneously.

Synchronous processes between devices require clock signals that are ideally the same for all of them, but clocks in reality experiment a set glitches that might degrade their performance, up to a non-functional system, some of the fundamental clock characteristics are illustrated in Figure C-1, while some concepts are further detailed below.

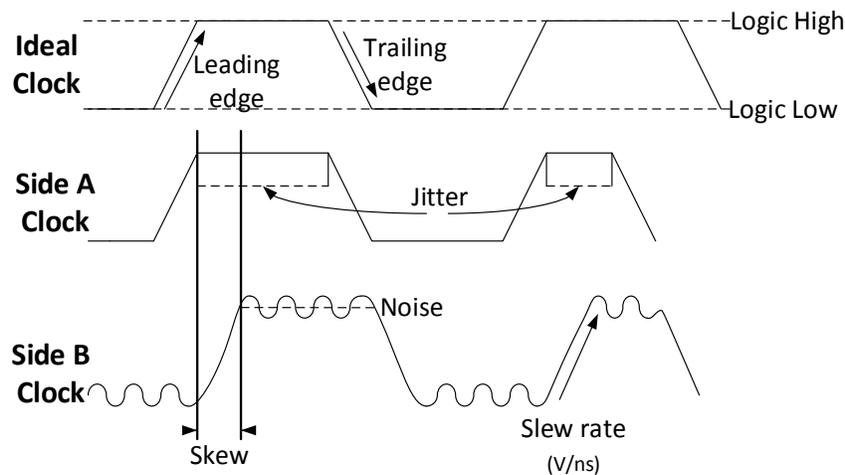


Figure C-1 Common clocking terminology

**Skew:** At high frequencies, distance traveled by the clock signals can represent a large percent of the signal wavelength, causing variations between the received signals; this can affect synchronism, careful distance compensation can solve the problem.

**Jitter:** Clock signals sometimes have time differences between period's length, or duty cycles, causing non uniform timing cycles, these timing errors can be measured relative to the prior cycle (*cycle to cycle jitter*), or relative to the ideal period (*period jitter*) [48]. Period jitter can also be expressed in terms of the average time difference (*RMS jitter*) [156]. Jitter might cause insufficient time between states, leading to malfunctions; further more *absolute jitter* is a measure of the absolute deviation from the fundamental frequency.

**Noise:** although noise in clocking systems is relatively unheard in low speed designs, it plays an important aspect of high-speed communications [156], while reaching higher speed means more computing power, power consumption also rises. To cope with this, manufacturers lowered interfaces voltages, which make them more susceptible to system noise. Noise can cause RMS jitter (*Phase Noise*) [156] but can also falsely trigger clock phase detectors, causing loss of synchronism, some low voltage interfaces, such as LVPECL and LVDS use differential signaling to improve noise performance, nevertheless proper shielding must be considered.

### **C.1.1.Clock generation**

There are several ways to generate clock signals, ranging from RC circuits to complex low jitter, temperature compensated reference clocks, and some important clock sources in the embedded world are further discussed below.

**Crystals/Oscillators:** A quartz crystal oscillates according to its shape and cutting parameters due to its piezoelectric properties [48]. These devices provide signals with a fixed value, manufacturers often describe its error (absolute deviation from the fundamental frequency) in parts per million (ppm), a 20 MHz clock exhibiting a 50 ppm can oscillate in between 19.999 and 20.001 MHz, or a  $\pm 50$  us per second variation. Some key aspects of oscillators are drift (Frequency variation with temperature and time), output level, noise characteristics, and capacitance load.

**Voltage Controlled Oscillators:** (VCO) offer designers the possibility of dynamically adjusting output frequencies, these frequencies are usually in a predefined range, pre-adjusted via external components. VCO's suffer from period jitter and power supply fluctuations, requiring some form of external control to make them more stable, and clean.

**Frequency divisors:** When master clocks need to be used by different devices, some signals might need to be scaled, this usually done by counting the number of pulses before toggling downstream clocks, frequency divisors allow RMS jitter cleaning, but can't correct absolute jitter.

**Phase Locked Loop:** (PLL) circuits compare a reference clock signal with a secondary signal (usually driven by a VCO), outputting its phase difference, this phase difference can be used by a control unit (charge pump) that drives the VCO, thus creating a closed loop system. These systems with the help of frequency divisors can create complex source clocks, due to its feedback topology it can be used to clean clock drift in VCO circuits, and to improve RMS jitter by overdriving the VCO and then using frequency divisors to scale it to desired frequency. In Figure C-2 a typical PLL architecture is illustrated, some components can be removed or added to improve its characteristics, typically the phase comparator and charge pump systems are components requiring detailed analysis, this type of analysis can be further be consulted in [157].

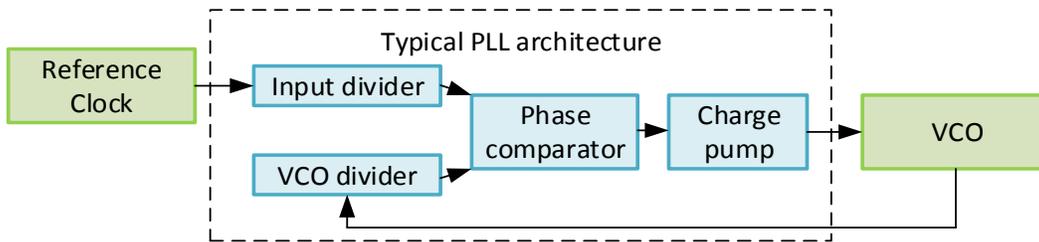


Figure C-2 PLL internal components



## D. Common terms used in cryptography

In this section a core vocabulary is given to serve as fast reference to the reader, further information referring to cryptographic terms is available at [67].

**Plain Text:** Refers to data that can be read according to a certain protocol, such as a clear text conversation, a data table, and encoded data such as TCP/IP packets.

**Ciphertext:** it refers to a transformed message that ideally can't be read by unauthorized parties, this ciphertext must be protected by a Message Authentication Code (MAC), or hash function to prevent tampering.

**Cipher Algorithm:** Is the encoding algorithm responsible for transforming a plain text message into a cipher text given a message, certain key, and an initial cipher status.

**Deciphering Algorithm:** Is the decoding algorithm responsible for transforming a cipher text into a plain text message

**Key:** It is a secret entity, which works as the input password in the ciphering/ deciphering algorithm, providing repeatability.

**Random Numbers.** Random numbers are ideally a chaotic number sequence, without any order, sequence or meaningful pattern, which must pass a set of statistical randomness tests. In cryptography there are three types of random numbers sources, that typify how random numbers are generated [67].

**True random number generators (TRNG):** TRNG's are derived from physical phenomenon, outputted sequences should not be reproducible, nor predictable. Some examples of TRNG are semiconductor thermal noise, cosmic radiation, and radioactive decay rates.

**Pseudo random number generators (PRNG):** Pseudo random numbers (PRN) are algorithm generated random numbers that are reproducible, i.e. they are deterministic, which means they can be regenerated from a seed value (start sequence), these PRNG are the most used random number generators (RNG) in a typical computer, they are used to run simulations, or statistical studies. Standardized version of RNGs are often available in languages such as C [67], FORTRAN and interpreters such as Matlab® [158], which make them reproducible.

**Cryptographic secure PRNG (CS-PRNG):** In a similar manner to PRNG, they are algorithm based, but they are unpredictable, i.e. "given a sequence  $\{S_0, S_1, \dots, S_n\}$  an attacker cannot determine  $S_{n+1}$ " [67], hence they are not distinguishable from true random number generators.

**Block Ciphers:** work by splitting data into fixed sized blocks, in this blocks cryptographic operations are performed recursively until sufficient confidentiality is achieved, since the input size block is equal to the output block size, decryption functions also work in blocks of data. Block ciphers are used in a cipher block mode to provide security in real applications.

**Diffusion and Confusion:** Claude Shannon established in 1949, that encryption must have two basic building blocks, diffusion and confusion [67]. Diffusion refers to the property that a single bit change in the plaintext must result in a total change of the encrypted message, thus avoiding frequency analysis attacks, while confusion refers to property that it should be complex to find a relationship between the cipher text and key. These two building blocks must be repeated several times until a sufficient level of confidence exists, such as an attacker is not able to map input and outputs.

**Substitution Boxes:** Provide a non-linear replacement for data, i.e. they replace n-bits by another set of n-bits. Substitution boxes provide with the diffusion principle on most ciphers.

**Permutation box:** These units change the order of bits dynamically (scrambling) providing confusion properties into most ciphers.

**Key length:** Since a brute force attack, is the simplest form of retrieving the original plaintext (but not the fastest), the key that protects a message should be sufficiently long to deter attacks. For example a 32 bit key allows up to  $2^{32}$  possible combinations, that is to say if a message was encrypted with a 32 bit key, and considering some 10000 keys can be per second, it would take a maximum of 120 hours to break (in average it would only take 60 hours, See Section **¡Error! No se encuentra el origen de la referencia.**). Currently, based on hardware capabilities, and assuming no flaws on the current encryption standards, only key lengths greater than 128 bits are recommended for symmetrical ciphers.

## E. Advanced Encryption Standard.

The Advanced Encryption Standard (AES) is NIST approved encryption specification; it was selected due to its simplicity and security as the champion of the competition to replace DES in 2001. Its specification is open to the public under FIPS-197, and can be found at [70], AES operates on blocks of data, of size sixteen, its data blocks are called the internal states or the *state matrix*, they are represented in the form of a 4x4 matrix, where a continuous substitution-transposition process is repeated in order to provide security. The AES standard supports key lengths of 128, 192 and 256 bits; it is a cryptographic primitive that must be used in a mode of operation, or other tools to provide the required security.

The AES is a block cipher that operates in rounds, it was designed to be resistant to linear and differential cryptanalysis, by using a set of non-linear substitution tables and special permutations executed during the “mix column” step, in the next sections some of the core elements of the AES algorithm are described [70].

- ❖ Key expansion: A set of keys are derived from the cipher using the Rijndael’s key schedule, these generated keys are XORed with the ciphertext at the end of each round (AddRoundKey)
- ❖ SubBytes: This process is also known as the S-Box substitution and is designed to provide a non-linear byte substitution on each of the elements of the *state matrix* via a lookup table.
- ❖ ShiftRows: In this process, certain rows of the state matrix are rotated a certain number of bytes, it provide part of the diffusion principle.
- ❖ MixColumns: This part of the round fulfills the diffusion principle by mixing the columns contents. It does this by employing a Galois Field multiplication.

### E.1. AES Galois field( $2^8$ )

AES uses the Galois Finite Field for the MixColumns function. This base  $2^8$  field only contains a finite number of states and as such, arithmetic operations are different from those on traditional math, two of these arithmetic field operations are explained below [67].

**Addition and Subtraction:** On finite binary fields, these operations are equal to bitwise XOR operations.

**Multiplication:** Multiplication on binary fields is done exclusively by bit-shifting techniques and subsequent additions, no other techniques such as the ones used by decimal multiplications can be employed. Since multiplication can cause some numbers to leave the Galois field, a reducing polynomial is sometimes used to reduce the word size. In AES the Rijndael’s polynomial is employed, this polynomial is equal to  $x^8 + x^4 + x^3 + x + 1$  or 0x011B in binary form; it shall be used every time an operation yields more than 0xFF. An example of some multiply operations are given in Table E.3.

Table E.3. Sample operations under the Galois  $2^8$  finite field.

Operation 1	Operation 2	Operation 3
0x5d*0x01=0x5d	0xD4*0x02= 0x01A8	0xBF*0x03=0xBF*(0x02⊕0x01)=0x17E⊕0xBF=0x1C1
Since 0x5D<0xFF no reduction is necessary	Since 0x01A8 >0xFF a reduction is necessary	Since 0x1C1>0xFF a reduction is necessary
=0x5d	0x01A8 ⊕ 0x011B=0xB3	0x01C1 ⊕ 0x011B = 0xDA

## E.2. AES general algorithm description

In this section a rough explanation of the AES-128 algorithm is given, although the process is explained for the forward function (encryption) it can be adapted to explain the decryption process by reversing the order at each round, most of this section is based on the description given on [70] and the illustrated guide found at [159].

### E.2.1.AES setup process

A key point of every security function is the setup process, in this initial step the encryption/decryption function is fed with its corresponding inputs and configurations, simultaneously all buffer initializations as well as state machines must be reset, this is a core point to remember on software/hardware implementations.

#### E.2.1.1.Message parsing

The first step in AES algorithm is to form a “state matrix” with the original message data, by ordering data contents in columns, in Figure E-1 a “Sample Message” is fed into the state matrix.

S	l	e	g	53	6C	65	67
a	e	s	e	61	65	73	65
m		s		6D	20	73	00
p	M	a		70	4D	61	00

Figure E-1. State Matrix for a “Sample message” encoded as 0x534F4D452031323820424954204B4559

### E.2.1.2.Key Setup

AES uses a key scheduling algorithm based on simple matrix operations, and thus requires that the input key be parsed into a “*key state matrix*” as illustrated by Figure E-2 . After the initial key setup is performed, the “*key state matrix*” is XORed with the “*state matrix*”, storing the result at the “*state matrix (0)*”

1		R	K	31	20	52	4B
6	S	E	E	36	53	45	45
	E	T	Y	20	45	54	59
B	C			42	43	20	00

Figure E-2 Key state matrix for a “16 B SECRET KEY”

### E.2.1.3.Key scheduling

Although the process of key scheduling can be done at each round of the AES algorithm, some abstraction can be gained by considering this step part of the setup process. The key scheduling process works by repetitively XORing columns of the key state matrix, creating an avalanche effect (see section G.1.1). On certain columns, additional XORing operations are included to make the key round-dependent, with an S-BOX substitution occurring in a single column of each round. The common data flow for a single round is given by Figure E-3, in this figure only the initial round is computed (1) but can be adapted to  $n + 1$  by changing the rounding constant.

By considering the key scheduling process part of the setup process, each of the generated key rounds must be stored for posterior use in a vector matrix of the form:  $key[round]$ , and disposed at the end of the AES rounds process.

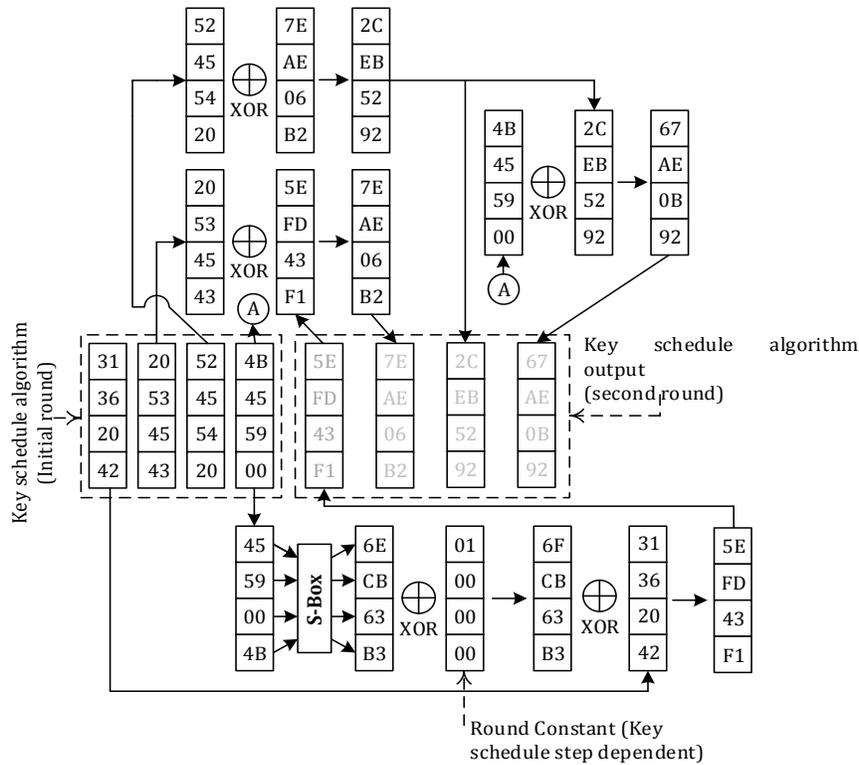


Figure E-3 Key scheduling algorithm.

### E.2.2.AES round operation

AES relies on round operations to achieve the diffusion-confusion principals proposed by Claude Shannon, the number of rounds depends on the key size used. For 128-bit keys a ten round process is used, all the steps are the same for each round, except the last one where the MixColumns step is omitted. At the start of the round operations, the “state matrix [0]” is used as the input matrix.

#### E.2.2.1.SubBytes function

The *S-Box* Substitution table is used as a non-linear byte substitution lookup table (see Figure E-4), it forms part of the confusion step of AES, and formal mathematical definition can be found at [70]. In practice, the *SBox* can be precomputed and stored into a lookup table that is similar to the one given in Figure E-4.

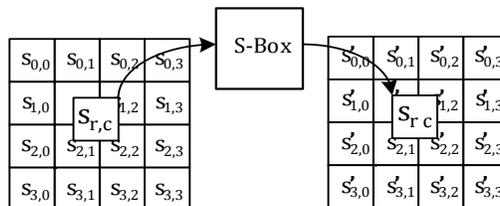


Figure E-4 The S-BOX transformation concept, a form of substitution [70].

LB \ HB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
10	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
20	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
30	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
40	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
50	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
60	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
70	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
80	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
90	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A0	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B0	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C0	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D0	70	3E	B5	66	48	3	F6	0E	61	35	57	B9	86	C1	1D	9E
E0	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F0	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure E-5 S-BOX transformation, with a nonlinear substitution table, adapted from [70].

### E.2.2.2. ShiftRows

A core component of the diffusion property is provided by the ShiftRows process, in this step the state matrix rows are rotated to left by an  $n$  number of bytes, where  $n$  represents the row number (0 based), this process can be observed in Figure E-7.

### E.2.2.3. MixColumns

The MixColumns operation is the core diffusion property of the round operation, in this step a Galois field matrix multiplication is executed, creating a byte substitution scheme that depends on the *state matrix*, although the operation is reversible, it is hard to reverse once the other diffusion and confusion steps are repeated multiple times.

### E.2.2.4. Key XORing

The key XORing operation XORs the state matrix with the derived key at each round  $n$ , employing the keys obtained through the key scheduling operation, this operation ensures that encryption is tied to the key. The key XORing process can be observed at Figure E-6.

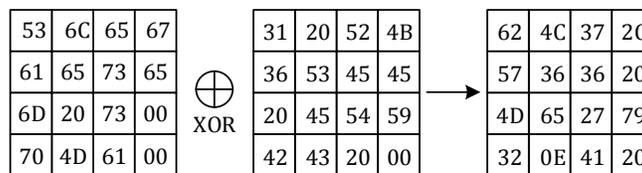


Figure E-6 Message 'XORing' with a key, the key scheduling algorithm provides different keys for each round.

### E.2.2.5.Round operation epilogue

Once all of the operations are completed, they are repeated an  $x$  number of times, that depends on the key size. To summarize all the processes involved during each round operation Figure E-7 shows the data flow across all the processes, in this case the sample is done for the first round of the state matrix used at the beginning of the section.

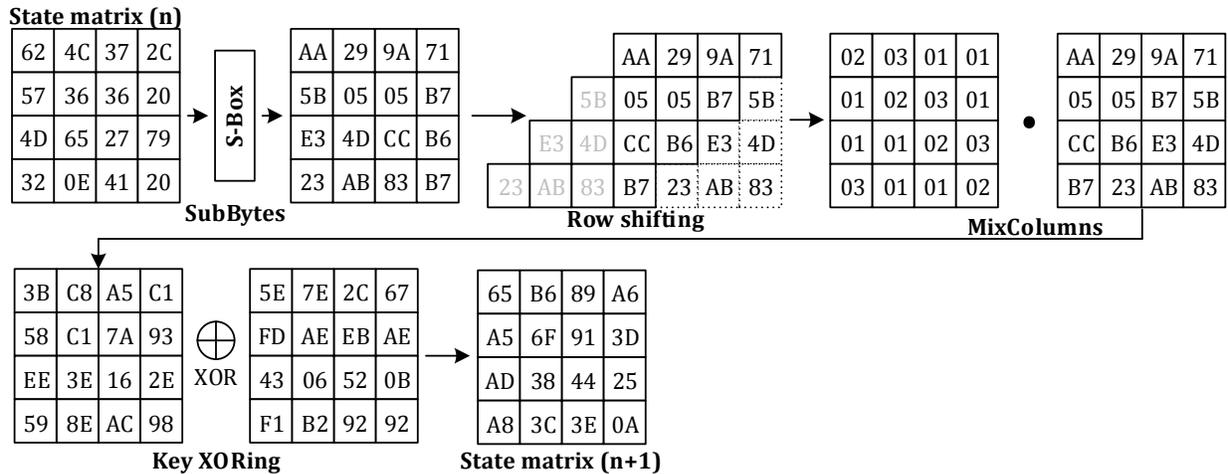


Figure E-7 Message transformation, after the round key step.

### E.3. AES optimization on matrix multiplication

Matrix multiplication is a very common operation on signal processing, and several electrical studies, but it also plays an important role on encryption algorithms, the traditional way of matrix multiplication is shown on Eq. E.2. This process takes about 64 multiplications and 64 additions for a 4x4-matrix multiplication, using a set of transformations it can be seen that matrix multiplication can be viewed as a linear combination of vector spaces. This “linear” way of seen matrix multiplication is helpful when designing high-speed encoders and decoders [160], in the next section the process of representing multiplication as a linear combination is reviewed. This is of particular interest for speed oriented software implementations, and it is the optimization method used by the author to implement AES on chapter 7.

Let us begin by defining two square Matrixes A, B

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad \text{Eq. E.1}$$

Doing a full expansion of matrix product, we have

$$C = AB = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \end{bmatrix} \quad \text{Eq. E.2}$$

Grouping common terms

$$C = \left[ \left[ b_{11} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + b_{21} \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} + b_{31} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \right] : \left[ b_{12} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix} + b_{22} \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} + b_{32} \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \right] \dots \right] \quad \text{Eq. E.3}$$

Replacing occurrences of  $a_{i,j}$  by vectors

$$C = [b_{11}[A_{:,1}] + b_{12}[A_{:,2}] + b_{13}[A_{:,3}]] : [b_{21}[A_{:,1}] + b_{22}[A_{:,2}] + b_{23}[A_{:,3}]] : [b_{31}[A_{:,1}] + [A_{:,2}] + [A_{:,3}]] \quad \text{Eq. E.4}$$

Where  $[A_{:,j}]$  denotes all elements of  $j$  column, and ":" separate matrix columns.

If a limited set of  $[A_{:,j}]$  vectors exist, such as in AES algorithm (eight in total for mix columns function), a LUT can be generated to speed up matrix multiplication process, in Eq. E.5 an unshifted AES block is presented, requiring a GF ( $2^8$ ) matrix multiplication. Since additional modulus, operations are required with respect traditional multiplications, the number of required operations increases significantly.

$$C = AB = AS(U_b) = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} S \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix} \quad \text{Eq. E.5}$$

$$= \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{22} & b_{23} & b_{24} & b_{21} \\ b_{33} & b_{34} & b_{31} & b_{32} \\ b_{44} & b_{41} & b_{42} & b_{43} \end{bmatrix} = \text{col1: col2: col3: col4}$$

where

$$\begin{aligned} Col_1 &= [LUT.C1(b_{11}) + LUT.C2(b_{12}) + LUT.C3(b_{13}) + LUT.C4(b_{14})] \\ Col_2 &= [LUT.C1(b_{22}) + LUT.C2(b_{23}) + LUT.C3(b_{24}) + LUT.C4(b_{21})] \\ Col_3 &= [LUT.C1(b_{33}) + LUT.C2(b_{34}) + LUT.C3(b_{31}) + LUT.C4(b_{32})] \\ Col_4 &= [LUT.C1(b_{44}) + LUT.C2(b_{41}) + LUT.C3(b_{42}) + LUT.C4(b_{43})] \end{aligned} \quad \text{Eq. E.6}$$

From Eq. E.6 the use of each  $LUT.Cx(value)$  takes a data byte, and outputs 4 bytes, requiring a total of four tables at 1 kb each (since there are 256 input possibilities each outputting a word). Thus the total number of operations reduces itself to 16 table lookups and 16 additions, an improved version of this algorithm shown on Eq. E.7 takes 8 table lookups and 8 additions, and uses 2 tables, each having 65536 entries and outputting 4 bytes, giving a total of 256kb per table. Another aspect of a  $LUT.Cx(value)$  is that S-Box substitution can be precomputed inside the tables, combining all round steps into simple table Lookups and additions(XOR's)

$$\begin{aligned}
 Col_1 &= [LUT.C_{12}(b_{11}b_{12}) + LUT.C_{34}(b_{13}b_{14})] \\
 Col_2 &= [LUT.C_{12}(b_{22}b_{23}) + LUT.C_{34}(b_{24}b_{21})] \\
 Col_3 &= [LUT.C_{12}(b_{33}b_{34}) + LUT.C_{34}(b_{31}b_{32})] \\
 Col_4 &= [LUT.C_{12}(b_{44}b_{41}) + LUT.C_{34}(b_{42}b_{43})]
 \end{aligned}
 \tag{Eq. E.7}$$

## F. Block Cipher Encryption Modes

Most data applications that require cryptography handle data sizes that are not exactly the size of the cipher block. Some small data sets will need padding, and large data pieces will need to be broken up, to handle these situations NIST has published 12 block cipher operation modes [161]; five of these modes are designed to work with block cipher encryption modes, three of these modes are described in the following sections.

### F.1. Electronic Code Book

The electronic code book (ECB) mode is the simplest form of encryption possible, it splits a data container into blocks, according to a chosen cipher block size, at each of this blocks the cipher primitive is applied, optionally the encrypted blocks can be joined together to form an encrypted data container. It has the particular characteristic that for a given message 'm' and a key 'k' it always outputs the same value creating vulnerabilities in certain cases. It also enables parallel encryption/decryption processing due to its per block architecture, and allows errors to be isolated to a particular block (i.e. transmission errors), in Figure F-8 a general diagram of ECB mode is shown.

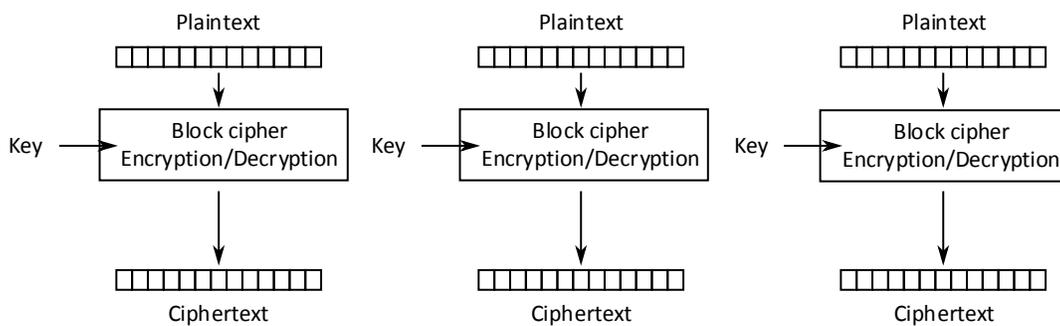


Figure F-8 ECB mode, adapted from [162], [161].

### F.2. Cipher-Block Chaining

The cipher block chaining (CBC) mode is a chaining mechanism, designed to create data dependent blocks. It uses an Initialization Vector (IV) that alters the encryption result versus using a traditional message- key pair, thus if the IV space is big enough, it generates unique ciphertexts every time a repeated message is encrypted with the same key  $e(m, k)$  [72]. The operation of the CBC mode is shown in Figure F-9, because the encryption process uses a chaining design it must be performed

serially, whereas the decryption process can be done in parallel, if a particular block is damaged, the error remains in that block, the error bits can propagate to the next cipher block, inverting certain bits [163].

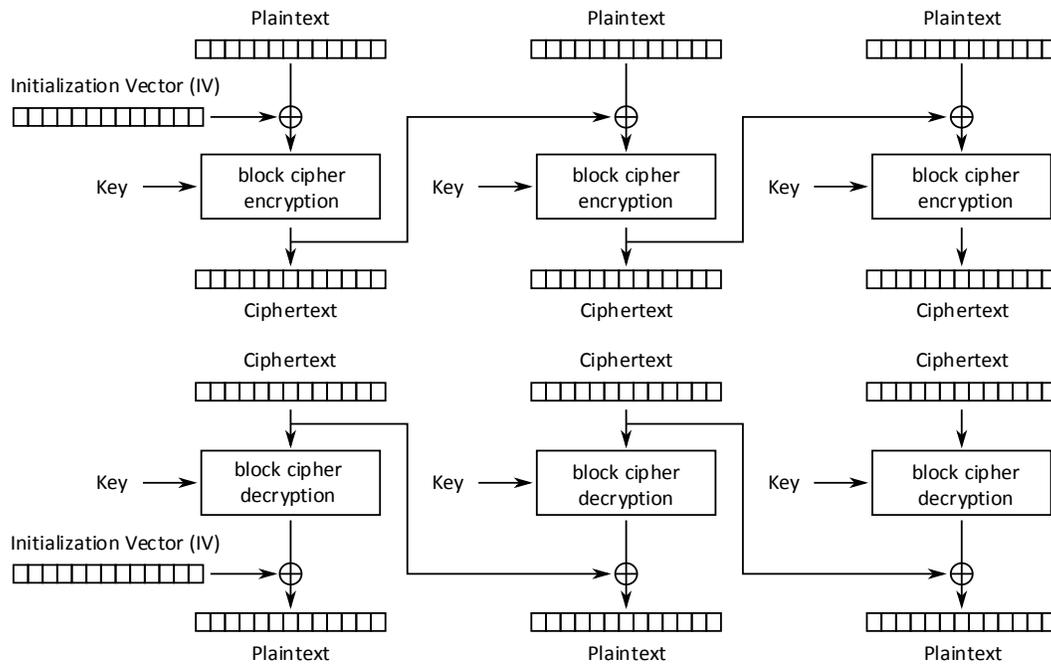


Figure F-9 ECB mode, adapted from [162], [161].

The IV generation is perhaps the most important aspect of the ECB mode; IV's cannot be fixed, since they would not generate unique  $(m, k)$  pairs, leading to plaintext-ciphertext attacks, thus creating security holes (see Section 3.5). The solution is to use a unique IV per transmitted message (assigning an IV via an internal counter according to the number of sent messages), or using a CS-PRNG to generate unique IV's. In order to use random IV's, a large IV space is required, due to the "birthday attack"; thus in practice this IV space usually has the same length as the key.

### F.3. Counter Mode

The counter mode (CTR) is an improvement over the ECB mode that adds a form of IV that is known as the "Nonce-Counter" pair. A Nounce (short for "number used once"), is a sequential IV generated value, that is bit shorter than a typical IV, the counter as it name implies is used incrementally in each block to provide a unique "Nonce-Counter" pair. A particular aspect of this cipher mode is that the Nounce-Counter pair is the one encrypted in the cipher function, allowing creating parallel

processing algorithms/hardware, increasing throughput rates. The typical operation of CTR mode is illustrated in Figure F-10

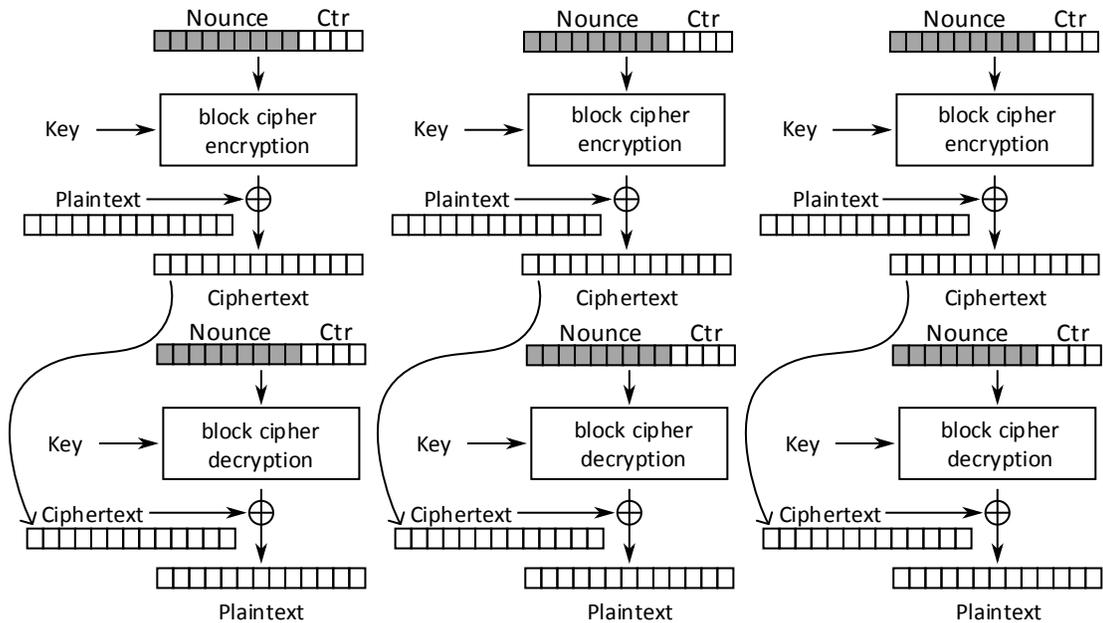


Figure F-10 CTR mode, adapted from [162], [161].

As always there are other cipher modes that are suited specific applications, these can be reviewed in detail in [164].



## **G. Authentication Functions**

Cryptography in its basic form provides confidentiality i.e. other parties cannot access/read data, but does not guarantee that data is intact, or has not been modified, authentication on the other hand, allows to check for data integrity. Authentication can be provided by cipher primitives under the MAC generation, hash functions, or Encryption-Authentication Block Cipher modes discussed on [164]. On the following sections, some authentication methods are described.

### **G.1. Message Integrity**

Cryptography functions alone only hide information to others, but does not guaranty data integrity, as mentioned earlier data integrity refers to the assurance that data has not been inserted, deleted, or crafted by a third party [66]. Integrity can be viewed as an improvement over error detection codes (intended to detect usually random errors and not malicious attacks). Some well-known error detection codes are repetition (send messages several times), use of parity (addition of a 0 or 1 to indicate parity of a data stream), checksums (insertion of the result of a mathematical function given an input data), or Cyclic Redundancy Check (CRC).

Some cryptographic functions have been designed to provide message integrity, such as the cryptographic hash functions, which allow to extract a digital fingerprint from a data stream (see section G.1.1), allowing the receiving end to check for data integrity. Another type of function, called the MAC provides integrity and authenticity for a given message, authenticity provides proof of message origin; this MAC is usually is a data field transmitted along the original message, and it is referred as a tag on literature. Secure MAC's can rely on cryptographic functions to prevent attackers from forging messages, and must output a minimum length tag that prevents brute force generation.

#### **G.1.1. Hash functions**

A generic hash function outputs a fixed size identifier, from an arbitrary sized data stream [165]. This identifier is commonly used on computer applications to perform fast, index based searching, for example during a search query, the input is hashed, and the output value is searched in a LUT (containing all stored messages hashes), if a value is found, the algorithm checks bit per bit to discard false positives. In the previous hash example, there exists the possibility of collisions (i.e.

two or more messages have the same hash value), which can be aggravated depending on the unique ID length, and hash function type.

On cryptography hash functions are used to generate a unique digital fingerprint from a data stream (providing integrity, which in turn can be adapted to provide authentication), this introduces certain requisites that traditional hashing functions don't have. Cryptographic hash functions should be collision resistant, they should be one-way functions (given a hash value, an attacker cannot generate a message with the same fingerprint), finally it is desirable that the function has an avalanche effect, i.e. Small changes in a data stream should create large variations on the hash value.

An example of an insecure, but easy to understand hashing function is the XOR function, if a 16 bit buffer is used over a stream of data, where it continually XOR's blocks of data, until the end of stream is reached, generating a 16 bit hash value, as illustrated in Figure G-1. Although it can detect most data changes due to transmission errors, it offers a very weak protection against attackers.

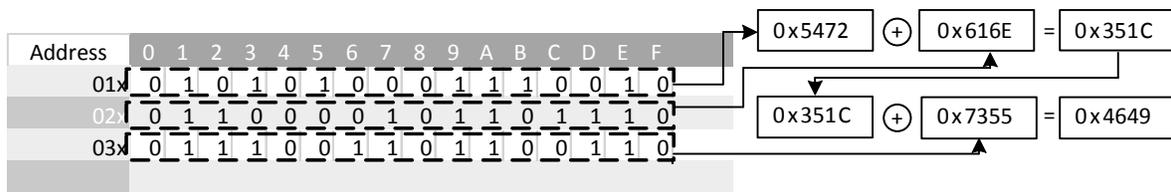


Figure G-1 Basic XOR based hashing function.

As mentioned earlier cryptographic hash functions must possess some core characteristics, which are important to ensure data authenticity against attackers, the importance of those characteristics, and possible attacks to weak hashing functions are exemplified by XOR hashing function in the following sections.

### G.1.1.1. One way functions

Over the years, a consensus has been made to define a one-way function, with different restriction levels, but in general, a function  $f$  is a one-way function if it meets the following criteria:

- “ $f$  can be applied to any argument of any size, producing a fixed size output.” [166]
- “Given  $f$  and  $x$ , it's easy to compute  $f(x)$ .” [166]

- “Given  $f$ , it is computationally infeasible to find any pair  $(x, x')$  such that  $x \neq x'$  and  $f(x) = f(x')$ ” [166]

Perhaps the last line is the most restrictive of all, it can be expanded further into three core issues mentioned below [167]:

**Collision Resistance:** It should be computationally infeasible to find, encounter two or more messages with the same output value. (See details in section 0)

**Preimage Resistance:** establishes that it should be computationally infeasible to generate, or find an input message given the output value

**Second Preimage Resistance:** establishes that it should be computationally infeasible to generate, or find an input message given the output value of another message

The previous definitions, add another layer of “should have properties”, in order to comply with the prior requirements, these properties are usually seen on approved hash functions.

**Complex Mapping functions:** It should be computationally infeasible for an attacker to do any type of attack, in order to predict, or alter the resulting output.

**Embedded message size:** This additional constraint decreases the risks of counterfeit, by limiting the message size. [168]

**Chained structure operation:** The ' $n$ ' position hash value must depend on the hash value calculated up to the ' $n - 1$ ' position. [168]

As an example of a bad one-way function, the previously suggested XOR function suffers from the following weaknesses.

**Simple mapping function:** The extremely easy to compute mapping function, allows an attacker to propose an altered message and adjust it, outputting the same digest value as the original message. A possible solution to this problem is to use cipher primitives to obfuscate the relationship between the input and output. In Figure G-3 an example exploit is done to the XOR hashing function based on the previously discussed weaknesses.

**Length extensions:** If an attacker wasn't able to alter the message by generating the same digest value, he would recur to adding data (rubbish), in order to alter the final digest value, the effects of this type of attack can be seen Figure G-2. The addition of data could be unnoticeable to the receiving end, even passing future coherency checks, for example, adding white spaces to text documents, or

NULL characters (which terminate strings in some computer languages such as C) as it can be seen by Figure G-2 example.

Intended Message														Tampered Message																											
Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII	Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII	ASCII					
01x	0	1	0	1	0	1	0	0	0	1	1	1	0	0	1	0	'T'	'r'	01x	0	1	0	1	0	1	0	0	0	1	1	1	0	0	1	0	'T'	'r'	'r'			
02x	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0	'a'	'n'	02x	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1	'a'	'n'	'n'			
03x	0	1	1	1	0	0	1	1	0	1	1	0	0	1	1	0	's'	'f'	03x	0	1	1	1	0	0	1	1	0	1	1	0	1	0	0	1	1	's'	'f'	'f'		
04x	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0	'e'	'r'	04x	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	0	'e'	'r'	'r'			
05x	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	' '	'1'	05x	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	' '	'1'	'1'
06x	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	'6'	' '	06x	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	'6'	' '	' '			
07x	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	1	'U'	'S'	07x	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	'U'	'S'	'S'		
08x	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	'D'	' '	08x	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	'D'	' '	' '			
09x	0	1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	't'	'o'	09x	0	1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	't'	'o'	'o'			
0Ax	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	' '	'A'	0Ax	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	' '	'A'	'A'		
0Bx	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	1	'c'	'c'	0Bx	0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	1	'c'	'c'	'c'			
0Cx	0	1	1	0	1	1	1	1	0	1	1	1	0	1	0	1	'o'	'u'	0Cx	0	1	1	0	1	1	1	1	0	1	1	1	0	1	0	1	'o'	'u'	'u'			
0Dx	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	0	'n'	't'	0Dx	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	0	'n'	't'	't'			
0Ex	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	' '	'B'	0Ex	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	' '	'B'	'B'			
0Fx	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	'C'	'2'	0Fx	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	0	'C'	'2'	'2'			
10x	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	0	'4'	'2'	10x	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	1	'4'	'2'	'2'			
Orig. Hash	0	1	0	0	0	1	0	1	0	1	1	0	0	1	0	0	'E'	'd'	Mod. Hash	0	1	1	0	0	1	0	1	0	1	0	1	0	0	1	0	0	'e'	'd'	'd'		
																			Ins. Data	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	' '	NULL	NULL			
																			Final Hash	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	'E'	'd'	'd'			

Transfer 16 USD to Account BC422

Transfer 16 USD to Account Fg400+” “+ NULL

Figure G-2 Forging messages by adding data, due to a not size-dependent hash function

### G.1.1.2.Collision resistance

Although it seems obvious that collisions will occur in hashing functions, given the fact that an infinite number of messages are mapped to a finite set of possible values, hashing functions must cause these collisions to occur with the least probability possible. Although one might initially think that a “n bit” message digest (the output value from a hashing function), has a  $1/2^n$  probability to encounter a collision, in reality this number is much lower, due to the birthday attack being closer to  $1/2^{n/2}$ , since the effective bit length is reduced by half [167]. The hashing bits should output sufficiently large message digests, ideally having a random relation between the input and output, and using all available bits. As an example of bad collision resistance, the previously suggested XOR function suffers from the following weakness.

**Short message digest space:** Although initially the bit complexity seems to be 16 bits, it is reduced to 14 bits, by using only 7 bits out 8 bits per byte if the original message is ASCII encoded, this bit space is further reduced by the birthday attack to 7 bits, reducing to  $1/128$  the possibility of a



The merkle-damgård construction only gives the user the big picture of hashing functions, another core problem is to design the 'h' compression function, a typical 'h' function is based on a set of nonlinear functions, which provide most of the cryptographic properties, but since some operations are reversible, additional steps must be executed to ensure that  $f(h)^{-1}$  is not existent [171] [67]. There are several ways to construct a secure compression functions, mostly based on using XOR operations to discard information, and hamper reversing algorithms, two of the most common constructions are explained below:

**Davies-Meyer:** the 'h' function is composed of a 'e' cipher primitive, which uses the previous hash value ( $H_{i-1}$ ) as the key to encrypt the current message block ( $M_i$ ), to undermine the decryption function, the output is XORed with ( $H_{i-1}$ ) to produce ( $H_i$ ). The dataflow diagram can be seen in Figure G-5

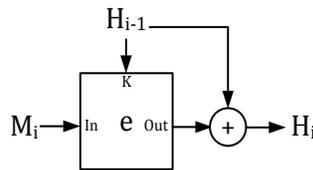


Figure G-5 Davies-Meyer 'h' compression function

**Matyas-Meyer-Oseas:** the 'h' function is composed of a 'e' cipher primitive, which uses the current message block ( $M_i$ ) as the key to encrypt the previous hash value ( $H_{i-1}$ ), to undermine the decryption function, the output is XORed with ( $M_i$ ) to produce ( $H_i$ ). The dataflow diagram can be seen in Figure G-6.

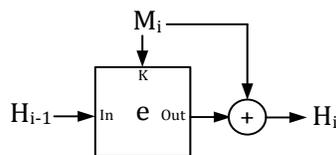


Figure G-6 Matyas-Meyer-Oseas 'h' compression function.

Most general-purpose hash functions use the previous constructions in their inner workings to satisfy the cryptographic hash requirements, some well-known hashing functions are the MD5 and the SHA family of functions, these functions are discussed in further detail in the next sections.

### G.1.2.1. Message Digest Algorithm-5

The Message Digest Algorithm 5 (MD5) was developed in 1992 by R. Rivest; it quickly gained popularity to ensure data integrity due to its simplicity, finding uses in digital signatures and

certificates [171]. MD5 has 128-bit digest size and it is based on the merkle-damgård construction, it uses a proprietary encryption block, based on the Davies-Meyer block operation. MD5 works in chunks of 512 bits, with internal 128-bit operation sub-blocks, at the last block padding and a 64-bit field length information is appended [171]. During each round, the message is compressed from 512 bits to 32 bits, by a set of successive XORing, and rotation operations, which depend on the value of  $h_{i-1}$  by using a 'F' nonlinear function, as it can be seen in Figure G-7.

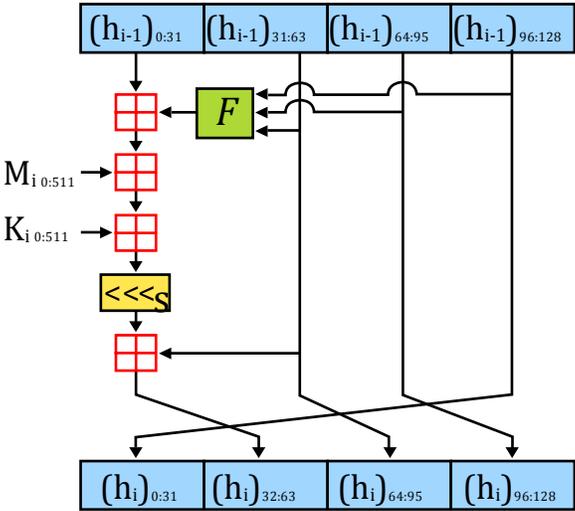


Figure G-7 MD5 single operation general diagram, where  $\boxplus$  denotes  $2^{32}$  modular addition, adapted from [172], [171].

**Attacks on MD5**

Possibly the worst recorded attack done on the MD5 checksum is the forgery of a X.509 certificate that faked a Microsoft digital signature, enabling the installation of a malware application, called Flame. Since Flame was digitally signed in windows systems, it was able to setup proxies on windows update, creating a huge security hole, the problem was discovered and addressed two years later (2012) by changing the required signature type on the root certificate (see section [173]. The previously described attack occurred due to a relative short digest space, and specific algorithms designed to exploit collision resistance, theoretical collision attack’s weaknesses were reporter as earlier as 1993 [173], but practical collisions were found until 2005, including faking files and x.509 certificates [174]. Most of these attacks were done based on linear cryptanalysis techniques [175], and exploiting specific merkle-damgård construction weakness.

### G.1.2.2. Secure Hashing Algorithm-1 (SHA 1)

The Secure Hashing Algorithm was designed as the successor of the MD5 by the National Security Agency (NSA) in 1995 [84], and published as FIPS-180-1 by NIST, it works on 512 bit data blocks outputting a 160 bits digest, it is based on the MD5 algorithm inner workings but with greater simplicity in mind [84]. Initial message preprocessing requires padding and  $H_{i-1}$  initialization, padding is done by adding a bit '1' to the original message followed by an 'n' number of zero's such the block size is 448, finally a 64-bit length identifier is appended [176]. For the  $H_{i-1}$  initialization a 160 IV is broken into five 32-bit blocks denoted 'A – D', the IV value is set to 0x67452301EFCDA8998BADCFE10325476C3D2E1F0 [176].

At each 512 data block, the data is broken into sixteen, 32-bit data containers, these data blocks are expanded to 80 blocks by Eq. G.1 (a simplified expansion based on the MD5 function). Each of these new blocks is iteratively compressed by the use of non-linear function  $f_i$  (Eq. G.2), and several  $H_{intermediate}$  rearrangements using the A-D blocks nomenclature (Eq. G.3), Figure G-8 illustrates a single SHA-1 operation.

$$w[i] = (w[i - 3] \oplus w[i - 8] \oplus w[i - 14] \oplus w[i - 16]) \lll 1 \quad \text{Eq. G.1}$$

$$f_i; k_i \begin{cases} (b \wedge c) \oplus (\bar{b} \wedge d); k = 5A827999 & i < 19 \\ b \oplus c \oplus d; k = 0x6ED9EBA1 & 20 \leq i < 39 \\ (b \wedge c) \oplus (b \wedge d) \oplus (c \wedge d); k = 0X8F1BBCDC & 40 \leq i < 59 \\ b \oplus c \oplus d; k = 0xCA62C1D6 & 60 \leq i < 79 \end{cases} \quad \text{Eq. G.2}$$

$$\begin{aligned} temp &= (a \lll 5) + f_i + e + k_i + w[i] \\ e &= d; d = c \\ c &= b \lll 30 \\ b &= a; a = temp \end{aligned} \quad \text{Eq. G.3}$$

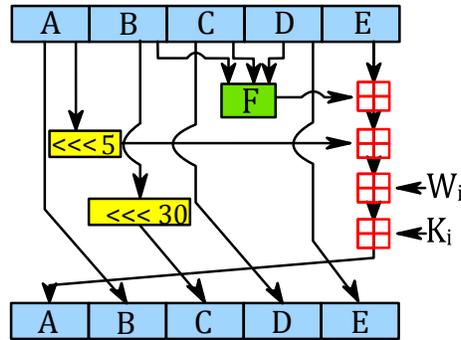


Figure G-8 SHA-1 single operation general diagram, where  $\boxplus$  denotes  $2^{32}$  modular addition, adapted from [177]

### Attacks on SHA-1

SHA-1 is a patched version of the original SHA-0 specification published in 1993, that was withdrawn due to an unpublished bug, vulnerabilities on the original SHA-0 algorithm were rediscovered in 1998, lowering the initial 80 bit complexity to a 61 bit complexity, further attacks were done over the years, until a 39 bit complexity was achieved in 2005 [178] [175]. Since SHA-1 is based on the SHA-0 standard, a likely attack can be done on SHA-1 [178], possible between 2018 and 2021 (due to hardware capabilities) [179], this has led to a migration to higher security hash functions, such as SHA-2 and SHA-3, in the US, NIST has suggested immediate migration for government agencies to SHA-2 [180].

#### G.1.2.3. Secure Hashing Algorithm-2 (SHA-2)

The Secure Hashing Algorithm-2 was designed as the successor of the SHA-1 by the National Security Agency (NSA) in 2001, and published as FIPS-180-2 by NIST; it works on 512-bit data blocks outputting a variable bit size digest. Some valid digest sizes are 224, 256, 384, and 512 bits, each having variations in their internal architecture, but sharing common operations, such as XOR, rotations and modulo additions, as an example of the SHA-2 inner workings a description of the 256-bit digest is given in the following sections.

The SHA-256 initial padding and block splitting is similar in nature to the one described in section G.1.2.2, the differences arises at the internal digest registers, in this case at  $H_{i-1}$ , since the new digest value is 256, the  $H_{i-1}$  container is divided into a set of 8 internal 32 bit blocks denoted 'A – G'. Also

for this case the  $H_{i-1}$  value is set to 0x6A09E667BB67AE853C6EF372A54FF53A510E527F9B05688C1F83D9AB5BE0CD19 [176].

At each 512 data block, the data is broken into sixteen, 32-bit data containers, these data blocks are expanded to 64 blocks by Eq. G.4 (an elaborate expansion based on the SHA-1 function). Each of these new blocks is iteratively compressed by the use of non-linear function  $f_i$  (Eq. G.2), and several  $H_{intermediate}$  rearrangements using the A-D blocks nomenclature (Eq. G.3), Figure G-9 illustrates a single SHA-256 operation.

$$\begin{aligned}\sigma_0 &= (w[i - 15] \ggg 7) \oplus (w[i - 15] \ggg 18) \oplus (w[i - 15] \gg 3) \\ \sigma_1 &= (w[i - 2] \ggg 17) \oplus (w[i - 2] \ggg 19) \oplus (w[i - 2] \gg 10) \\ w[i] &= (\sigma_1 \oplus w[i - 7] \oplus \sigma_0 \oplus w[i - 16]) \ll 1\end{aligned}\tag{Eq. G.4}$$

$$\begin{aligned}\Sigma_1 &= (E \ggg 6) \oplus (E \ggg 11) \oplus (E \gg 25) \\ \Sigma_0 &= (A \ggg 2) \oplus (A \ggg 13) \oplus (A \gg 22) \\ Ch &= (E \wedge F) \oplus (\bar{E} \wedge G) \\ Maj &= (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C) \\ Temp_1 &= H + \Sigma_1 + Ch + k[i] + w[i] \\ Temp_2 &= \Sigma_0 + Maj\end{aligned}\tag{Eq. G.5}$$

$$\begin{aligned}h &= g; g = f; f = e \\ e &= d + Temp_1; d = c; c = b \\ b &= a; a = temp1 + temp2\end{aligned}\tag{Eq. G.6}$$

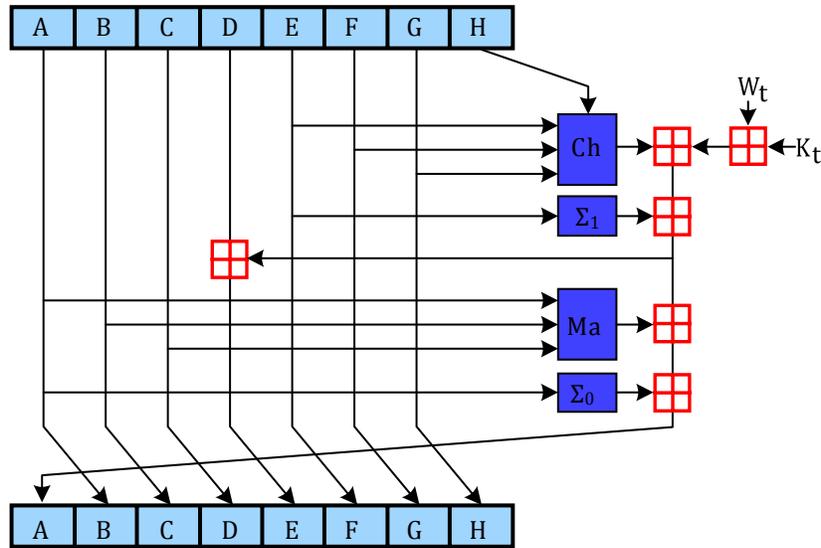


Figure G-9 SHA-256 single operation general diagram, where  $\boxplus$  denotes  $2^{32}$  modular addition, adapted from [181]

### G.1.3. Message Authentication Codes

Message Authentication Codes (MACs) provide message integrity and authentication at the same time, by creating an identification “tag”, this tag depends on the contents of the data stream and a symmetric key. There are several ways to construct the “tag” by using cipher primitives in the *authentication block cipher modes*, or by using the *hash based MAC* constructions. Regardless of the MAC function or construction used, a MAC must provide the following properties [67]:

**Cryptographic checksum:** The tag created should have cryptographic properties, meaning it should be impossible to forge by a third party, or impossible to an attacker to extract any useful information.

**Symmetric key use:** The tag must depend on a symmetric key; this key is used by all the individuals that are authorized to participate in the communications channel.

**Arbitrary message size:** The MAC function should be able to generate a tag for any message length.

**Fixed output length:** The generated tag should have a fixed determined sized, independent of the message or key length.

**Message integrity:** The user should detect any change in the message by comparing the transmitted tag vs the self-computed tag.

**Message authentication:** The user should be able to validate the tag origin by using the symmetric key.

**No nonrepudiation:** Since MACs use symmetric keys, anyone authorized can create a message without

### G.1.3.1.Hash based Message Authentication Codes

Hash functions provide the ability to check data integrity, but they do not provide authentication means, this allows an attacker to generate a custom message and attach an integrity digest, without the end user knowing if the message came from an attacker or a trusted party. To prevent this, hash functions are used are packaged or used in functions to provide authentication, some of this transformations are Hash-Based Message Authentication Codes (HMAC), and X.509 certificate files. HMACs are a form of Message Authentication Code (MAC) that enables to check simultaneously for data integrity and authenticity by generating a new hash value starting from the normal hash function and appending a secret shared key, the basic idea of a HMAC function can be seen on Eq. G.7. Since hash functions work on blocks of data, the key ( $K$ ) should be ideally as long as the block size, but since little security is gained with such a large key [182], a short key is used. The key is padded in such a way its size is equal to the block size, the padded key ( $K_{ZP}$ ) is then XORed with a predetermined value to increase the overall security, the HMAC function with padded keys, for a generic function is shown in Eq. G.8 [183].

$$HMAC(K, m) = H(K|H(K|m)) \quad \text{Eq. G.7}$$

$$HMAC(K, m) = H((K_{ZP} \oplus OPAD) || H(K_{ZP} \oplus IPAD || m)) \quad \text{Eq. G.8}$$

where

$H =$  *Criptographic hash function.*

$m =$  *message;  $K =$  Key; and  $|| \rightarrow$  denotes concatenation*

$K_{ZP} \rightarrow$  *Key padded with zeros, such that  $size(K_{ZP}) = size(H_{Block})$*

$IPAD =$  *a series of 0x3636 ... with size equal to  $H_{Block}$*

$OPAD =$  *a series of 0x5C5C ... with size equal to  $H_{Block}$*

The HMAC function (architecture, data flow) has been standardized under FIPS-198, as a “Keyed-hash message authentication code” [183], allowing generic hashing functions to be used as the ‘ $H$ ’ function, an important aspect of HMAC functions is that they are considered collision resistant, even if the underlying function has weak collision properties (e.g. MD5, SHA-1) [184], thus a MD5 based

HMAC function is still considered secure. HMAC functions are usually called according to the underlying digest function (e.g. HMAC-MD5, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-3, etc.) [185] [186].

### G.1.3.2. Block cipher authentication modes

Block ciphers can also be used in authentication modes, its history dates back from the 1970's when DES was seen as possible checksum algorithm, later on, the CBC cipher mode was adapted to generate a message authentication code, it does this by fixing the IV to 'zeroes' and discarding the encryption data. The CBC-MAC mode is illustrated in Figure G-10, although it works in a similar principle to the merkle-damgård construction, it suffers from chosen message attacks, which means that under certain cases a valid tag can be generated (see statement 4.1). CBC-MAC should only be used for previously established fixed data lengths, some implementations add another encryption level to the final TAG to prevent this type of attack ( $TAG' = e(TAG, K_1)$ ) [187].

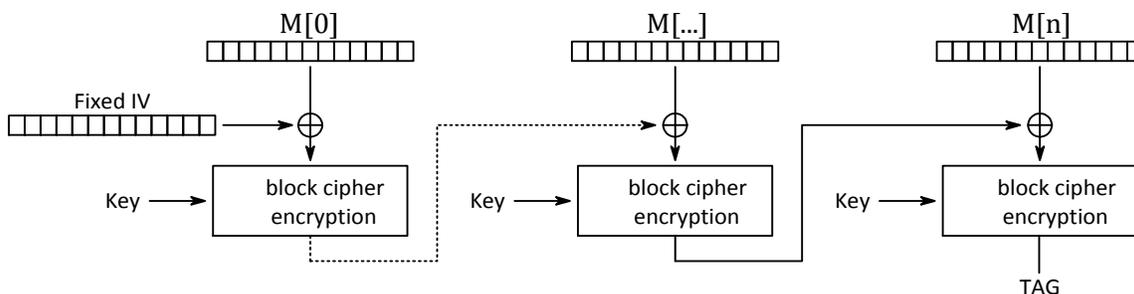


Figure G-10 CBC-MAC raw mode of operation, at the TAG step additional functions can be implemented to improve security.

In CBC-MAC mode, if for a single block sized message 'm' an attacker obtains  $f(m, k) = tag$  then Statement 4.1 he can compute  $f(m || (m \oplus tag), k) = tag$ . [187]

Proof:

$$\begin{aligned}
 f(m || (m \oplus tag), k) &= f(f(m, k) \oplus (m \oplus tag), k) = f((tag) \oplus (m \oplus tag), k) \\
 &= f((tag \oplus tag) \oplus m, k) = f(m, k) = tag
 \end{aligned}$$

### G.1.3.3. Cipher-based message authentication

The cipher-based message authentication (CMAC) is designed to solve some of the issues of the CBC-MAC mode, it does this by using two additional keys (derived from the primary key) and XORing it

with the last block, the use of these keys ( $k_1, k_2$ ) depends on the message size and block size, the block size

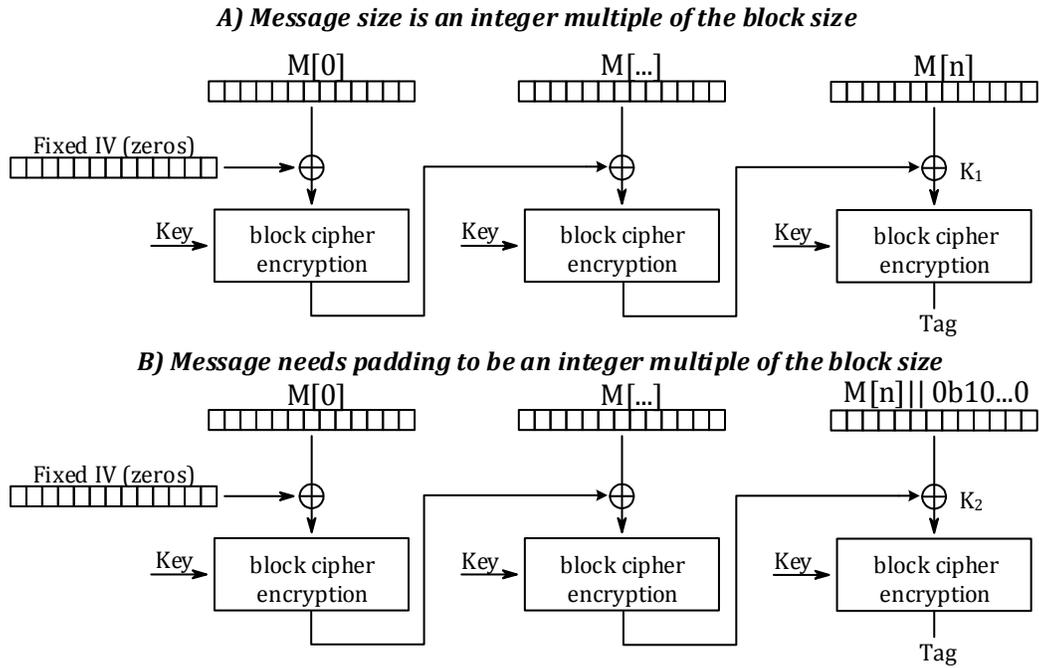


Figure G-11 CMAC mode of operation, showing key usage according to message size.

## H. Digital Communications

Analog communications allowed the transmission of most wireless signals (e.g. radio, TV) of the XX century, but with the introduction of digital information systems, a new set of transmission technologies were required. These technologies receive the name of digital communications, and introduce additional steps with respect the basic communication diagram shown in Figure 4-1, since digital communications are based on the existence of two states (e.g. 0's and 1's), data encoding and decoding steps must be added, as shown in Figure H-1.



Figure H-1. The intervening blocks of digital communications.

The way that information is encoded is an important issue of digital communications, since the channel only supports digital information, analog data must be converted into a digital format (ADC preprocessing) that might introduce noise and raise the overall SNR, some examples of analog data being transmitted digitally are satellite radio and HDTV broadcasts.

Another key aspect of digital communications is the modulation scheme, as seen previously modulation allows to embed a signal into a carrier frequency, in the digital world the embedded signal represents only two states, 0's and 1's allowing the modulator to work on discretized levels, some digital modulation schemes are further explained below.

### H.1.1. Digital modulation

Digital modulation enables the transmission of digital data across a communication medium, digital data is preferred since it allows data verification, and process layerization. Digital modulation works on the principles laid by analog modulation (AM, FM, PM), but it uses finite states to represent data patterns, .i.e. it allows to map a digital sequence to signals for transmission over a communication channel, on the following sections some commonly used modulation techniques are described. Each of these modulation techniques has an associated error rate that determines the probability of decoding an incorrect bit in a noisy environment, although useful in RF designs it is outside the scope of this work and further details should be examined at [188].

### H.1.1.1.Preamble

Since in digital communications is possible to alter the signal phase, magnitude or a combination of both, it is useful to represent this transformation in a polar or rectangular form. In the polar form a signal can be represented with magnitude and angle (i.e. phasors). On the rectangular grid communication engineers often use the I-Q plane (also called the I-Q constellation) which is equivalent to the X-Y plane, with additional overlaid circles to represent the signal amplitude levels [189], an example of the I-Q plane can be seen in Figure H-2.

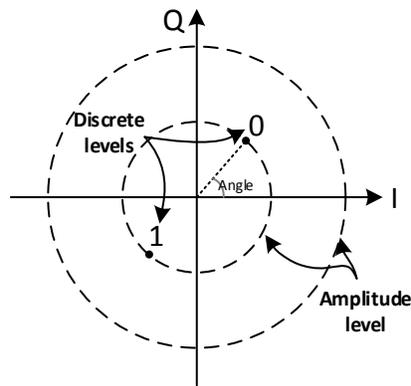


Figure H-2. The  $I - Q$  representation used on digital communications.

The I-Q constellation enables an orthogonal representation of signals by offsetting the Q component by 90 degrees ahead of the I component, allowing RF engineers to represent any wave traveling signal in a 2 dimensional plane. In addition, the I-Q orthogonal plane allows the creation of modulated signals by using an orthogonal modulator, known as the I-Q generic modulator, which can be seen in Figure H-3.

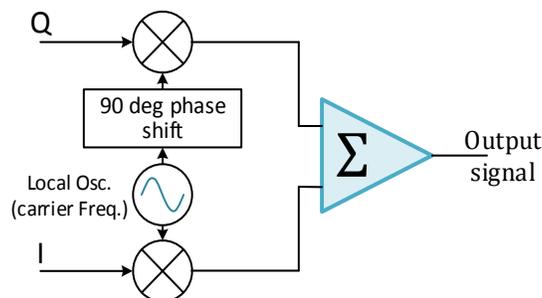


Figure H-3. A generic I-Q modulator.

### H.1.1.2. Amplitude-Shift Keying

The Amplitude-Shift Keying (ASK) is based on the same principals laid by AM modulation, differing itself in two key aspects, the amplitude levels, and the introduction of the concept “symbol period”. ASK uses a finite number of amplitudes to represent up to  $2^n$  digital states, also called symbols, e.g.  $n = 1$  allows to transmit the symbols  $\{0,1\}$  while  $n=2$  allows to transmit the symbols  $\{00,01,10,11\}$ . Since this modulation only affects the signal amplitude, the discrete levels are spread over the  $I$  axis of the I-Q constellation, see Figure H-4. ASK modulation suffers from the same drawbacks of AM modulation, and as such requires a high SNR to operate correctly.

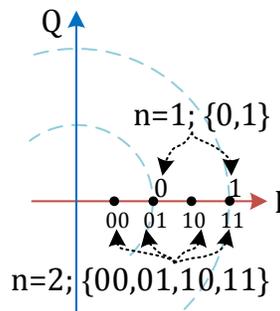


Figure H-4. The ASK modulation shown in the I-Q plane.

Digital modulations use the term “symbol period” ( $T$ ) to define the time taken to transmit a single symbol, where depending on the modulation scheme it can represent up to  $n$ -bits. The symbol rate depends on the channel characteristics, and the associated probability of error  $P_{err}$  of the chosen modulation. In a general sense, higher symbol rates enable faster communication (bits per second) but are more susceptible to noise, an example of a time domain representation of ASK modulation is given in Figure H-5.

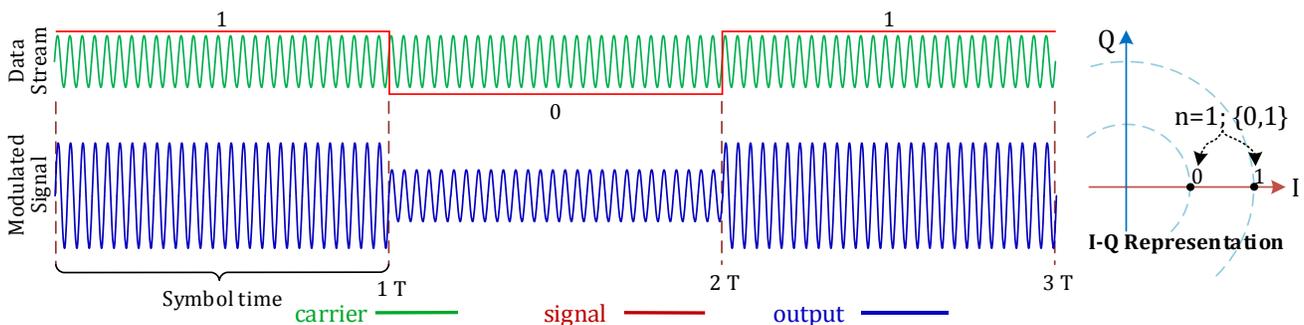


Figure H-5. The ASK time domain modulation for  $n = 1$ .

### H.1.1.3.Phase shift keying

The phase-shift keying (PSK) modulation is similar to the analog phase modulation, using discrete phase offsets to represent binary data, consequently the signal amplitude does not affect the demodulation process, decreasing the error rate with respect its ASK counterpart. However, PSK modulation suffers from limited phase states, mainly because receivers have limited phase discriminating abilities (i.e. decoding errors can occur when phase separation is small), thus PSK is often limited to encode up to  $2^4$  bits. Some examples of PSK derived modulations are Binary Phase Shift Keying (BPSK) that enables to encode 1-bit fields, or the Quadrature Phase Shift keying (QPSK) that encodes 2-bit fields by using  $90^\circ$  phase offsets.

PSK is often chosen because of its simplicity, and low  $P_{err}$ , it is ideally suited for medium speed communications, such as IEEE 802.11 (Wi-Fi), and IEEE 802.15.4 (Smart meter communications). Some PSK variants are explained in detail in the next section

#### Binary Phase Shift Keying (BPSK)

The BPSK scheme is the simplest and most robust form of PSK available, it uses a  $180^\circ$  phase difference to signal the bit level, and as such can be represented by using the  $I$  axis of the  $I - Q$  constellation. Likewise, the signal can be constructed by using the  $I$  channel of the generic  $I - Q$  modulator seen in section H.1.1.1, a time domain example of the BPSK modulation can be seen in Figure H-6. The BPSK modulation scheme can also be expressed mathematically, to represent the binary states, on Eq. H.1,  $s(t)$  can take up two values,  $S_1$  is used to represent the bit value {1}, while  $S_0$  represents the bit value {0}.

$$s(t) = \begin{cases} S_1 = A_0 \sin(2\pi F_c), & \text{when bit} = 1 \\ S_0 = -A_0 \sin(2\pi F_c), & \text{when bit} = 0 \end{cases} \quad \text{Eq. H.1}$$

where:

$A_0 = \text{Transmitted amplitude}$

$F_c = \text{Carrier frequency}$

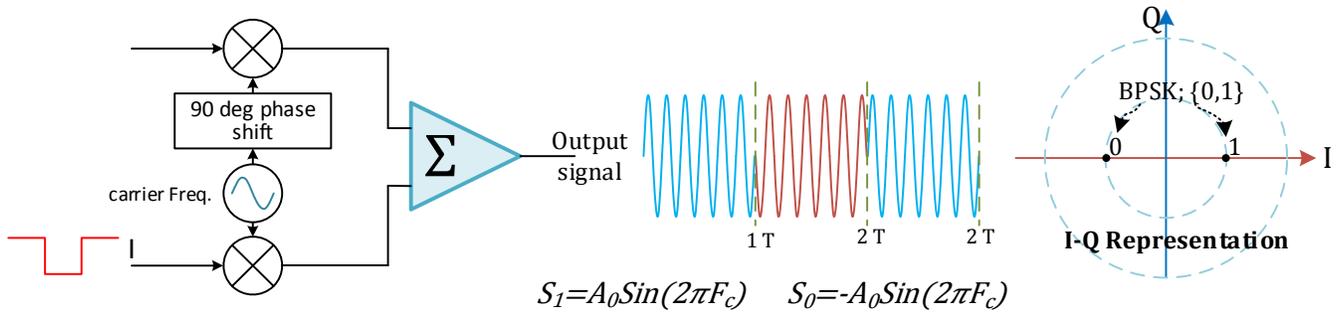


Figure H-6. The BPSK time domain modulation for the data sequence “101”.

### Quadrature Phase Shift Keying (QPSK)

QPSK enables transmission at double the data rate of BPSK modulation, while using the same bandwidth; it does this by using four phases that are offset 90° between each other. Although these phases could be aligned to each axis in the  $I - Q$  constellation, literature often shows them rotated by 45° as a result of using a 45° phase angle carrier frequency as shown on Fig [93] [188]

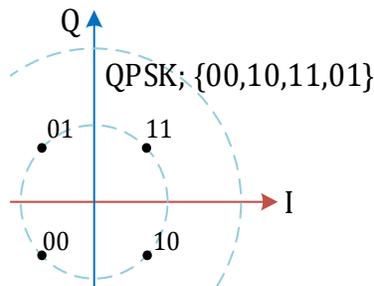


Figure H-7. The QPSK modulation expressed on the I-Q constellation [93].

QPSK can be defined mathematically by using the general high order PSK modulation formula given by Eq. H.2 [188], where  $M$  denotes the PSK order, which is 4 for the QPSK case; as an example of the general formula the Eq. H.3 shows the components for each transmitted symbol, considering the  $\frac{\pi}{M}$  rotation used on higher order PSK modulation schemes.

$$S_m = A_0 \cos \left( \frac{2\pi(m-1)}{M} + 2\pi F_c t \right) - A_0 \sin \left( \frac{2\pi(m-1)}{M} + 2\pi F_c t \right) \quad \text{for } m = 1, 2 \dots M \quad \text{Eq. H.2}$$

$$S_m = A \cos\left(\frac{2\pi(m-1)}{M}\right) \cos\left(2\pi F_c t + \frac{\pi}{M}\right) - A \sin\left(\frac{2\pi(m-1)}{M}\right) \sin\left(2\pi F_c t + \frac{\pi}{M}\right) \quad \text{for } m = 1, 2 \dots M \quad \text{Eq. H.3}$$

$$S_1 = A \cos(0) \cos\left(2\pi F_c t + \frac{\pi}{4}\right) - A \sin(0) \sin\left(2\pi F_c t + \frac{\pi}{4}\right) = -A \sin\left(2\pi F_c t + \frac{\pi}{4}\right)$$

$$S_2 = A \cos\left(\frac{1}{2}\pi\right) \cos\left(2\pi F_c t + \frac{\pi}{4}\right) - A \sin\left(\frac{1}{2}\pi\right) \sin\left(2\pi F_c t + \frac{\pi}{4}\right) = A \cos\left(2\pi F_c t + \frac{\pi}{4}\right)$$

$$S_3 = A \cos(\pi) \cos\left(2\pi F_c t + \frac{\pi}{4}\right) - A \sin(\pi) \sin\left(2\pi F_c t + \frac{\pi}{4}\right) = A \sin\left(2\pi F_c t + \frac{\pi}{4}\right)$$

$$S_4 = A \cos\left(\frac{3}{2}\pi\right) \cos\left(2\pi F_c t + \frac{\pi}{4}\right) - A \sin\left(\frac{3}{2}\pi\right) \sin\left(2\pi F_c t + \frac{\pi}{4}\right) = -A \cos\left(2\pi F_c t + \frac{\pi}{4}\right)$$

The equation shown on Eq. H.3 allows constructing an I-Q based modulator by using a simple LUT to generate a unique wave pattern depending on the particular symbol to encode, this method is based on unitary multiplications, and its time domain response can be seen in Figure H-8.

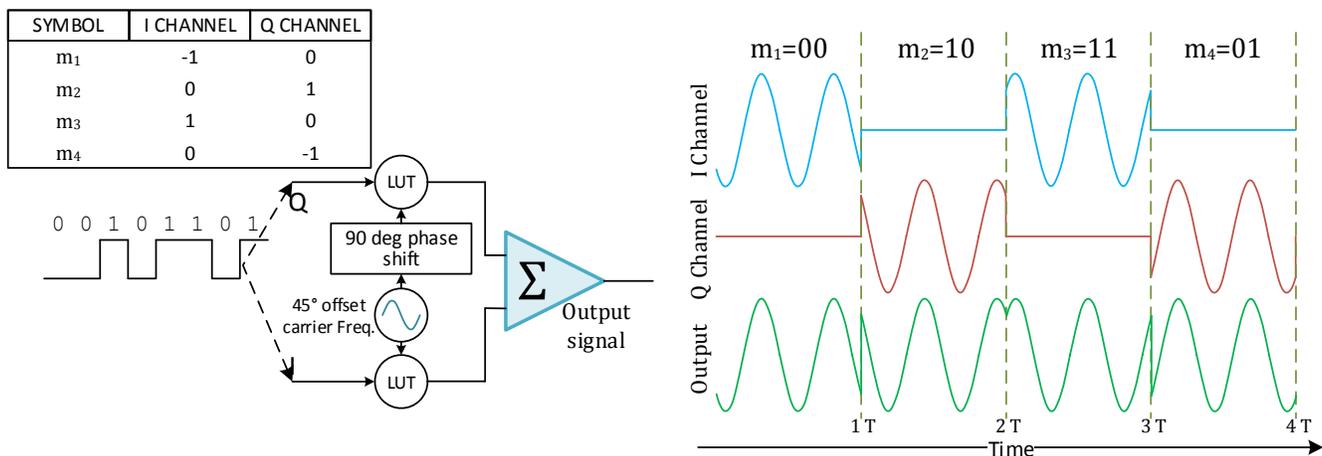


Figure H-8. The QPSK modulation on time domain based on the generic I-Q modulator.

Some drawbacks of QPSK modulation are phase jumps (signal discontinuities) that occur when transmitting symbols, this jumps can be sharp in some cases, reaching up to 180° phase shifts, when transitioning from {m<sub>1</sub>} to {m<sub>3</sub>}, since these phase jumps can't be avoided other QPSK based modulations were developed.

### Offset Quadrature Phase-Shift Keying (O-QPSK)

Offset Quadrature Phase-Shift Keying (OQPSK, O-QPSK) is designed to overcome certain overshooting characteristics of demodulators when using QPSK, caused by sudden magnitude and

phase changes between transmitted symbols. The overshoots are usually a consequence of low pass filtering on the received signal (to reduce noise), and the most common method to overcome this drawback is to use expensive linear circuits [190], which themselves encouraged the design of O-QPSK.

To explain the fundamentals of O-QPSK modulation, an alternate representation of QPSK must be first given, on Eq. H.4 a rectangular representation of the  $I - Q$  constellation shown in Figure H-2 is employed, this equation uses  $q$  to represent the "i" axis of complex planes, it requires a symbol translator in order to transmit a particular value. This symbol translator or LUT transformation, takes two bits, mapping each of them from  $\{0 \Rightarrow -1\}$  and  $\{1 \Rightarrow +1\}$ , enabling them to be used as orthogonal field multipliers. Using the same principals a generic I-Q modulator can be designed by splitting each symbol into two components and multiplying each of them and offsetting the Q component, as it can be seen in Figure H-9.

$$S_m = LUT(m[0]) \frac{\sqrt{2}}{2} A \cos\left(2\pi F_c t + \frac{\pi}{M}\right) - q LUT(m[1]) \left[ \frac{\sqrt{2}}{2} A \sin\left(2\pi F_c t + \frac{\pi}{M}\right) \right] \quad \text{Eq. H.4}$$

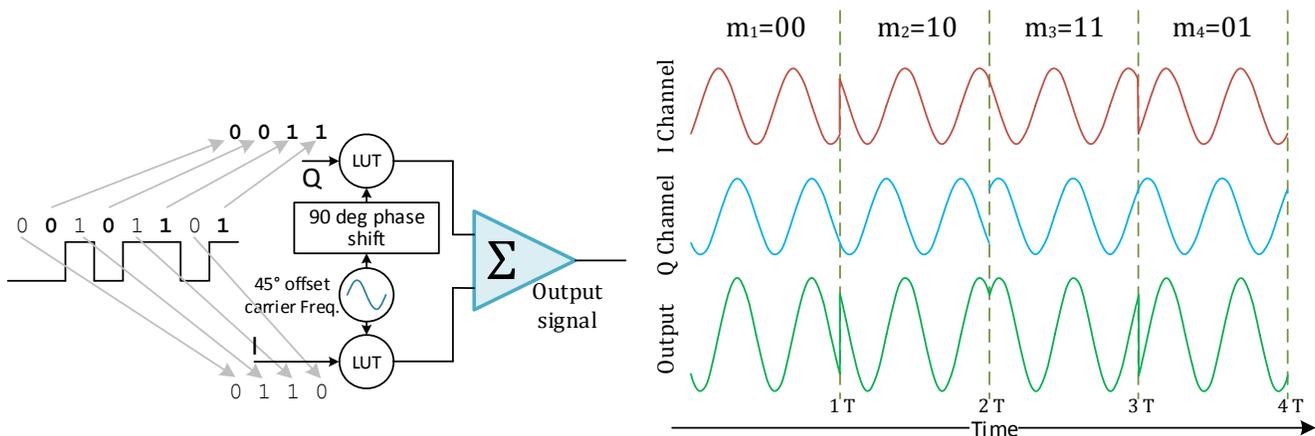


Figure H-9. Alternative QPSK modulation based on a rectangular representation.

Based on the signal modulation scheme shown in Figure H-3, the O-QPSK modulation can be implemented by offsetting the Q channel by  $1/2 T$ , this means that the transmitted signal is only valid for the upper half of the symbol period, reducing signal discontinuities (from  $180^\circ$  in QPSK to  $90^\circ$ ) that results in improved SNR and reduced hardware costs.

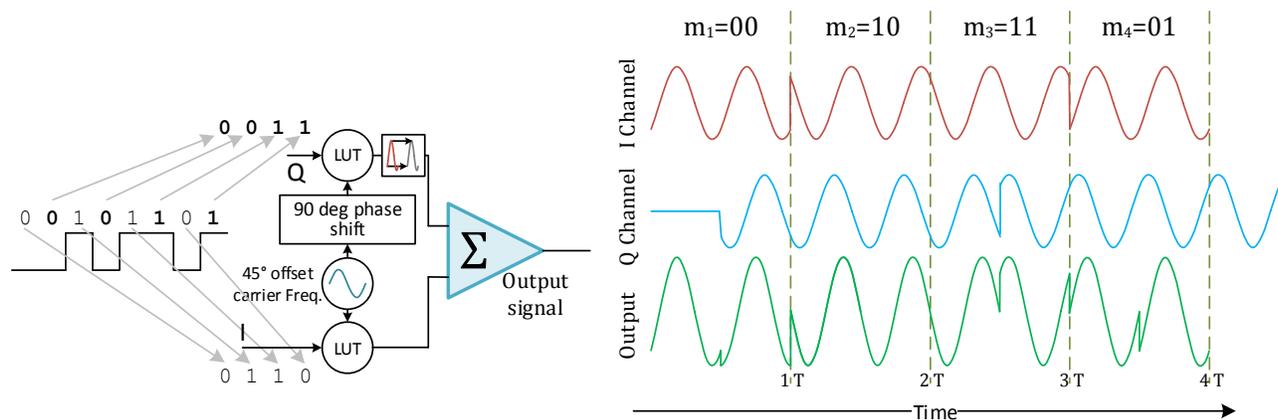


Figure H-10. O-QPSK modulation based on a rectangular representation.

#### H.1.1.4. Digital demodulation

Demodulation is the process of extracting the modulated signal from the carrier frequency, this is often done through analog matched filters, correlation methods, and in some cases thru software driven filters. Demodulators often have associated hardware that removes high frequency noise as well as Low Noise Amplifiers (LNA) to boost the received signal, and in some cases impedance matching circuitry.

Digital demodulation of wireless signals imposes an additional burden with respect its analog counterpart, since data must be decoded according to symbol periods, some form of synchronization between the receiver and transmitter must exist. These synchronization mechanisms must consider the effects of carrier frequency offsets, initial carrier phase and symbol timing, they are often discussed on depth by communication engineer's curses, but for the purpose of this work, a previously synchronized environment will be considered.

#### H.1.1.5. The correlation receiver

A traditional way of extracting the I-Q components of a modulated signal would be to use filtering techniques to extract the carrier frequency, but in certain cases, those filters can cause decoding errors due to overshoots and nonlinearities of analog filters, to solve these problems the correlation receiver was developed. The correlation receiver works by correlating the expected waveforms with the raw received signals and determining correlation indexes, these raw signals are first

amplified and low pass filtered to reduce noise and avoid aliasing effects (often correlation hardware is digital).

A simple correlation BPSK receiver is shown in Figure H-11, which uses a correlation-driven comparator to determine the received value; it does this by employing the formula described on Eq. H.5. This type of receiver is considered an optimal demodulator, having the least  $P_{err}$  of decoding an incorrect bit under noisy environments [99]. Noisy environments can drastically change the signal properties of a transmitted message, by adding delays, high frequency noise, and signal reflections, as well as reducing the signal amplitude, an example of noisy channel and its effect on the received signal can be appreciated in Figure H-12 [99].

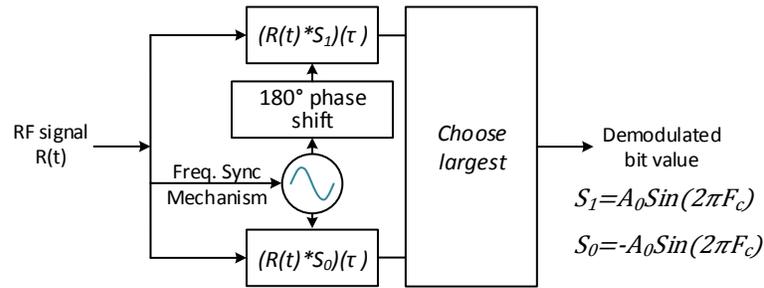


Figure H-11. Sample BPSK correlation demodulator, adapted from [99].

$$\hat{b}(n) = \text{Max}_{\{i=0,1\}} \left( \int_{nT}^{(n+1)T} R(t) * S_i(t - nT) dt \right) \tag{Eq. H.5}$$

where

$n = \text{Received symbol number}$

$T = \text{Received symbol period}$

also illustrates the time domain value of the cross correlation function, in this situation, a typical correlation receiver will pick its choice at the end of symbol period or if no symbol timing information is available, the decoded value will be picket al the midpoint between correlation crosses. In this case, the receiver has successfully decoded the modulated signal, in Figure H-13 a zoom-in version of the received signal is shown, with discrete time correlation points added.

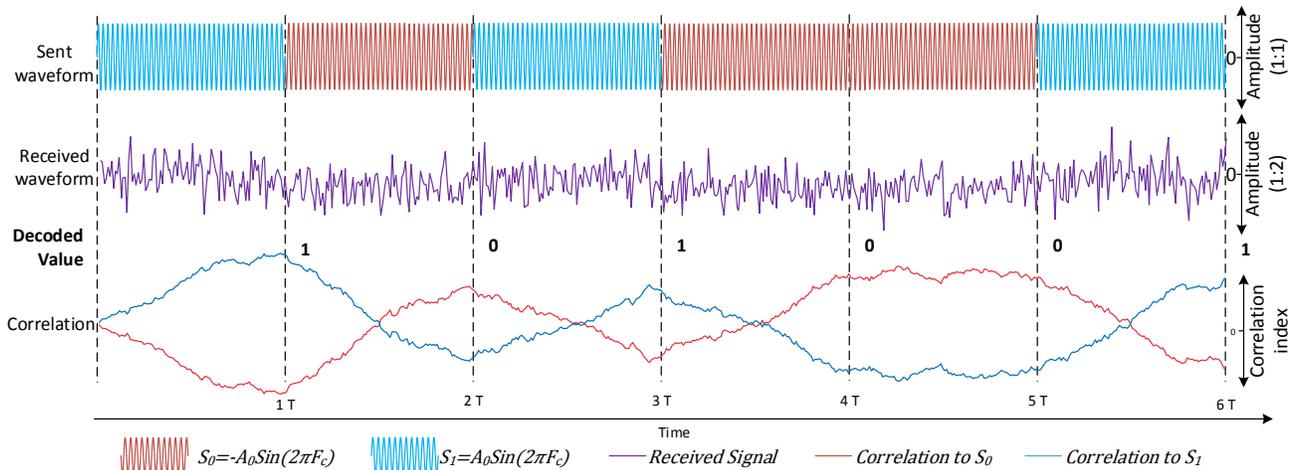


Figure H-12. Using a correlation receiver to demodulate a radio signal with value {0b101001}, adapted from [99].

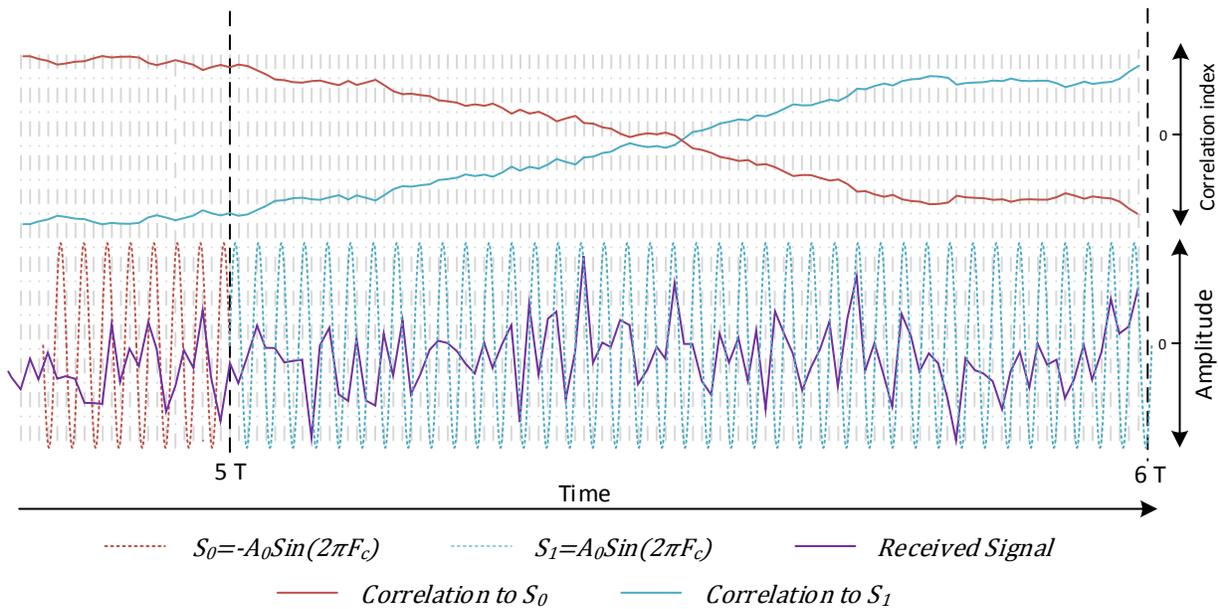


Figure H-13. Using a correlation receiver to demodulate a radio signal (Zoomed version), adapted from [99].

Correlation receivers are mostly affected by the channel attenuation, noise variability and frequency spectrum of the noise (more effects can be seen by reflections and other RF transmitters on the same channel) but most importantly by the type of modulation used, in the general sense PSK based modulations offer the best  $P_{err}$  performance.

### H.1.1.6. Direct-Sequence Spread Spectrum

As mentioned earlier some form of synchronization must exist between the transmitter and receiver, which could be difficult to implement in noisy environments, or when the transmitter is transmitting an off-nominal carrier frequency. To solve this Direct-Sequence Spread Spectrum (DSSS) creates unique patterns that enable the receiver to find the start and ending position of transmitted symbols, it does this by inserting a low autocorrelation pseudo random noise in the transmitted sequence [94].

These pseudo random noises are known as chips, and alter the previously modulated carrier phase  $n$  times during the transmitted symbol length, these chips are known as barker codes, and exist in a number of lengths (which represents the number of signal changes per transmitted symbol), being the most common 11 and 13 [94]. Some commonly used Baker codes can be viewed in Table H.1, where each digit establishes the signal amplitude multiplier at each subinterval (phase inverter), these pseudorandom chips can be identified at the receiver end by using sliding window correlations techniques to synchronize the symbol periods improving overall signal decoding.

Table H.1. Various Barker codes organized by chip length, adapted from [93]

Chip length	Codes
7	+1+1+1-1-1+1-1
11	+1+1+1-1-1-1+1-1+1-1
13	+1+1+1+1+1-1-1+1+1-1+1-1

DSSS is often implemented on communication protocols such as IEEE 802.11 and 802.15 [93], and besides synchronization, it allows improving bit-decoding properties by reducing interference susceptibilities. In Figure H-14 a generic structure of the DSSS modulation technique is shown, while in Figure H-15 a time domain example of a BPSK-DSSS modulation is given.

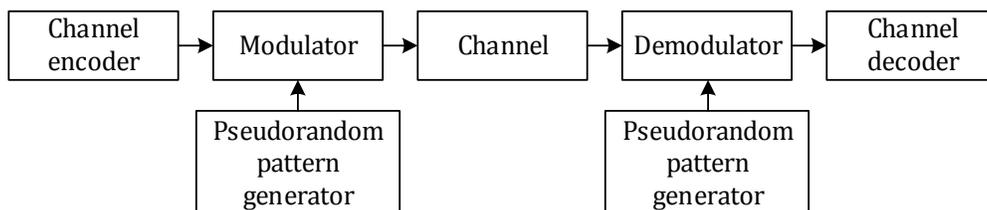


Figure H-14 The DSSS modulation scheme block diagram

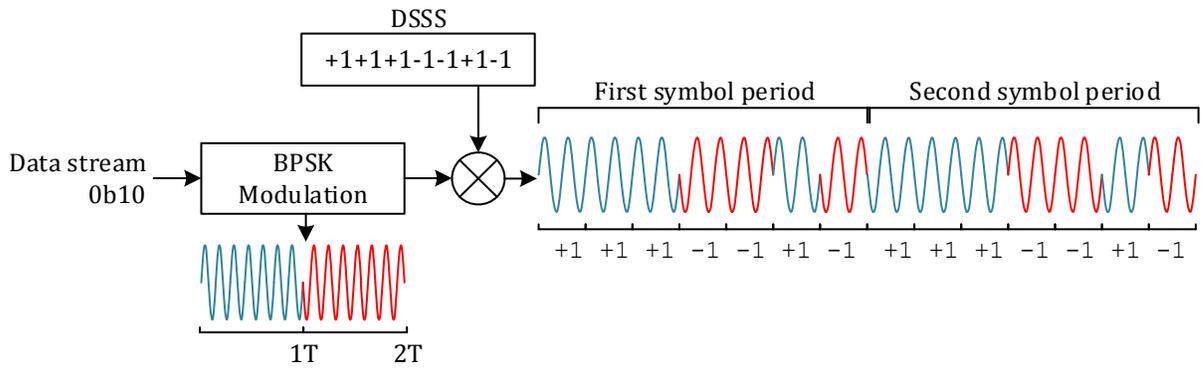


Figure H-15. Sample DSSS encoding for a BPSK modulated signal

## I. IEEE 802.15.4g Physical layer format

### I.1. The physical layer frame format

The IEEE 802.15.4g standard establishes the use of a *frame* as the physical layer data container that enables transceivers to transmit and receive information, this frame is split into three payloads (see Figure I-1). The first, known as the Synchronization Header (SHR) enables a receiver unit to locate the frame start position by transmitting a set of zeros (preamble) followed by an identifier (SFD). On the second part, a Physical Layer Header establishes the frame characteristics such as the size and rate mode; finally, on the third part the intended data payload is appended, this last field is also known as the Physical Layer Service Data Unit (PSDU).

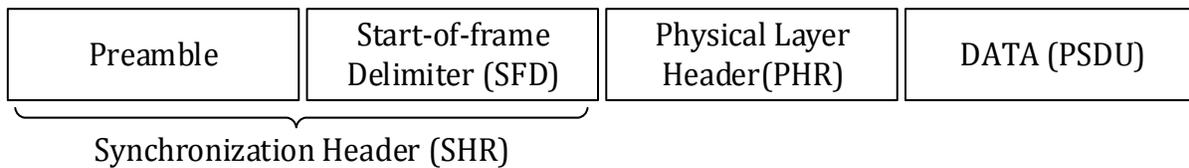


Figure I-1 The IEEE 802.15.4g frame format [98].

For the IEEE 802.15.4g the preamble is defined as being a field of 56-bits that are set to “0”, while the SFD value is defined to be {0b1110101101100010} or {0xEB62}, this distinguishes from the value of {0xA7} used on the traditional IEEE 802.15.4. For the PHR header, a definition of its composing fields is given in Table I.1.

Table I.1. IEEE 802.15.4g PHR field for the physical frame, adapted from [10]

Bit string index	0	1	2	3	4	5-15	16-23
Bit mapping	SM	RM <sub>1</sub>	RM <sub>0</sub>	R <sub>1</sub>	R <sub>0</sub>	L <sub>10</sub> -L <sub>0</sub>	H <sub>7</sub> -H <sub>0</sub>
Field name	Spreading mode	Rate Mode		Reserved		Frame Length	HCS

From Table I.1 there are a set of concepts that must be explained, the “*spreading mode*” refers to the use of fixed DSSS algorithm or a one that varies with time (*spreading mode* =1). The *Rate Mode* depends on the transmission speed, and it is set according to Table 4.4, finally HCS is a CRC based algorithm that validates this field (see details on [10]).

The IEEE 802.15.4g PHR field varies substantially from the header proposed in the core IEEE 802.15.4 standard, enabling frame lengths up to 2048 bytes vs the limited 127 bits in the core

standard, for comparison reasons Table I.2 shows the PHR field according to the IEEE 802.15.4-2011 core standard.

Table I.2. Standard IEEE 802.15.4-2011 PHR field for the physical frame, adapted from [101]

Bit string index	<b>0-6</b>	<b>7</b>
Bit mapping	L <sub>6</sub> -L <sub>0</sub>	Reserved
Field name	Frame Length	Reserved

After the frame has been created, with the intended payload data, further low level processing is done to improve noise immunity during transmission. This occurs through the insertion of error correcting codes, differentially encoded streams, insertion of known markers, and spectral improvement properties by using “chip whitening” and DSSS mapping functions. The extra low level processing is done according to the data flow diagram shown in Figure I-2, for this figure the optional processes occur according to Table 4.4, each of these steps is further explained in the following sections.

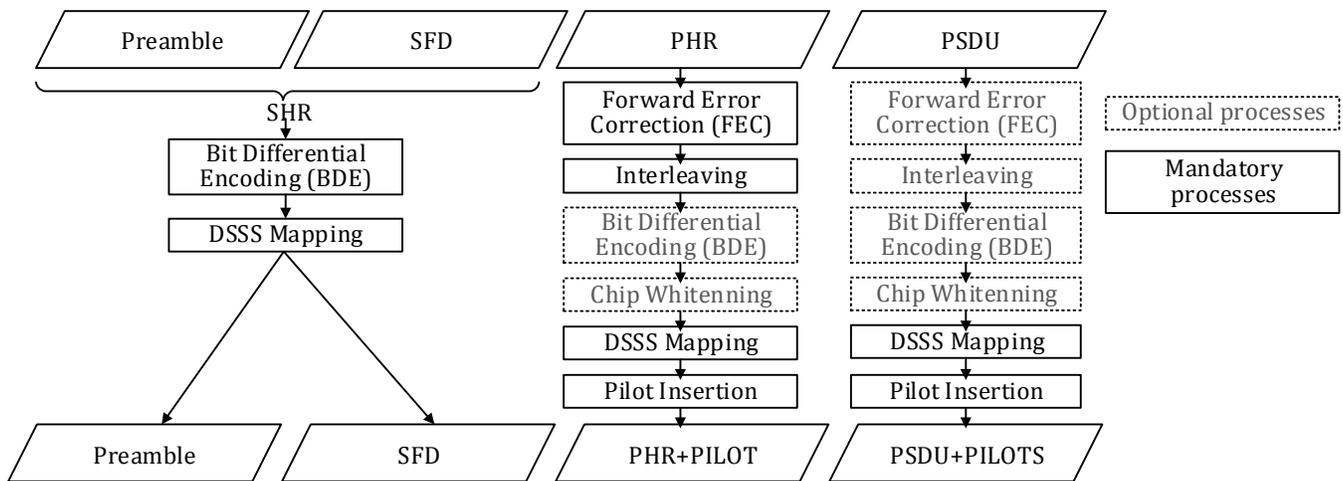


Figure I-2. The IEEE 802.15.4g frame processing.

### I.1.1. Forward Error Correction (FEC)

As mentioned earlier the physical layer has limited error-checking capabilities, and therefore requires the insertion of data redundancy bits, for the IEEE 802.15.4g standard this is done by using

the Forward Error Correction (FEC) code. The FEC redundancy is used for the PHR field, and for the PSDU, if the DSSS spreading method is constant (see Table 4.4).

The FEC code is a type of repetition code that works by creating a new data stream that contains two check bits for each original bit, each of these bits can be generated by using the pseudo-code shown in Table I.3. In order to correctly use the FEC generation algorithm the original string must be first modified by appending six zeros at the end in addition to a padding field (composed of zeros). The inserted padding should make the stream size a multiple of 7 if the encoded field is the PSDU; or 6 if the data stream contains the PHR (See example in Table I.4 for a PSDU field). According to the standard after the padding process, a set of six zeros shall be pre-appended to the stream in order to initialize the algorithm buffers; in Table I.4 the result for the proposed sample message is given.

Table I.3. The IEEE 802.15.4g FEC stream generation pseudo-code, based on the equations described on [10].

<pre> For i=6 to len J=i-6; G[j*2]=X[i]⊕ X[i-2] ⊕ X[i-3] ⊕ X[i-5] ⊕ X[i-6]; G[j*2+1]= X[i]⊕ X[i-1] ⊕ X[i-2] ⊕ X[i-3] ⊕ X[i-6]; end for </pre>	<p>Where</p> <p>X denotes the original bit stream. G denotes the generated stream.</p>
---	--

Table I.4. FEC code generation example for a PSDU field, based on the description given in [10].

Intended bit stream (size=56)
0100 0000 0000 0000 0101 0110 0101 1101 0010 1001 1111 1010 0010 1000
Six zeros appended to the end plus a padding field (size=63)
0100 0000 0000 0000 0101 0110 0101 1101 0010 1001 1111 1010 0010 1000 0000 000
Six zeros pre-appended at the string. (size=69)
0000 00 01 0000 0000 0000 0001 0101 1001 0111 0100 1010 0111 1110 1000 1010 0000 0000 0
New data stream generated by using the pseudo-code described in Table I.3. (size=126)
0011 0111 1100 1011 0000 0000 0000 0000 0011 0100 1000 1101 1011 1101 1001 1100
0010 0110 1001 1110 0111 0110 0000 1011 1010 0011 1110 1101 1110 1100 0000 00

In conclusion, the FEC code enables the receiver to reconstruct the transmitted message when sporadic noises cause demodulation errors, providing data redundancy, although the damage must occur in non-continues bits. To improve error resistance, an additional bit-remapping algorithm occurs through a process called *interleaving*.

### I.1.2. Interleaving

The IEEE 802.15.4 uses a bit remapping algorithm to improve the FEC redundancy process; this algorithm changes the bit positions of the computed FEC stream in order to increase resistance to burst errors. The interleaving algorithm works by reassigning bit positions on predetermined data windows ( $N_{INTRLV}$ ), which size depends on the field type being encoded, the window size is set to 60 bits for the PHR and 126 bits for the PSDU field. In case the encoded data size is lesser than the assigned window size, the algorithm is modified by setting  $N_{INTRLV}$  to the stream size (which must be multiple of six or seven bits according to the encoded field). On Eq. 1., the forward interleaving function is given (used on the transmitting node), while on Eq. 1.2 the reverse interleaving function is given (used on the receiving node to recover the original stream).

Continuing with the example given in Table I.4, which encodes a PSDU field of size=126; in Table I.5 the forward bit mapping function is applied to a data window of size 126 (equivalent to the PSDU buffer size), listing all of the new bit positions ( $k$ ) for each of the original bit positions( $i$ ). Finally, in Table I.6 the remapped PSDU field is given (based on the input data generated by Table I.4).

$$i = \frac{N_{INTRLV}}{\lambda} ((N_{INTRLV} - 1 - k) \bmod \lambda) + \text{floor} \left( \frac{N_{INTRLV} - 1 - k}{\lambda} \right), k = 0 \dots N_{INTRLV} \quad \text{Eq. I.1}$$

where

$N_{INTRLV}$  = Stream size after the FEC process, or buffer size

Floor = Round to lowest integer, function

mod = Modulus division, i. e. the remainder in a division

$k$  = original bit position,  $i$  = new bit position

$\lambda$  = Interleaving window size, 6 for the PHR field and 7 for the PSDU

$$k = \lambda(N_{INTRLV} - 1 - i) - (N_{INTRLV} - 1) \cdot \text{floor} \left( \frac{\lambda \cdot (N_{INTRLV} - 1 - i)}{N_{INTRLV}} \right), i = 0 \dots N_{INTRLV} \quad \text{Eq. I.2}$$

where

$N_{INTRLV}$  = Stream size after the FEC process, or buffer size

Floor = Round to lowest integer, function

$k = \text{original bit position}, i = \text{new bit position}$

$\lambda = \text{Interleaving window size}, 6 \text{ for the PHR field and } 7 \text{ for the PSDU}$

Table I.5. Generated bit positions for interleaving algorithm considering a 126-bit length PSDU field, ( $\lambda = 7$ ) [10].

$k$	$i$										
0	125	21	122	42	119	63	116	84	113	105	110
1	107	22	104	43	101	64	98	85	95	106	92
2	89	23	86	44	83	65	80	86	77	107	74
3	71	24	68	45	65	66	62	87	59	108	56
4	53	25	50	46	47	67	44	88	41	109	38
5	35	26	32	47	29	68	26	89	23	110	20
6	17	27	14	48	11	69	8	90	5	111	2
7	124	28	121	49	118	70	115	91	112	112	109
8	106	29	103	50	100	71	97	92	94	113	91
9	88	30	85	51	82	72	79	93	76	114	73
10	70	31	67	52	64	73	61	94	58	115	55
11	52	32	49	53	46	74	43	95	40	116	37
12	34	33	31	54	28	75	25	96	22	117	19
13	16	34	13	55	10	76	7	97	4	118	1
14	123	35	120	56	117	77	114	98	111	119	108
15	105	36	102	57	99	78	96	99	93	120	90
16	87	37	84	58	81	79	78	100	75	121	72
17	69	38	66	59	63	80	60	101	57	122	54
18	51	39	48	60	45	81	42	102	39	123	36
19	33	40	30	61	27	82	24	103	21	124	18
20	15	41	12	62	9	83	6	104	3	125	0

Table I.6. PSDU field interleaving example, highlighting the original vs new bit positions.

PSDU field previously processed by the FEC algorithm, described in Table I.4. (size=126)
0011 0111 1100 1011 0000 0000 0000 0000 0011 0100 1000 1101 1011 1101 1001 1100
0010 0110 1001 1110 0111 0110 0000 1011 1010 0011 1110 1101 1110 1100 0000 00
New stream generated by using the bit permutations described by the interleaving algorithm. (size=126)
0011 0011 1011 0100 0101 0110 1101 0110 0011 0111 1010 0110 0000 0000 1010 0011
1100 0001 0100 0101 0011 1000 1101 1011 1000 1000 0110 0111 0011 0100 1001 1000

\*\*The example value given on annex N of [10], appears to be wrong, see justification on annex XX

### I.1.3.Bit Differential Encoding (BDE)

The BDE algorithm optionally performs a differential encoding on the data stream, which depending on the used modulation scheme and transmitting speed can be first altered by the (FEC +interleaving) algorithm (see Table 4.4). This algorithm also creates a new data stream ( $E$ ) based on the original ( $R$ ) data stream by constantly XORing the input with the newly created stream, see Eq. I.3.

$$E_n = R_n \oplus E_{n-1} \quad \text{Eq. I.3}$$

### I.1.4.DSSS mapping

As mentioned earlier in section H.1.1.6, DSSS techniques improve overall symbol detection properties by inserting pseudo random noise sequences, for the IEEE 802.15.4g these PRNs are defined according to LUT's that are used to transform the post-processed frame according to the transmission speed and radio operating frequency (see Table 4.4). In Table I.7 the LUT table for 500 kbps transmissions is given, it maps a set of four bits into 8-chip sequences, while in Table I.8 a similar mapping is given but for 250 kbps transmission links (4-bits into 16 chips).

Table I.7. (8,4) DSSS bit to chip mapping, taken from [10]

Input bits ( $b_0 b_1 b_2 b_3$ )	Chip values ( $c_0 c_1 \dots c_7$ )
0000	00000001
1000	11010000
0100	01101000
1100	10111001
0010	11100101
1010	00110100
0110	10001100
1110	01011101
0001	10100010
1001	01110011
0101	11001011
1101	00011010
0011	01000110
1011	10010111
0111	00101111
1111	11111110

Table I.8. (16,4) DSSS bit to chip mapping, taken from [10]

Input bits ( $b_0 b_1 b_2 b_3$ )	Chip values ( $c_0 c_1 \dots c_{15}$ )
0000	00111110 00100101
1000	01001111 10001001
0100	01010011 11100010
1100	10010100 11111000
0010	00100101 00111110
1010	10001001 01001111
0110	11100010 01010011
1110	11111000 10010100
0001	01101011 01110000
1001	00011010 11011100
0101	00000110 10110111
1101	11000001 10101101
0011	01110000 01101011
1011	11011100 00011010
0111	10110111 00000110
1111	10101101 11000001

### I.1.5. Pilot insertion

Although DSSS offers symbol synchronization abilities, due to the short PRN sequences employed in the bit-to-chip mapping process, some loss of synchronism can occur. To prevent this, a sequence of known chips is inserted every certain interval, this sequence is sufficiently large ( $size = N_p$ ) to allow identification in noisy environments, this chip sequence is denominated as the pilot sequence, and is appended after the PHR field, and every  $M_p$  chips in the PSU field (see Figure I-3), the  $N_p$  and  $M_p$  interval is given in Table I.9.

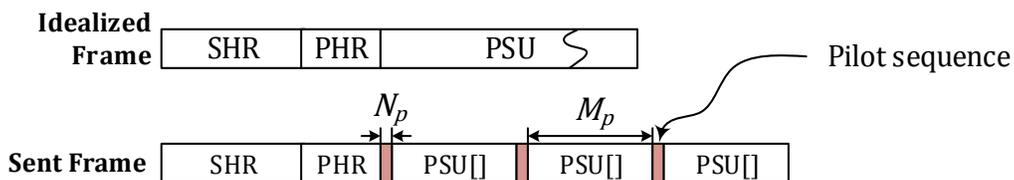


Figure I-3 Pilot insertion on a standard IEEE 802.15.4g PHY frame [99].

Table I.9. Pilot sequence insertion characteristics for 2.4GHz carrier band, adapted from [10]

Frequency Band (MHz)	Length $N_p$ (# of chips)	Spacing $M_p$ (# of chips)	Chip Sequence
2400-2483.5	128	2048	0x988B4E42526DC7A0D467D875E7DF80AB

### I.2. Clear Channel Assessment (CCA)

The physical layer uses a Clear Channel Assessment to reduce the likelihood of a collision, the sampling time is established by the IEEE 802.15.4g standard in multiples of the symbol time  $\left(\frac{1}{ChipRate}\right)$ , during the required sampling time, the signal ED levels should be below -90db.

Table I.10. CCA sampling time for various Frequency bands, taken from [10]

Frequency Band (MHz)	CCATime (# of symbols)
470-510	4
779-787	8
868-870	4
902-928	8
917-923.5	8
2400-2483.5	8

## **J. Computer Architecture**

There are many aspects of computer architecture that can be discussed about components, systems, design principles and etc., yet they are out the scope of this work, in the following sections some core aspects related to this work are discussed.

### **J.1. CPU Types**

Among computer technologies there are common hardware devices: memory, CPU, storage media, and IO devices, yet since there is a vast range of computer needs several ranges of processing power are needed, in the following subsections the main classes of CPU are discussed.

#### **J.1.1.Desktop**

Desktop CPU are mostly standardized in to the Intel x86 CPU (and subsequent releases), although other CPU architectures exist, their market presence is small. The x86 architecture is based on a Complex Instruction Set Computer (CISC) architecture, with many extension and backward compatibility options. Backward compatibility is such that current x86 architectures are actually Reduced Instruction Set Computers (RISC) that interpret CISC code [191]. The CPU's are independent of other components and are highly standardized, and although computing power in recent years has been shifted from high-speed single core, to power efficient multicore medium speed execution, power consumption is in the order of tens of watts to half a watt (Atom Architectures).

#### **J.1.2.Microcontrollers.**

When CPUs are embedded into a die with other auxiliary components, it receives the name of microcontroller. A microcontroller contains all basic computing units to enable the execution of programs and provides basic I/O communication channels, since the components are on the die and they are non-upgradeable, components are included in the die according to the price range by default. Most CPU microcontrollers are power efficient and consume very little power, although their operating speed is much lower than desktop CPU's, they mostly use RISC architectures to improve throughput. In addition, since the CPU is highly integrated with other components, various CPU components (bus handlers, caches, and interrupts) are arranged in proprietary manner, making code incompatible even among product families.

### **J.1.3. System on Chip**

Systems on a Chip (SoC) embed a microcontroller plus additional hardware to fulfill a specific task, such as cellphone functionality, since these designs are highly integrated they offer low overall cost, faster time to market, and provide basic software, all of this at the cost of flexibility. This loss of flexibility can be a problem for some companies, as well as providing generic attacks if security is breached for devices based on a particular SoC.

### **J.2. System buses**

System buses serve as the communication pathways that enable data interchange between different units, buses contain data, address and control signals, although in the beginning only a single bus existed, on modern architectures there are usually a set of parallel buses that enable data transmission between different units at the same time.

Traditionally bus control relied only on the CPU, but present hardware requirements have made essential multi-control buses, meaning other devices might initialize data transfers, this has added complexity to bus control, requiring bus arbitration schemes, semaphores and in some cases speed variable capabilities,

### **J.3. Random Access Memory (RAM)**

Random Access memory offers the possibility to store variables and retrieve them in a non-sequential manner, with the introduction of the x86 family this capability was extended to execute programs residing in RAM, although programs are initially loaded from other storage mediums.

### **J.4. Static Random Access Memory, or Read Only Memories (SRAM-ROM)**

Static Random Access Memory or Read only memory traditionally stores the program contents for microcontroller environments, program storage has evolved from single write ROMs to literally millions of writes SRAM enabled flash technology. This SRAM data containers now a days store additional calibration information, backup software, security keys, and any other data that needs to be kept persistently.

### **J.5. Von Newman Architecture**

On the Von Newman Architecture a single bus exists to transfer data from the memory space to the CPU, since in the beginning only ROM contained code, it limited the speed for fetching variables and executing code.

### **J.6. Harvard Architecture**

The Harvard architecture can be seen as an improvement over the Von Newman architecture by having dedicated buses for code and data containers.

### **J.7. Performance**

Computer performance is traditionally measured in the number of instructions that can be performed per second, usually counted in millions (MIPS), these terminology will not be used in the following sections due to possible confusion related to the MIPS core, discussed in section 6.3. Other means to measure performance are discussed in section 6.2.1.

### **J.8. Interrupts**

Interrupts serve as the multitasking components of microcontrollers that enable the CPU to switch tasks; these often handle I/O events by special subroutines created by the programmer, interrupts are often used to insert periodic events via timers or counters that alter the normal program flow. Interrupts can improve program readability, but they must be designed carefully to avoid unnecessary code overheads.



## K. The Simplex algorithm

Optimization theory and related methods deal with selecting the best alternative in the sense of the given objective function. For this particular work, linear programming optimizations are employed to solve the proposed matrix system. The goal of linear programming is to determine the values of decision variables that maximize or minimize a linear objective function, where the decision variables are subject to linear constraints. A linear programming problem is a special case of a general constrained optimization problem. In the general setting, the goal is to find a point that minimizes the objective function and at the same time satisfies the constraints.

### K.1. Linear Programs in Standard Form

Before the simplex method can be applied, a set of transformations must be executed before a linear problem can be converted into standard form (see Eq K.1)

$$\begin{array}{ll} \text{Min} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{subject to} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ & x_1 \geq 0; \dots; x_n \geq 0 \end{array}$$

Where:

The objective is maximized, the constraints are equalities and the variables are all nonnegative.

This is done as follows:

- If the problem is max  $z$ , convert it to min  $-z$ .
- If a constraint is  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$ , convert it into an equality constraint by adding a nonnegative slack variable  $s_i$ . The resulting constraint is  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + s_i = b_i$  where  $s_i \geq 0$
- If a constraint is  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$ , convert it into an equality constraint by subtracting a nonnegative surplus variable  $s_i$ . The resulting constraint is  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - s_i = b_i$ , where  $s_i \geq 0$

## K.2. Generalities for solving linear problems by using a matrix based approach

The combination of variable constraints and slack variables can also be written using a block structured matrix notation (Eq. K.1)

$$\begin{bmatrix} A & I \\ C^T & ZV \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad \text{Eq. K.1}$$

Where:

$A \in \mathcal{M}_{m \times m}(\mathbb{R})$  and contains  $m$  restricting equations with  $n$  variables (the tableau method uses a squared version)

$I \in \mathcal{M}_{m \times m}(\mathbb{R})$  and is an identity matrix that represents the slack variables

$C \in \mathbb{R}^m$  is a vector containing the minimizing function coefficients

$B \in \mathbb{R}^m$  is a vector containing the restrictions that limit equations in  $A$

$z \in \mathbb{R}^m$  is a vector containing the result during the minimizing process

$ZV \in \mathbb{R}^m$  is a vector containing zeros of size  $m$

$x \in \mathbb{R}^m$  is a vector containing the  $C^T B$  result during the process

This system defines the slack variables  $\mathbf{I}$  and  $\mathbf{z}$  as linear combinations of the variables  $\mathbf{A}$ . This system is known as dictionary for the linear problem. More specifically, it is the initial dictionary for any Linear Problem (LP). This initial dictionary defines the objective value  $z$  and the slack variables as a linear combination of the initial decision variables. The variables that are “defined” in this way are called the basic variables, while the remaining variables are called non-basic. The set of all basic variables is called the basis. A particular solution to this system is easily obtained by setting the non-basic variables equal to zero (see Eq. K.2 ) although it does not minimize the requested solution.

$$\begin{bmatrix} A & 0 \\ C^T & 0 \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad \text{Eq. K.2}$$

Every dictionary identifies a particular solution to the linear system obtained by setting the non-basic variables equal to zero. Such a solution is said to be a basic feasible solution (BFS) for the LP if its variables are non-negative, (i.e. the point lies in the feasible region for the LP)

The core idea behind the simplex algorithm is to move from one feasible dictionary representation of the system to another (and thus from one BFS to another) while simultaneously increasing the value of the objective variable  $z$ .

Each feasible dictionary is associated with one and only one feasible point. This is the associated BFS obtained by setting all of the non-basic variables equal to zero. To change the feasible point, the value of one of the non-basic variables must increase its value from its current value of zero.

The simplex algorithm is used to perform the prior mentioned steps by using an augmented matrix format, also known as the *Tableau Format*. In the next section a simplified simplex algorithm is presented in order to show the method.

### K.3. The simplex method algorithm by using the Tableau Format

The simplex algorithm can be summarized by the following steps:

Step 0. Form a tableau corresponding to a BFS. For example, if we assume that the basic variables are  $x_1, x_2, \dots, x_m$  the simplex tableau takes the initial form shown in Eq. K.3.

$x_1$	$x_2$	$\dots$	$x_m$	$x_{m+1}$	$x_{m+2}$	$\dots$	$x_j$	$\dots$	$x_{n-1}$	$x_n$	<i>RHS</i>	
$a_{1,1}$	$a_{1,2}$	$\dots$	$a_{1,m}$	$1$	$0$	$\dots$	$0$	$\dots$	$0$	$0$	$b_1$	
$a_{2,1}$	$a_{2,2}$	$\dots$	$a_{2,m}$	$0$	$1$	$\dots$	$0$	$\dots$	$0$	$0$	$b_2$	
		$\vdots$				$\vdots$		$\vdots$				
$a_{m-1,1}$	$a_{m-1,2}$	$\dots$	$a_{m-1,m}$	$0$	$0$	$\dots$	$0$	$\dots$	$1$	$0$	$b_{m-1}$	
$a_{m,1}$	$a_{m,2}$	$\dots$	$a_{m,m}$	$0$	$0$	$\dots$	$0$	$\dots$	$0$	$1$	$b_m$	
$c_1$	$c_2$	$\dots$	$c_m$	$0$	$0$	$\dots$	$0$	$\dots$	$0$	$0$	$-z$	

Eq. K.3

Where

*RHS* is the right hand side (*B* vector)

*z* is the evaluated value of  $c'x$

Step 1. If each  $c_j \geq 0$ , stop; the current basic feasible solution is optimal.

Step 2. Select  $q$  such that  $c_q < 0$  to determine which nonbasic variable is to become basic.

Step 3. Calculate the ratios  $b_i/a_{iq}$  for  $a_{iq} > 0, i = 1, 2, \dots, m$ . If no  $a_{iq} > 0$ , stop: the problem is unbounded. Otherwise, select  $p$  as the index  $i$  corresponding to the minimum ratio, (see Eq. K.4 )

$$\frac{b_q}{a_{pq}} = \min \left\{ \frac{b_i}{a_{iq}}, a_{iq} > 0 \right\} \tag{Eq. K.4}$$

Step 4. Pivot on the  $(p, q)_{th}$  element, updating all rows, including the  $z - row$ . Return to Step 1.

### K.3.1.Additions to the basic simplex algorithm

Degenerate basic feasible solutions (i.e. one of the basic variables takes a zero value) may occur during the simplex algorithm solution. In such a situation the minimum ratio  $b_i/a_{iq}$  is zero. Therefore, even though the basis changes after we pivot about the  $(p, q)_{th}$  element, the basic feasible solution cycles and degeneration occurs. To prevent this; a rule known as the Bland's rule must be used to choose an adequate q and a p, thus eliminating the cycling problem (see Eq. K.5)

Eq. K.5

$$q = \min\{i, r_i < 0\}$$
$$p = \min\left\{\frac{b_j}{a_{jq}} = \min\left\{\frac{b_i}{a_{iq}}, a_{iq} > 0\right\}\right\}$$

The average computational complexity of the simplex algorithm is often considered polynomial, although exponential worst-case solving times have been shown. To address some problems many authors propose to use  $3m$  as the loop limit to use during the algorithm described on the previous section.