FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Low-Cost Wireless Electricity Metering System with User-Friendly Interface

António Damião das Neves Rodrigues

A thesis submitted under the Integrated Master in Electrical and Computers Engineering

Supervisor: Dirk Elias (Prof. Dr.-Ing.)

Co-supervisor: Ana Aguiar (Dr.-Ing.)

February 2010

© António Damião das Neves Rodrigues, 2010

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES 2009/2010

A Dissertação intitulada

"LOW-COST WIRELESS ELECTRICITY METERING SYSTEM WITH USER-FRIENDLY INTERFACE"

foi aprovada em provas realizadas em 23/Fevereiro/2010

o júri

Presidente Professor Doutor António José de Pina Martins
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Araújo Mendes
Professor Auxiliar do Departamento de Electrónica Industrial da Escola de Engenharia da Universidade do Minho

Professor Doutor Dirk Christian Elias
Professor Catedrático Convidado do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professora Doutora Ana Cristina Costa Aguiar
Professora Auxiliar Convidada do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

AUTOR: **ANTÓNIO DAMIÃO DAS NEVES RODRIGUES**

Faculdade de Engenharia da Universidade do Porto

Abstract

Excessive energy consumption, electricity in particular, is becoming a major concern in present day society, not only due to economical but also due to environmental issues. Over the past years, regulations have been applied to manufacturers, instructing them to produce and distribute energy efficient electrical appliances, hoping to solve the problem by delivering sustainable tools to the hands of the general consumer. However, the general public is not aware on how to take full advantage of these efforts. In order to solve this issue, there is the need to create tools which allow the general public to understand their electricity consumption behaviour and become more energy efficient consumers.

The goal of this project was to specify and develop an electricity metering system, a tool capable of providing a common user with information about their electricity consumption behaviour. Although other electricity metering solutions exist, the developed system has been required to follow a low cost strategy, to be easily deployed and maintained and provide high usability.

The system's architecture comprises three main blocks - *home*, *backoffice* and *user interface* - each one with its specific components and interactions between them. The *home* and the *backoffice* are separated, with the Internet as a communication channel, following a client-server scheme, where the first is the client and the second the server side. The *home* components are the *electricity meters*, a *collection point* and a *gateway*. The *electricity meters* perform electricity consumption readings and are equipped with an RF module. The *collection point* is also equipped with an RF module, and together with the *electricity meters* form a wireless sensor network. The first periodically requests the last for electricity consumption readings, and re-directs them to the *gateway*, which in turn sends them to the *backoffice*, via Internet. The *backoffice* holds a *database* and a *web server*. The first is used for storing electricity consumption data while the second runs a *website*, used to visualize the same consumption data. The *user interface* is composed by various GUI elements, including a *website* and a digital photo frame RSS feed mechanism.

In the end, and by following the architecture presented in the previous paragraph, a proof-of-concept prototype of a electricity metering system has been achieved. The final system fulfils its base requirements, providing a modular approach for measurement of electricity consumption in buildings, offering user-friendly, platform-independent visualisation of data from any device with Internet access.

Resumo

O consumo excessivo de energia eléctrica é uma preocupação emergente na sociedade actual, não só pelos problemas causados a nível económico como também do ponto de vista ambiental. Ao longo dos últimos anos, regulamentos têm sido aplicados a fabricantes de aparelhos eléctricos favorecendo a produção e distribuição de aparelhos eficientes, tentando assim reduzir o consumo excessivo de energia eléctrica. No entanto, o público nem sempre possui a informação necessária para usar estes aparelhos de forma eficiente. Neste sentido, existe a necessidade de criar ferramentas que permitam ao público perceber o seu comportamento de consumo assim como torná-lo mais eficiente.

O objectivo deste trabalho é especificar e desenvolver um sistema de monitorização de energia eléctrica para ambientes residenciais, uma ferramenta capaz de fornecer informações relativas ao comportamento de consumo a um dado utilizador. Apesar da existência de várias soluções para monitorização energética, o sistema desenvolvido tem como requisitos apresentar um baixo custo global, pouco esforço de instalação e manutenção assim como fornecer uma boa experiência de utilização.

O sistema está dividido em três blocos principais - *casa*, *servidor remoto* e *interface de utilizador* - cada um contendo componentes específicos assim como as interacções entre os mesmos componentes. A *casa* e o *servidor remoto* estão separados, tendo a Internet como canal de comunicação, seguindo um esquema de cliente-servidor. Os componentes incluídos no bloco *casa* são os *medidores de energia*, um *ponto de colecção* e um *gateway*. Os *medidores de energia* medem o consumo de potência e/ou energia eléctrica de um electrodoméstico e estão equipados com um módulo RF. O *ponto de colecção* está também equipado com um módulo RF, e em conjunto com os *medidores de energia* formam uma rede de sensores sem fios. O *ponto de colecção* efectua pedidos periódicos de medições de potência/energia eléctrica aos *medidores de energia*, e de seguida redirecciona a informação recebida para o *gateway*, que por sua vez envia a informação para o bloco *servidor remoto*, através da Internet. No ambiente *servidor remoto* residem uma *base de dados* e um *servidor web*. O primeiro componente é usado para guardar a informação do sistema enquanto o segundo disponibiliza uma *aplicação web*, usada para visualizar a informação de consumo energético. O ambiente *interface de utilizador* é composto por vários elementos gráficos, nomeadamente a *aplicação web* e uma aplicação para molduras digitais baseada em RSS feeds.

No final, uma prova de conceito da arquitectura apresentada no parágrafo anterior foi conseguida. O sistema final atinge os seus requisitos iniciais, mostrando ser uma solução modular para um sistema de monitorização de energia eléctrica em ambientes residenciais, oferecendo usabilidade e visualização de consumo multi-plataforma a partir de qualquer dispositivo com ligação à Internet.

Ackowledgments

I would like to thank my supervisors, Dirk Elias and Ana Aguiar, whose encouragement, guidance and support enabled me to deliver all the work documented in this text. I would also like to show my gratitude to Cláudia Peixoto, designer at Fraunhofer Portugal AICOS, for her help in the graphical design of the web application. This first paragraph could not end without mentioning the help given by Luis Miguel Sampaio, an electrical engineer from whom I have learned some interesting things about power supply design.

I also offer my regards to all my colleagues at Fraunhofer Portugal AICOS and all of those who supported me in any respect during the completion of the project. Last, but definitely not least, I owe my deepest gratitude to both my parents, António and Zélia, for their encouragement and their unconditional belief in the successful accomplishment of my goals, specially during moments when my confidence was not as strong.

The Author

"If you cannot measure it, you cannot improve it."

Lord Kelvin, Scottish mathematical physicist and engineer (1824 - 1907)

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Project Description . . . 2
- 1.3 Project Objectives . . . 2
- 1.4 Document Structure . . . 3

2 State of the Art **5**
- 2.1 Related Work . . . 5
  - 2.1.1 Commercial Solutions . . . 5
  - 2.1.2 Academic Solutions . . . 14
- 2.2 Electricity Metering . . . 15
  - 2.2.1 Cirrus CS54xx . . . 15
  - 2.2.2 Teridian 71M65xx . . . 16
  - 2.2.3 Analog Devices ADE71xx/75xx . . . 17
  - 2.2.4 Microchip MCP3905/06 . . . 17
  - 2.2.5 Comparing Metering ICs . . . 18
- 2.3 Indoor Communications . . . 18
  - 2.3.1 Wireless Networks for Residential Environments . . . 19
  - 2.3.2 ZigBee . . . 21
  - 2.3.3 Z-Wave . . . 25
  - 2.3.4 Comparing ZigBee and Z-Wave . . . 29
- 2.4 Remote Communication . . . 31
  - 2.4.1 Web Services . . . 31
  - 2.4.2 Simple Object Access Protocol . . . 32
- 2.5 Summary . . . 33

3 System Specification **35**
- 3.1 General System Specification . . . 35
  - 3.1.1 Components . . . 35
  - 3.1.2 Interfaces . . . 37
- 3.2 Home Environment . . . 38
  - 3.2.1 Electricity Meter . . . 38
  - 3.2.2 Home Wireless Network . . . 39
  - 3.2.3 Gateway . . . 40
- 3.3 Backoffice Environment . . . 41
  - 3.3.1 Web Services and SOAP . . . 41
  - 3.3.2 Security in Remote Communications . . . 42
  - 3.3.3 Database . . . 42

3.4 User Interface . . . 43
3.4.1 Concept . . . 43
3.4.2 Structure Overview . . . 44
3.5 Summary . . . 45

4 System Implementation **47**
4.1 Home Environment . . . 47
4.1.1 Electricity Meter . . . 47
4.1.2 Home Wireless Network . . . 58
4.1.3 Gateway Application . . . 65
4.2 Backoffice Environment . . . 68
4.2.1 Web Services and SOAP . . . 68
4.2.2 Security in Remote Communications . . . 69
4.2.3 Database . . . 69
4.3 User Interface . . . 70
4.3.1 Website Structure . . . 70
4.3.2 Web Page Description . . . 71
4.3.3 Digital Photo Frame . . . 74
4.4 Summary . . . 76

5 Evaluation and Validation **77**
5.1 Evaluation Scenario . . . 77
5.1.1 Electricity Meter . . . 78
5.1.2 Home Wireless Network . . . 81
5.1.3 Security in Remote Communications . . . 83
5.1.4 User Interface . . . 83
5.2 Cost Evaluation . . . 88
5.3 Summary . . . 88

6 Conclusions **91**
6.1 Achievements . . . 91
6.1.1 General Achievements . . . 91
6.1.2 Specific Achievements . . . 92
6.2 Future Work . . . 93

References **95**

A Electricity Meter Schematic **103**

B Power Supply Schematics **107**

C User Interface Screenshots **111**

D Evaluation Scenario Screenshots **117**

E BuildSys 2009 Paper and Reviews **121**

List of Figures

2.1 The three basic components of the OWL wireless electricity monitor: wireless transmitter (left), energy sensor (center) and the main unit receiver, with an LCD screen (right) . . . 7
2.2 The three basic components of the Eco-Eye metering system: wireless transmitter (left), measurement sensor (centre) and the LCD display (right) . . . 8
2.3 The optical sensor of the Black & Decker power monitor being assembled on a legacy wheel-based power meter . . . 9
2.4 Example of a Z-Wave network topology (a) and its corresponding routing table (b) 28
2.5 Architecture of a Web Service . . . 32

3.1 General system architecture . . . 36
3.2 High level design of the *electricity meter* . . . 39
3.3 Typical layout of the *Home* wireless network . . . 40
3.4 Description of the communications between the *Home* and *Backoffice* sides . . . 41
3.5 Entity-relationship model for the system database . . . 43
3.6 *Website* structure overview . . . 44

4.1 Current-sense transformer interface circuit for the Teridian 71M6511 . . . 49
4.2 Capacitive power supply circuit . . . 51
4.3 Basic aspects of a SMPS type, the buck converter . . . 52
4.4 Final design of the buck converter used to supply the *electricity meter* . . . 53
4.5 Simple circuit for the I^2C interface between the *measuring* and *communication* modules . . . 55
4.6 The *electricity meter* prototype achieved by using the Teridian 71M6511 Demo Board . . . 57
4.7 State diagram that describes the *communication module* state machine . . . 59
4.8 State diagram that describes the *collection point* state machine . . . 61
4.9 Explanation of the data format changes that occur during the exchange of data between the *measuring* and *communication* modules . . . 63
4.10 Communication scheme during the normal operation of the wireless network (a) and the addition of a new node to the wireless network (b) . . . 64
4.11 Description of the serial communication section for the *gateway* application . . . 66
4.12 Activity diagram for the SOAP client section of the *gateway* application . . . 67
4.13 Description of the communications between the *Home* and *Backoffice* sides, with implementation details using NuSOAP . . . 68
4.14 Final *website* structure specification . . . 70
4.15 Two views of the *website* current consumption pages: (a) timeline chart (b) bar chart 72

4.16 Description of the communication scheme between the digital photo frame and the *Backoffice* 74
4.17 Description of the PHP script responsible for generating the images to be shown at the digital photo frame 75

5.1 Disposition of the modules during the test sessions 78
5.2 SMPS evaluation results 79
5.3 SMPS evaluation results (cont.) 80
5.4 Messages exchanged within the *Home* wireless network during the node addition tests 81
5.5 Message exchanged within the *Home* wireless network during the normal operation tests 82
5.6 Wireshark logs for the evaluation of the security in remote communications . . . 84
5.7 Screenshots of the *website*, the main graphical interface of the system: (a) bar chart (b) timeline chart 86
5.8 Different perspectives over the digital photo frame application 87

C.1 *Website*'s welcome page 113
C.2 *Website*'s current consumption page - bar chart 114
C.3 *Website*'s current consumption page - timeline chart 115

D.1 An *energy meter* (1) metering the electricity consumption of a simple lamp, equipped with a incandescent light bulb of 40 W (2) 119
D.2 A *collection point* (1) connected trough an RS-232 cable (2) to a serial port of a laptop (3) running the *gateway* application 119
D.3 3 examples of *communication modules* (in the form of slave hardware units from the Z-Wave development kit) 120

List of Tables

2.1 Electricity metering systems comparison 13
2.2 Cirrus Logic CS5463 characteristics 16
2.3 Teridian 71M6511 characteristics 17
2.4 Analog Devices ADE75xx/71xx family characteristics 18
2.5 Microchip MCP3905/06 characteristics 19
2.6 Comparison between energy metering ICs 20
2.7 Comparison between PLC transceiver ICs (prices in in Euros and U.S. Dollars) . 21
2.8 The ZigBee protocol stack 22
2.9 Examples of devices implemented over the ZigBee AMI application profile . . . 23
2.10 Raw data throughputs for ZigBee 24
2.11 Comparison between ZigBee transceiver ICs 25
2.12 Z-Wave protocol structure 26
2.13 Comparison between Z-Wave transceiver ICs 29
2.14 Comparison between ZigBee and Z-Wave 30

4.1 Comparison between shunt resistors and current-sense transformers. 48
4.2 Definition of the power supply general specification 50
4.3 Interpretation of the *communication module* LED combinations 60
4.4 Details of the commands and frames specified by the new EMA command class . 65

5.1 Cost of the main components included in the *electricity meter* design 89

Symbols and Abbreviations

A	Ampere
AC	Alternating Current
ACK	Acknowledgment
AICOS	Assistive Information and Communication Solutions
AMI	Advanced Metering Infrastructure
AODV	Ad Hoc On-Demand Distance Vector Routing
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BJT	Bipolar Junction Transistor
BPSK	Binary Phase Shift Keying
^{o}C	Celsius
CARMA	Carbon Monitoring for Action
CBA	Commercial Building Automation
CO_2	Carbon Dioxide
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSS	Cascading Style Sheets
DCOM	Distributed Component Object Model
DEEC	Departamento de Engenharia Electrotécnica e de Computadores
DSSS	Direct Sequence Spread Spectrum
EAP	Ethernet Access Point
EEPROM	Electrically Erasable Programmable Read Only Memory
EMA	Energy Metering Application
EMI	Electromagnetic Interference
ESR	Equivalent Series Resistance
E.U.	European Union
F	Farad
FEUP	Faculdade de Engenharia da Universidade do Porto
FFD	Full Function Device
FSK	Frequency Shift Keying
GOF	General Operational Framework
GPIO	General Purpose Input Output
GUI	Graphical User Interface

H	Henry
HA	Home Automation
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
Hz	Hertz
I	Electric Current
I/O	Input and Output
I^2C	Inter Intergrated Circuit
IC	Integrated Circuit
ICT	Information and Communication Technologies
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IR	Infrared
ISM	Industrial, Scientific and Medical
JAVASE	JAVA Standard Edition
JSSE	Java Secure Socket Extension
LAN	Local Area Network
LCD	Liquid Crystal Display
LCSP	Low-Profile Chip Scale Package
LED	Light Emitting Diode
LQFP	Low-Profile Quad Flat Package
MAC	Media Access Control
MCU	Microcontroller Unit
MOV	Metal Oxide Varistor
NC	Network Coordinator
OEM	Original Equipment Manufacturer
Ω	Ohm
ORDBMS	Object Relational Database Management System
OSI	Open System Interconnection Reference Model
OTS	Off the Shelf
PC	Personal Computer
PCB	Printed Circuit Board
PHHC	Personal Home and Hospital Care
PHP	Hypertext Preprocessor
PHY	Physical Layer
PLC	Power Line Communications
QPSK	Quadrature Phase Shift Keying
RF4CE	Radio Frequency for Consumer Electronics
RF	Radio Frequency
RFD	Reduced Function Device
RMS	Root Mean Square
RS-232	Recommended Standard 232
RSS	Really Simple Syndication

SCL	Standard Clock
SDA	Standard Data
SMPS	Switched-mode Power Supply
SOAP	Simple Object Access Protocol
SOC	System on Chip
SPI	Serial Peripheral Interface
SQL	Structured Query Language
SSL	Secure Sockets Layer
SSOP	Shrink Small Outline Package
TCP/IP	Transmission Control Protocol/ Internet Protocol
UART	Universal Asynchronous Receiver/ Transmitter
UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locater
U.S.	United States
USB	Universal Serial Bus
UWB	Ultra Wide Band
V	Volt
VA	Volt-Ampere
VAC	Volts of Alternating Current
VAR	Volt-Ampere Reactive
VARh	Volt-Ampere Reactive-Hour
VAh	Volt-Ampere-Hour
W	Watt
WLAN	Wireless Local Area Network
WSDL	Web Services Description Language
Wh	Watt-Hour
XML	Extensible Markup Language
bps	Bits per Second
g	Gram
n	Nano (10^{-9})
μ	Micro (10^{-6})
m	Micro (10^{-3})
k	Kilo (10^{3})
M	Mega (10^{9})

Chapter 1

Introduction

This first chapter introduces the project, specifically its motivation, general description and the establishment of its main objectives. A final section briefly summarizes the structure of this document.

1.1 Motivation

Efficient energy usage is now a major concern in society, not only on industrial environments where lower power consumption translates into lower maintenance and production costs, but also in residential complexes. Along with the financial benefits brought about by an efficient energy policy, environmental and sustainability issues are also becoming major concerns which boost the necessity for general public awareness of the problem [1]. In developed countries, residential consumption represents 30% to 40% of the total energy demand, and it is predicted to rise whithin the following years [2, 3]. In this context, efficient energy usage by the population is considered to have the potential for the greatest impact on reducing general over-consumption [4].

Recent research work identifies three ways to improve energy usage in the home: improving residential building efficiency, appliance efficiency and changing public's energy consumption behaviour [3]. The first two strategies are being applied by product manufacturers and governmental institutions in the form of energy labels and efficiency ratings, as an attempt to influence the purchase of energy efficient goods [3]. However, these strategies only provide a partial solution since the general public is not aware of how to use these products in an energy efficient way [1]. Several studies, conducted in different countries, concluded that 26% to 36% of house energy consumption is a result of resident's behavior [3], supporting the importance of this last aspect in the context of residential energy efficiency.

After this last realization, the next step is to understand how to encourage home inhabitants on adopting sustainable behaviours. Recent results show that householders install energy efficiency systems in their homes mostly "for saving money, maintaining a comfortable setting" [5] and

that they desire real-time information on their in-home resource consumption [5, 1]. However this may not be sufficient to perform behavioural changes in the residents, since they may not know the course of action to follow upon a certain information set. A more effective approach is to provide consumers with sustainable education while supported by the numbers of their own consumption information [1, 3], as a continuous process, i.e., both before, during and after the energy consuming action. This way home residents not only have access to their consumption information, but also have the possibility to understand it and decide on the most appropriate action to follow.

1.2 Project Description

The idea of this project is to develop a low cost electricity metering system, capable of displaying information in a user-friendly manner, with the potential to encourage sustainable energy usage within residential environments. It should also be easy to deploy and require low maintenance efforts. Several other systems existent in the market already perform similar functionalities lacking, however, the cost efficiency that would make it a market appealing product worthy of mass production and distribution.

Although this project is oriented at electricity consumption, it should be extendable to other types of energy metering such as gas, water and heating in order to allow integrated energy consumption management. It should also be low power consuming, since an high availability power source may not be present when monitoring other types of energy other than electricity.

1.3 Project Objectives

The main goal of the project is to specify and implement a functional electricity metering system which fulfils its base requirements: it should be low cost, present a modular architecture (extendable to other types of energy metering, ease of deployment and maintenance), low power consuming and user-friendly. In order to accomplish that goal, the following set of work objectives has been defined:

- Research of already existent electricity metering solutions, both commercial and academic, as well as general public interest in such type of products.
- Using the knowledge gathered in the initial study, the elaboration of a list of suitable technologies to be employed in the development of an electricity metering system should follow;
- Specification of the complete electricity metering system's architecture, including the necessary software and hardware components;
- Implementation/integration of the software and hardware components specified by the previously elaborated architecture;

- Comprehensive evaluation of the final solution, clearly indicating which of the objectives have, or not, been achieved. In the case of existence of non-accomplished objectives or interesting further developments, they should also be clearly mentioned in this final study.

1.4 Document Structure

This document describes the work developed at Fraunhofer Portugal AICOS for the final dissertation under the Integrated Master in Electrical and Computer Engineering, Major Telecommunications, at Faculdade de Engenharia da Universidade do Porto (FEUP).

The document is divided into six chapters. The first chapter introduces the project, specifically its motivation, general description and the establishment of its main objectives. The second chapter starts with a review of already existent systems that share some of their main objectives with this project and is followed by a survey of technologies that provide useful resources to accomplish the project main goals. The third chapter documents the system specification, starting with a definition of the general system structure and continuing with the description of each individual component. The fourth chapter describes the implementation details of the components specified in the previous chapter. The fifth chapter describes the tests that were performed in order to evaluate and validate the resulting system. The last chapter contains the final conclusions of the developed work, including a section that presents the main goals achieved by this project and another that indicates possible future developments.

Chapter 2

State of the Art

This chapter documents the research and preparation work prior to the specification and implementation phases of the project. The first part provides an analysis of commercial and academic solutions for electricity metering systems. The remaining sections present and compare technologies that provide useful resources to accomplish the requirements established on section 1.2. The technologies are grouped in function of the requirement they allow to achieve: energy measurement, communication within a residential environment and remote communication.

2.1 Related Work

The idea of designing electricity metering systems is not new. Most of the households are already equipped with metering systems designed to track the overall electric energy consumptions. However, these are not optimized to make the inhabitants understand and control their consumption, due to their location within the premises and lack of flexibility. The present day paradigm is focused on creating more energy efficient structures and appliances. However, end-users are not always aware of how to take full advantage of these efforts [1]. Thus, the creation and delivery of energy-efficient technology must be accompanied by ways of encouraging sustainable energy usage [1], which can be accomplished through electricity metering systems capable of discriminating real-time consumptions and providing advice on that matter.

This section presents several solutions for electricity metering systems, both commercial and academic, designed to give household inhabitants detailed information about their consumption patterns and allow them to understand their electric energy usage.

2.1.1 Commercial Solutions

This section provides information about commercial solutions for electricity metering systems, designed to give the users information about their consumption and allow them to understand how

is their electric energy being used. Each one of the solutions is briefly evaluated, following the parameters listed below:

- Solution identification;
- General description;
- Cost;
- Relevant aspects to the project.

2.1.1.1 OWL Wireless Electricity Monitor

General Description

OWL is a simple wireless electricity metering system, developed by 2 Save Energy Ltd [6]. OWL system uses three main physical components (see figure 2.1 [6]):

- A small, easy to use measurement sensor, positioned around the power cable of the monitored device;
- A wireless transmitter, attached to the measurement sensor;
- A main unit receiver equipped with an LCD screen, capable of displaying consumption data.

The LCD screen is capable of displaying the instantaneous power consumption (in W), the instantaneous consumption cost (in Euros, British Pounds or U.S. Dollars) and equivalent greenhouse emissions (in kg per hour). It is interactive and provides a wide set of options. For example, the user can establish a maximum value for the instantaneous consumption cost, activating an alarm that should ring if that threshold is reached. The wireless transmitter and the main unit receiver are powered by three AA alkaline batteries each. The receiver can also be powered by an AC adapter.

Cost

The OWL wireless energy monitor is available for £ 34.95 (British Pounds) [6].

Relevant Aspects

The following features of OWL were considered to be interesting for this project:

- The system is easy to use since it does not require specific knowledge from the user. It is provided in reduced sized packages with an appealing design;
- The system is flexible due to a wireless approach for data transmission;



Figure 2.1: The three basic components of the OWL wireless electricity monitor: wireless transmitter (left), energy sensor (center) and the main unit receiver, with an LCD screen (right)

- OWL immediately translates electricity consumption into monetary cost, for both power (W) and energy (Wh), e.g., € per hour or total amount of € spent since the start of measurements;
- OWL introduces an alarm function, activated if a given threshold is reached;
- The information about equivalent greenhouse gas emissions is an interesting addition, considering the increasing general public awareness of environmental and sustainability issues;
- OWL uses one display per monitored device, i.e., a LCD unit cannot display the transmission of data of more than one wireless transmitter.

2.1.1.2 Eco-Eye Real Time Electricity Monitor

General Description

The Eco-Eye real time electricity monitor, developed by Modern Moulds & Tools Ltd [7], measures and displays electricity consumption of a single or set of electrical appliances. The measurement takes place over small intervals of time. This system makes use of three fundamental pieces of hardware:

- A small, easy to use measurement sensor, installed around the power cable of the monitored appliances;
- A wireless remote transmitter, attached to the measurement sensor, for the transmission of consumption data;
- A main unit receiver, equipped with an LCD screen, capable of receiving and interpreting the wireless signal sent by transmitting module. The measured data is then processed and displayed.



Figure 2.2: The three basic components of the Eco-Eye metering system: wireless transmitter (left), measurement sensor (centre) and the LCD display (right)

These components are distributed in reduced size and portable packages, portrayed in figure 2.2 [7]. The system is able to compute and display three different types of information: power consumption (in W); consumption costs (in several currencies) based on hourly, daily, weekly, monthly or even yearly projections; equivalent CO_2 emissions, also based on different period projections.

The system uses memory to keep the history of the total power consumption and costs for the last 32 hours, days, weeks, and even months [8]. The user is able to interact with the system on the display through the use of buttons, selecting one of the several options concerning the type of information to be displayed (power consumption, CO_2 emissions or cost projections), the type of currency to be used for the cost projections, among others. It is designed for low power consumption, being powered by 4 AA type batteries with an estimated life of approximately six months. It can also be powered by AC adapters. It completes consumption readings every 4 or 30 seconds.

Cost

There are two available versions, the Eco-Eye Elite, costing £ 49.99, and the Eco-Eye Mini, costing £ 39.99 (British Pounds) [7] (the Mini version has a smaller display).

Relevant Aspects

The following aspects are relevant to this project:

- The Eco-Eye system is similar to OWL (see section 2.1.1.1): easy to use and does not require specific knowledge from the user. It is provided in reduced sized packages with an appealing design;



Figure 2.3: The optical sensor of the Black & Decker power monitor being assembled on a legacy wheel-based power meter

- The capability of translating power consumption into cost, estimating the cost for periods of time up to one year. It makes use of memory, saving energy consumption of past periods, e.g., previous hours, days, weeks or months;

- It uses one display per monitored device, i.e., an LCD unit cannot display the transmission of data of more than one wireless transmitter;

- Its power efficiency is the Eco-Eye major advantage, beating OWL in that issue.

2.1.1.3 Black & Decker Power Monitor

General Description

The Black & Decker power monitor [9] uses a different approach for electricity monitoring. An optical sensor is attached to a wheel-based mechanical power meter and then calculates the rotation frequency of the wheel (see figure 2.3 [9]). The system includes two main pieces of hardware:

- An optical sensor that measures the rotation frequency of a wheel-based power meter. It is equipped with a wireless transmitter, responsible for sending the data to a receiver (in intervals of 30 seconds);

- A single receiver that is equipped with an LCD screen and converts the received data to power (in W) or cost units. It works properly within a distance to the meter up to 18 meters [9].

This system is supposed to read the power consumption as a whole since it does it directly from the main power meter. It is also capable of monitoring individual devices, however it is not very practical for that use [9].

Cost

The Black & Decker power monitor is available for $ 99.99 (U.S. Dollars) [10].

Relevant Aspects

The most interesting aspect of the Black & Decker power monitor is its approach for electricity monitoring: an optical sensor that reads the rotation frequency of a typical wheel-based power meter.

2.1.1.4 Tendril Residential Energy Ecosystem (TREE)

General Description

The TREE, developed by Tendril Networks Inc [11], is an energy metering system based on the same principle as OWL (see section 2.1.1.1). It uses ZigBee [12] as the standard for the wireless communication between the measurement modules and the receiver units. It provides an additional way of displaying information, as it is capable of sending data to a remote location via Internet. There, a web-based application processes the data into a wide range of useful figures and graphs. This system is composed by three main physical components:

- Measurement sensors, plugged between an ordinary power outlet and the desired device. It integrates a ZigBee module for transmitting the consumption data;
- A wireless gateway, also acting as a receiver of the data sent by measurement modules, capable of sending information to a specific remote location, via Internet;
- A receiver equipped with an LCD display, similar to the one used by OWL. It shows simple information such as instantaneous power consumption (in W) and hourly costs (in multiple currencies).

One of the distinguishing features of this system is the use of a web application which opens space for advanced data processing. This information can be easily accessed through a web browser. There is also a ratio of one receiver to many transmitter units, something that cannot be done when using OWL.

Cost

TREE is still not available as a direct-to-consumer product [13]. However, TREE is expected to be soon available in the market at a cost of between $ 30 and $ 50 (U.S. Dollars) [14].

Relevant Aspects

The solution developed by Tendril has its own interesting points for this project:

- A different energy sensor concept, as it consists of a middle module between the device plug and the power outlet. It is also safe and easily applied, and demands lower implementation costs. Besides, its power is taken directly from the AC power line;
- Use of ZigBee as a standard for wireless communications;
- A receiver-transmitter ratio of 1:N;
- Capable of transferring data to a remote location via Internet, where it can be processed at a higher level and easily accessed by the final user.

2.1.1.5 Plogg

General Description

Plogg, developed by Energy Optimizers Limited [15], delivers a complete and flexible solution for energy metering systems. It uses the concept of TREE (see section 2.1.1.4), providing a metering unit, applied between the power outlet and the device plug, that is capable of transmitting consumption data to a wireless network. In addition, Plogg provides software for visualization of the consumption data on a local PC. One of the main advantages of Plogg over the previously reviewed solutions is the range of supported wireless protocols. Plogg presents several types of metering units that differ on the wireless protocol they support [15]:

- PloggBlu: Equipped with a Bluetooth[1] [16] transceiver capable of transmitting consumption data to a Bluetooth enabled device;
- PloggZgb: Just as in TREE, the metering devices carry a ZigBee transceiver, allowing them to establish a ZigBee network. The consumption information is sent to a ZigBee reception point that can be connected to PC via USB [17]. There, the consumption data can be processed and visualized;
- PloggZgb EAP: The same solution as the PloggZgb with the addition of a ZigBee to Ethernet Access Point (EAP), which establishes a bridge between a ZigBee and an Ethernet network. The ZigBee network can then be connected to the Internet, making it accessible from any geographical position where an Internet connection is available.

Cost

Plogg costs may vary according to the product version. The values presented in the below list are the cost per device (British Pounds) [15]:

- PloggBlu: £ 97.00;
- PloggZgb: £ 97.00;

[1] Further information about Bluetooth can be found on section 2.3.

- PloggZgb EAP (ZigBee to Ethernet Access Point): £ 120.00;
- PloggBlu Bluetooth USB receiver: £ 30.00;
- PloggZgb ZigBee USB receiver: £ 30.00.

Relevant Aspects

Plogg is a complete solution that combines all the characteristics of the previously reviewed energy metering systems. Regarding the scope of this project, the most interesting feature of Plogg is its support for a wide variety of wireless standards.

2.1.1.6 Google Power Meter

General Description

The Google Power Meter, currently under development by Google [18], is focused on the visualization side of electricity metering systems. It consists of an application running at a remote location that collects information from already existent metering devices, displaying the data on the personal Google homepage of the device owners. The consumption information is presented in a visually appealing way, recurring to various types of graphics and charts. This application also intends to advise its users about better energy saving policies. Google Power Meter's goal is to provide a useful visualization tool for the general consumer while establishing strategic partnerships with electricity metering device manufacturers such as The Energy Detective (T.E.D.) [19], AlertMe [20] or Itron [21].

Cost

The product is free and already available, both as an extension of electricity metering devices from several partner manufacturers [] and as the Google Power Meter API [22], as announced by Google in early March 2010 [22]. The Google Power Meter API allows developers to easily integrate their metering devices with Google Power Meter, making the access to this utility not only confined to existent partnerships between Google and selected manufactures [18].

Relevant Aspects

The Google Power Meter is specialized in the visualization rather than the actual collection of consumption data. The most interesting fact about this solution is the idea of providing active and personalized advice about energy consumption to the users of the application. Besides, the idea of connecting this service to already existent metering systems, provided by several different manufacturers, is an interesting integration challenge, made easy by the release of the Google Meter API on early March 2010.

Table 2.1: Electricity metering systems comparison

Features	OWL	Eco-Eye	TREE	Black & D.	Plogg
Sensor type	No contact	No contact	Plugged	Optical	Plugged
Wireless transmission	Yes	Yes	Yes	Yes	Yes
Transmitter-receiver ratio	1:1	1:1	N:1	1:1	N:1
Portable display	Yes	Yes	Yes	Yes	No
Internet access	No	No	Yes	No	Yes
Consumption accumulation	No	Yes	Yes	Yes	Yes
Power supply	Batteries/AC adapter	Batteries/AC adapter	AC power line	Batteries	AC power line

2.1.1.7 Solution Comparison

Some of the solutions presented in the previous sections provide electricity metering systems oriented to help end-users understand their energy consumption patterns. However, they achieve that goal using different approaches. Table 2.1 summarizes the differences between them.

OWL and Eco-Eye are very similar to each other, using the same number and type of low power consumption devices, including a contactless energy sensor, a wireless transmitter and a receiver equipped with a simple display. Besides showing electricity consumption, they translate it into cost. One of the main handicaps of these two systems is the limitation of one transmitter per receiver (see the transmitter-receiver ratio on table 2.1).

The Black & Decker power monitor introduces a new approach for the metering unit: an optical sensor reads the rotation frequency of a typical wheel-based power meter. Although this idea is significantly different from OWL or Eco-Eye, it is more curious than useful.

TREE introduces several differences, when compared to OWL and Eco-Eye. It uses a lower cost approach for its metering unit (consisting of a middle module between the device plug and the power outlet). It also employs a different wireless technology, ZigBee, as well as the capability of sending data through the Internet, allowing it to be visualized in a web page. Plogg, just like TREE, is a complete metering system, providing the metering devices, supports different wireless communication standards (such as Bluetooth and ZigBee) and it also provides Internet access. One of the disadvantages of Plogg is the price of its components, by far the most expensive of the presented solutions.

The Google Power Meter is not a complete metering system, focusing on the visualization of consumption data. It offers some interesting ideas, specifically the introduction of active and semi-personalized advices about good energy consumption policies.

2.1.2 Academic Solutions

Besides commercial solutions such as those reviewed in section 2.1.1, there is also academic work on the area of electricity metering systems, mostly consisting in multidisciplinary solutions which combine several research topics such as general hardware development [23, 24], computer communication networks [25, 24], distributed systems [25], computer applications [23, 24, 26], information system applications [26] and social and behavioural sciences [23]. The following sections briefly present some of those solutions.

2.1.2.1 ACme

ACme [23], a project developed at University of California Berkeley, involved the design, implementation and preliminary evaluation of a electricity metering system composed by three main tiers [23]:

- An electricity metering unit which sits between the electrical outlet and the device being monitored, equipped with an RF (Radio Frequency) module comprising a complete IPv6/ 6LoWPAN stack [23];
- An IPv6/6LoWPAN mesh network [23], formed by the metering units, which enhances wireless coverage in comparison with Wi-Fi [23], and edge routers, which establish a bridge between the IPv6/6LoWPAN network and other IP networks;
- Application software: a master process constantly running on a web server requests values from the electricity metering units, saves the responses on a simple database which can be accessed by a web application that shows the electricity consumption in a graphical form.

The evaluation of the system has carried out at a campus buildings, monitoring several electrical appliances over a period of time of four months [23], with interesting results [23].

2.1.2.2 PowerNet

PowerNet, developed at Sanford University [24], consists in a large-scale sensing infrastructure which informs end-users about the electricity consumption of single devices on a building. It is similar to the ACme [23] system presented above, presenting the following components: power meters, utilization modules, a storage system and a web based user interface [24].

The system used both wired and wireless power meters (the latter are based on the ACme devices [24, 23]). The measured data is then routed to PC based base stations, which store it in a central database. A web application interacts with the database to display consumption information. The major difference between PowerNet and the ACme system is the study of the correlation between usage data (CPU load, colour and brightness level on monitors, etc.) and electricity consumption of the monitored devices [24].

2.1.2.3 Energy Dashboard

The study documented in [26], conducted at University of California, San Diego, is focused on characterizing energy usage in 'mixed-use' environments rather than the development of a electricity metering infrastructure. Pre-existing commercial meters were used to collect and store electricity consumption readings over a year [26] for a final analysis, which constitutes the main contribution of the study.

In short, the study presents the gathered results and shows how usage modality affects energy use [26]. Finally, the authors present ways of reducing overall energy consumption in 'mixed-use' buildings.

2.1.2.4 Other Projects

As a final remark, the subject of ICT (Information and Communication Technologies) for energy efficiency is becoming a very interesting research topic, and is the base for several European projects, which officially started in late 2008/middle 2009. These include examples such as eDiana [27], IntUBE [28] or BeyWatch [29].

2.2 Electricity Metering

A core element of this project is the electricity metering unit. These units are traditionally implemented as electromechanical mechanisms such as those used by the electric power providers to track the electricity consumption of most of households. However, and to meet the requirements of this project, electronic metering devices are more adequate than the usual electromechanical devices, mostly because of their lower cost and better versatility [30]. These are better suited for small operation schemes such as a residential environment. An electronic meter also provides better measurement accuracy: while the measurement errors on a electromechanical device are on the order of 1 %, these can drop to an order of 0.1 % or lower when using electronics [30]. Furthermore, electronic based metering units do not consume as much power [30] which is an important feature, considering that it is not reasonable to think of a power metering device that consumes a significant amount of power.

This subchapter presents and discusses electronic based solutions for this core functionality, namely energy metering ICs (Integrated Circuits).

2.2.1 Cirrus CS54xx

The CS54xx family is a series of energy meter ICs developed by Cirrus Logic [31], specially designed for single phase power metering applications. Table 2.2 summarizes the features of two of these solutions, using the CS5463 [32] as the base model.

[2] An energy-to-pulse output consists of a signal with a frequency proportional to the energy values calculated by the internal computation engine of the chip. This output method is present in most of the energy metering ICs presented in this subchapter.

Table 2.2: Cirrus Logic CS5463 characteristics

Feature	Description
Measured variables	I_{RMS} and V_{RMS} Apparent (VAh), reactive (VARh) and real (Wh) energy
Measurement accuracy	0.1 % error on energy measurements over a dynamic range of 1000:1 (25 ºC)
Analog input variables	Differential input current Differential input voltage Voltage reference
Supported input interfaces	Shunt resistors, current-sense transformers and Rogowski coils (current) Resistive dividers and potential transformers (voltage)
Supported output interfaces	UART (Universal Asynchronous Receiver/Transmitter) for bi-directional serial communication [33] Programmable energy-to-pulse[2]
Package	24-lead SSOP Length: 8.50 mm Width: 5.60 mm Height: 2.13 mm

The CS5463 takes the current and voltage applied to the monitored electrical appliance as inputs. Since this is an integrated circuit, appropriate interfaces for current and voltage have to be employed in order not to damage the chip. The supported interfaces are mentioned in table 2.2. The output data can be accessed through a bi-directional serial interface, suitable for the communication with a microcontroller. The CS5463 also provides an *auto-configuration* feature, allowing it to read calibration data and start-up instructions from an EEPROM upon power up, eliminating the need of a separate microcontroller module.

2.2.2 Teridian 71M65xx

The 71M65xx family, developed by Teridian Semiconductor Corporation [34] is a series of SOC (System on Chip) with an integrated MCU (Microcontroller Unit) core. These are designed for single-phase metering applications and specially aimed for residential power meters. The main features of this product series are shown on table 2.3. The Teridian 71M6511 will be chosen as a base model [35].

The 71M6511 is capable of measuring a variety of electricity related parameters through its internal computing engine and then use them for more elaborate calculations to be performed by the MCU core. As an attempt to minimize external circuitry, this energy metering IC supports multiple output interfaces. This variety widens the choice for the communication module to be built in parallel with the measuring infrastructure, giving the TSC 71M6511 the advantage of versatility over the solutions provided by the Cirrus CS54xx family (see section 2.2.1). The 71M6511 is available in the form of an evaluation PCB (Printed Circuit Board) that provides working examples of energy metering units and useful debugging tools, ideal for the development phase of this project.

Table 2.3: Teridian 71M6511 characteristics

Feature	Description
Measured variables	I_{RMS} and V_{RMS} Apparent (VAh), reactive (VARh) and real (Wh) energy
Measurement accuracy	Less than 0.1 % error on energy measurements over a dynamic range of 2000:1 (25 ºC)
Analog input variables	3 sensor inputs (2 current and 1 voltage channels)
Supported input interfaces	Shunt resistors, current-sense transformers and Rogowski coils (current) Resistive dividers and potential transformers (voltage)
Supported output interfaces	UART for bi-directional serial communication [33] I^2C (Inter-Integrated Circuit) [33] Optical interface optimized for IR (Infrared) applications Integrated LCD driver Programmable energy-to-pulse
Package	64-lead LQFP Length: 12.30 mm Width: 12.30 mm Height: 1.60 mm

2.2.3 Analog Devices ADE71xx/75xx

The products from the ADE71xx and ADE75xx families, developed by Analog Devices [36], integrate energy metering and microprocessor features on a single chip. Details about these features are shown on table 2.4 [37].

The solution from Analog Devices is similar to the one provided by the Teridian 71M6511 (see section 2.2.2), taking differential values of current and voltage as inputs and offering various options for output interfaces, mostly due to the use of a 8052 compatible MCU core [37]. The ADE7569 is available in the form of an evaluation PCB that provides working examples of energy metering units and useful debugging tools, ideal for the development phase of this project.

2.2.4 Microchip MCP3905/06

The MCP3905 and MCP3906 are two energy meter ICs developed by Microchip Technology [38], especially designed for single-phase residential energy metering applications. Table 2.5 summarizes the features of these two products [38].

The solution from Microchip Technology is similar to the Cirrus CS54xx family considering the input methods, the available output interfaces and the package characteristics. There is however a major disadvantage for this model, as it only supports an energy-to-pulse output interface. This implies the use of a microcontroller to interpret the pulsed signals provided by the output pins, a pulse with a frequency proportional to the RMS real power. This output method is optimized to drive electromechanical counters, devices that are not adequate to the requirements of this project.

Table 2.4: Analog Devices ADE75xx/71xx family characteristics

Feature	Description
Measured variables	I_{RMS} and V_{RMS} Apparent (VAh), reactive (VARh) and real (Wh) energy
Measurement accuracy	Less than 0.1 % error on real energy measurements over a dynamic range of 1000:1 (25 ºC)
Analog input variables	2 differential inputs (current and voltage)
Supported input interfaces	Shunt resistors, current-sense transformers and Rogowski coils (current) Resistive dividers and potential transformers (voltage)
Supported output interfaces	A UART for bi-directional serial communication I^2C Optical interface optimized for IR (Infrared) applications Integrated LCD driver Programmable energy-to-pulse
Package	64-lead LQFP or LCSP Length: 12 mm Width: 12 mm Height: 1.60 mm

2.2.5 Comparing Metering ICs

Table 2.6 shows a comparison between the solutions presented in sections 2.2.1 to 2.2.4. The comparison is based on parameters relevant to the project requirements, such as measured variables, measurement accuracy, supported output interfaces and cost. The latter is a factor of major importance due to the low cost guidelines of the project. The prices presented in table 2.6 are presented in Euros, unitary and correspond to the same order quantity (more than 1000 units).

The Microchip MCP39xx and Cirrus Logic CS54xx families do not have integrated MCU cores, which reduces the design flexibility due to the lack of programming capabilities. In the first case, the devices are designed for driving an electromechanical counter, making it unsuitable for this project. The solution from Cirrus is the less accurate, most expensive and does not provide development tools.

The solutions from Teridian and Analog Devices are very similar and seem to gather the appropriate set of options for this case, mostly because of their wide range of output interfaces, when compared to the others. Unlike the solutions discussed in the previous paragraph, integrated MCU cores are available in both of them, providing the design flexibility brought about by programming capabilities. The two provide evaluation kits, in the form of a PCB, that include reference designs and debugging tools, very useful for a development phase of a project such as this one.

2.3 Indoor Communications

As mentioned in chapter 1, the technology used for indoor communications should rely on a wireless approach to account for extensions to metering of other types of energy. However, since the

Table 2.5: Microchip MCP3905/06 characteristics

Feature	Description
Measured variables	Average (RMS) and instantaneous real (W) power
Measurement accuracy	Less than 0.1 % error on energy measurements over a dynamic range of 1000:1 (25 ºC) (MCP3905) Less than 0.1 % error on energy measurements over a dynamic range of 500:1 (25 ºC) (MCP3906)
Analog input variables	Differential input current Differential input voltage Voltage reference
Supported input interfaces	Shunt resistors (current) Resistive dividers and potential transformers (voltage)
Supported output interfaces	Programmable energy-to-pulse
Package	24 pin SSOP Length: 8.20 mm Width: 7.80 mm Height: 1.75 mm

goal of this specific system is to monitor electrical energy consumption, another type of communications technology is worthy to be mentioned, Power Line Communications (PLC) [42].

PLC defines the use of power lines as a medium for data communications. Besides being applied on local area networks (LANs) for residential environments, it can be used for broadband Internet access, Smart Grid [43] applications and communications for public infrastructures [42]. The HomePlug Powerline Alliance [44] is a group of industry members working to promote and implement standards for PLC networks and devices, providing a significant effort for the publication of the IEEE 1901 Draft Standard for PLC, in January 2010 [45]. The advantages of adopting this technology are the use of a communication infrastructure that is already installed in residential environments (the house main electric circuit) and the close relation between the implementation of the metering and communication aspects. This functionality is achieved by interfacing an IC with PLC capabilities with a power metering IC such as those reviewed in section 2.2. Table 2.7 shows several examples of such products, including the available interfaces and approximate cost [46, 47]. Although an interesting solution for electricity metering, this project is supposed to design a backbone to other types of energy metering, such as gas or water, where the sensors are not necessarily connected to the power line, making PLC not a major advantage.

2.3.1 Wireless Networks for Residential Environments

The wireless network is destined to a residential environment, with typical maximum ranges of around 20 meters. for which the following technologies are relevant:

- Wi-Fi [16];
- Bluetooth [16];
- Ultra Wide Band (UWB) [48];

Table 2.6: Comparison between energy metering ICs

Model	Variables	Accuracy	Output	Price
CS5463	Apparent, reactive, real energy	Around 0.1% error	Serial Energy-to-pulse	1,79 € [39]
71M6511	Apparent, reactive, real energy	Less than 0.1% error	Serial Optical LCD driver Energy-to-pulse	1,93 € [40]
ADE7569	Apparent, reactive, real energy	Less than 0.1% error	Serial Optical LCD driver Energy-to-pulse	3,24 € [39]
MCP3905/06	Real power	Less than 0.1% error	Energy-to-pulse	1,29 € [41]

- EnOcean [49];
- ZigBee [16];
- Z-Wave [50].

Wi-Fi

A typical 21^{st} century household is likely to be equipped with a router or other type of gateway that supports Wi-Fi technologies. Although the design and implementation of a custom reception point would not be necessary, Wi-Fi is capable of ranges of operation that typically surpass 20 meters [16], which requires relatively large amounts of power that would not be available in a stand-alone implementation.

Bluetooth

Bluetooth is a wireless standard oriented for communication over short distances (1 to 20 meters) [16] using low cost transceiver ICs. Despite all of these advantages, Bluetooth is considered to have a power consumption pattern which is too high for sensor networks [51]. Another problem lies with its lack of *self-organization*, a feature that is present in other technologies and is a major need in this project.

Ultra Wide Band

UWB is a technology that aims at high bandwidth communications over short distances, using low levels of power consumption. This type of wireless standard is used for applications that require high data rates, such as media applications and location systems. Furthermore, UWB technology is still expensive. [48]

Table 2.7: Comparison between PLC transceiver ICs (prices in in Euros and U.S. Dollars)

Model	Interfaces	Price
Echelon PL3120/PL3150	PLC, UART, SPI [33]	8,33 € [46]
Maxim MAX2981	PLC, UART, SPI	$ 7.50 [47]
Maxim MAX2986	PLC, Ethernet, USB	$ 8.50 [47]
Maxim MAX2990	PLC, UART, SPI, I^2C	$ 8.50 [47]

EnOcean

The work developed by the German company EnOcean GmbH. [52] is definitely an interesting and innovative wireless sensor solution. EnOcean focuses on the design and production of wireless sensors to be used in metering or control applications, both in residential and industrial environments. The innovative aspect introduced by EnOcean is the concept of *energy harvesting* [49]: a wireless sensor operates without any kind of external power source or batteries, collecting power from the environment [49]. The wireless transceivers developed by EnOcean, as well as the communication protocol they employ, are optimized to run on small amounts of energy, providing reliable and secure transmission at the same time. EnOcean provides a creative solution for indoor communications, however, its core business is the provision of finished products to other system integrators and, therefore, the actual development of the technology is confined to the company. Due to the lack of development flexibility, this solution is not suitable for this project.

Because of their low power and self-organization characteristics, as well as the availability of development tools, ZigBee and Z-Wave are the networking solutions that better suit the requirements of this project. The following sections are dedicated to a brief introduction of ZigBee and Z-Wave, including a comparison of the two technologies.

2.3.2 ZigBee

ZigBee defines an 'open'[3] protocol stack (Network to Application layers of the OSI model [54]) that works on top of the IEEE 802.15.4 (Physical and Link layers of the OSI model) [16] specifications. ZigBee targets are low data rate RF applications which require reliable, low power consuming and secure networking. Examples of such applications are home control and automotive networks, remote metering systems or patient monitoring. The ZigBee Alliance [53] is the entity responsible for the specification of the ZigBee upper protocol stack. Among their main concerns are the assurance of interoperability and certification of ZigBee devices, management of the technology evolution and promotion of the ZigBee brand worldwide [53]. The full structure of the ZigBee protocol stack is shown in table 2.8 [16]. The next sections will briefly cover pertinent aspects related to the ZigBee protocol layers presented in table 2.8.

[3] ZigBee specifications are available free to the general public for non-commercial purposes. A membership to the ZigBee Alliance is required to access unpublished specifications and create products for market using the specifications [53].

Table 2.8: The ZigBee protocol stack

Layer	Description	Protocol
Application layer	ZigBee devices and their application profiles	Defined by ZigBee
General Operational Framework (GOF)	Message formatting Application security	Defined by ZigBee
Network layer	Packet routing Route management	Reactive routing protocol Simplified AODV [55]
MAC layer	Based on IEEE 802.15.4	Based on CSMA/CA 8 byte MAC address
Physical layer	Based on IEEE 802.15.4	868 MHz (EU) 916 MHz (USA) 2.4 GHz (worldwide) DSSS with QPSK or BPSK modulation

2.3.2.1 ZigBee Protocol Overview

ZigBee Device Types and Operation Modes

Before diving into the lower layers of the protocol, the different ZigBee device types and operation modes are now introduced, providing an insight into the course of operations of a ZigBee network. ZigBee defines three types of devices [55]:

- Network Coordinator (NC): Keeps track of the general network information. From all the device types, it consumes the most power and computational resources;
- Full Function Devices (FFD): Capable of performing all the functions related to 802.15.4 and upper layers. It can also act as a Network Coordinator;
- Reduced Function Devices (RFD): Provides low capability functionalities by the use of lower cost and complexity implementations.

The operation modes on ZigBee networks are specifically designed to ensure long battery time by saving the power of the non-coordinator nodes. Typically, a non-coordinator node should be in a power saving mode for most of the time, waking up for short periods in order to confirm its presence. The power saving concerns cannot be taken too far though. While the exclusive use of RFDs would guarantee lower power consumption, it would exclude the capability for the network to self organize, which is only possible through the use of FFDs. This last point is of major importance as it allows the network to adapt to different environments. Two operation modes are foreseen for ZigBee networks [56]:

- Beacon mode: Used for power consumption control in extended networks. In this case all the nodes know when to exchange data between each other. This mode is indicated for the use of isolated battery operated NCs;
- Non-Beacon mode: In this case, the NC is constantly on waiting for a transmission from the other nodes, which are in deep sleep mode for most of the time. Due to the nature

Table 2.9: Examples of devices implemented over the ZigBee AMI application profile

Device	Description
Energy services portal	Communication between the energy services provider and an energy meter located at a house. It works as a gateway between a ZigBee network and the Internet.
Metering devices	A simple device that measures energy consumption being capable of sending the data to a ZigBee network.
Home metering displays	A display that is capable of receiving energy data via a ZigBee network.

of operation of the NC, power sources based on batteries do not provide the best of the solutions.

General Operation Framework and Application Layer

The General Operation Framework (GOF) consists in a connection between the application layer and the remaining protocol stack. It handles several aspects common to all ZigBee devices such as addressing modes and device descriptions. It also specifies methods, events and data formats used by applications to build commands and their responses.

Applications assigned to ZigBee devices are organized into profiles defined by the ZigBee Alliance [53]. The ZigBee application profiles provide a clear set of processing actions, assuring the interoperability between different manufacturers and a simple implementation of a wide range of application domains [53]. Several ZigBee application profiles are available (or ready to be available) such as Advanced Metering Infrastructure (AMI), Commercial Building Automation (CBA), Home Automation (HA) or Personal Home and Hospital Care (PHHC) [53]. The AMI solution is of particular interest since it is specifically designed for energy management applications. A list of device types that can be implemented using the AMI profile is shown in table 2.9.

Network Layer

The network layer of ZigBee deals with the following aspects:

- Security;
- Routing;
- Starting of new networks and inclusion of new devices;
- Associations and dissociations of ZigBee devices (carried out by the NC).

ZigBee routing is based on a simplified version of AODV (Ad Hoc On-Demand Distance Vector Routing) [55], a reactive routing protocol which only calculates a new route when requested by a source node [57]. ZigBee networks support several types of topologies such as star, cluster tree and mesh [50]. The latter allows the network to self-organize, representing one of the main features of ZigBee. Sizes can reach up to 264 nodes when using this complex type of topology [56].

Table 2.10: Raw data throughputs for ZigBee

Frequency	Modulation	Throughput
2.4 GHz (16 channels)	DSSS/QPSK	250 kbps
916 MHz (10 channels)	DSSS/BPSK	40 kbps
868 MHz (1 channel)	DSSS/BPSK	20 kbps

An appropriate scheme to be used on this project would consist in several nodes spread around the different rooms of a house. In that case, one ZigBee router node (FFD) could be used in each room as a reception point for other nodes (RFD). The FFD would then communicate with a central positioned NC, resulting in a mixture between a star and a mesh topology.

Medium Access Control (MAC) and Physical (PHY) Layers

ZigBee's PHY and MAC layers are defined and maintained by the IEEE 802.15.4. Its supported operation frequencies are 868 MHz (EU), 916 MHz (USA) and 2.4 GHz (worldwide), working in unlicensed industrial, scientific and medical (ISM) bands. It uses Direct Sequence Spread Spectrum (DSSS) when operating at 2.4 GHz (just like IEEE 802.11 [58]) with Quadrature Phase Shift Keying (QPSK) modulation. DSSS is also used with the 916 MHz and 868 MHz bands, but in both cases, Binary Phase Shift Keying (BPSK) modulation is applied [16]. Table 2.10 [56] shows the achieved data transfer throughputs, depending on the operation frequency and number of channels.

2.3.2.2 ZigBee Transceivers

After a brief description of the ZigBee protocol, the discussion will now address hardware issues related to the implementation of a ZigBee enabled device. In a similar way to section 2.2, several ZigBee transceiver ICs are presented and compared according to such parameters as output interfaces, power consumption or cost.

Since ZigBee works on top of IEEE 802.15.4 and is an 'open' standard, there is a wide variety of market available solutions for the implementation of a communicating device, as well as instructive documentation [59]. A collection of such solutions, as well as their characteristics, is shown on table 2.11. The prices are presented in Euros, unitary and correspond to an equivalent order quantity of more than 1000 units.

The models presented in table 2.11 interact with the energy metering IC through several interfaces[4] (UART, SPI or I^2C) requiring additional circuitry for the installation of an antenna. As stated above, all of the models are compliant with the IEEE 802.15.4 specifications and provide similar RF features. The most significant differences between them are related to the package and price. The larger variety of interfaces offered by the Freescale MC1321x family products is a result of the integration of a MCU core on the chip. The model from Microchip appears to have a slight advantage when concerning its cost and reasonable package type and size.

[4]Refer to section 2.2 for more details about MCU communication interfaces.

Table 2.11: Comparison between ZigBee transceiver ICs

Model	Interfaces	Power consumption	Package	Price
Texas Instruments CC2520RHDT [60]	ZigBee SPI GPIO pins	18.5 mA[1] 33.6 mA @ $P_t = 5$ dBm[2] <1 μA[3]	28 pin QFN Length: 5 mm Width: 5 mm	2,94 € [40]
Atmel AT86RF230-ZU [61]	ZigBee SPI GPIO pins	16 mA[1] 17 mA @ $P_t = 3$ dBm[2] 0.1 μA[3]	32 pin QFN Length: 5 mm Width: 5 mm	3,19 € [41]
Freescale MC1321x [62]	ZigBee SPI I^2C GPIO pins	-	71 pin LGA Length: 9 mm Width: 9 mm	3,45 € [40]
Microchip MRF24J40 [63]	ZigBee SPI GPIO pins	18 mA[1] 22 mA @ $P_t = 0$ dBm[2] 0.2 μA[3]	40 pin QFN Length: 6 mm Width: 6 mm	2,35 € [41]

[1] Receive mode.
[2] Transmit mode at maximum RF transmit power (P_t).
[3] Sleep mode.

2.3.3 Z-Wave

Z-Wave is a protocol designed for wireless sensor networks, specifically for residential control and automation. It is designed to communicate short blocks of information between several nodes, in a reliable manner. Unlike ZigBee (see section 2.3.2), which is an 'open' standard, the Z-Wave technology is proprietary of the Danish company Zensys [64]. Z-Wave is supervised by the Z-Wave Alliance [65], a group of leading manufacturers that agreed on building wireless residential products based on this technology. Among other things, the alliance assures the interoperability between Z-Wave certified devices, regardless of the product manufacturer [65].

2.3.3.1 Z-Wave Protocol Overview

Protocol Structure

As mentioned above, Z-Wave is a simple, reliable and low power consuming wireless protocol designed for residential environment applications. It is organized in a stacked structure, consisting of four layers, as shown on table 2.12 [66].

The remaining part of this section briefly introduces the different layers identified in table 2.12. The description starts with the different Z-Wave device types and finishes with an overview of its PHY layer.

Z-Wave Device Types

The Z-Wave protocol defines two basic kinds of devices, Controllers and Slaves. Controllers have the capability to initiate transmission and commands to other network nodes. Slaves just act as endpoints, executing commands from Controllers, without discrimination.

Table 2.12: Z-Wave protocol structure

Layer	Description
Application layer	Responsible for decoding and executing commands in a Z-Wave network Its specification can be altered according to the implementation characteristics
Routing layer	Controls the routing of frames between nodes Responsible for sending a frame with a correct repeater list and ensure its repetition Based on source routing (limited-size routing tables)
Transfer layer	Controls the data exchange between two nodes, including retransmission, checksum verification and acknowledgements
MAC layer	Controls the RF medium access Based on CSMA/CA [55]
Physical layer	868 and 916 MHz frequency bands FSK modulation with Manchester channel encoding 40 kbps data rate

Controllers have full routing tables which allows them to communicate with all the nodes included in a Z-Wave network. Controllers can be classified according to their existence time on a network and their mobility. A primary Controller is the first node on a Z-Wave network. There can be only one primary Controller per network and only this node type can include or exclude other nodes. Secondary Controllers are included in a Z-Wave network by a primary device and do not own inclusion or exclusion privileges. When it comes to mobility a Controller can be classified as Portable or Static [66]. Portable Controllers are capable of determining their position within the network by inquiring the surrounding nodes. They are also able to include or exclude devices in the network. A Static Controller, unlike the portable version, keeps a fixed position within the network. It is available for communication with other nodes at any time. It can also perform the same functionalities as a Portable Controller, such as storing network configurations at different times. Static Controllers have another important functionality: bridging. This unit type can work as an interface between Z-Wave and other network environments like TCP/IP or X10 devices [67]. From the point of view of the Z-Wave network, the devices belonging to the other environment would be seen as Z-Wave nodes, and vice-versa.

Slave devices are much simpler. They do not take the initiative of communicating with other nodes, they always respond to a request made by a Controller. Slaves can be used by Controllers as a mean of routing by re-transmitting messages, acquiring the designation of Routing Slaves.

Application Layer

The Z-Wave application layer handles the decoding and execution of network related commands [66]. Applications in Z-Wave are carried out by the exchange of different commands and are oriented to the device type. A different set of libraries is available for each type. These libraries relate to several aspects of Z-Wave such as protocol administration, power management or Z-Wave chip

interfaces. The two most important elements on a Z-Wave application layer frame are the Application Command Class and the Application Command. The first specifies the type of the second. Application Command Classes can be general or specific to certain types of applications such as lighting or sensor control. There are three basic command types that serve as base for more elaborate Application Commands [66]:

- SET: Used to modify data on a given node;
- GET: Used to obtain data from a given node;
- REPORT: An answer to an inquiry of the type SET or GET.

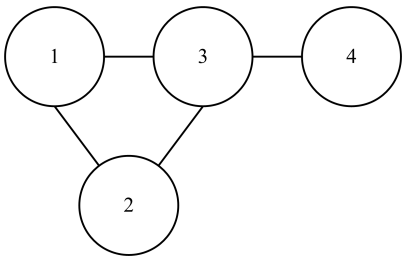
The application layer is responsible for the assignment of network identifiers. Each Z-Wave network has a unique 32 bit identifier called Home ID. Controllers have a network ID. Slaves get the Home ID from the controller as they join the network. Each one of the individual nodes is addressed using a Node ID, unique within each network, made of 8 bits. The latter is assigned by a Controller. A special type of frame is used for exchanging node information: Node Information Frames. These enclose information such as the node type or the ability to frame repetition.

Routing Layer

The routing layer in Z-Wave controls the routing of frames between the different nodes on a network. The layer responsibilities include sending a repetition list for a specific frame and the assurance of its repetition from node to node. Other functions include the management of routing tables and network topology scanning [66]. A routing layer frame can take one of two forms, used when routing functionalities are necessary [66]:

- Routing Singlecast Frame: It consists of a frame destined to one node only. This type of frame advances through the network by repetition from one router node to another, until the final destination node is reached.
- Routing Acknowledge Frame: It works as a complement to the previously mentioned type. It is used by the receiver nodes to inform the network controller that a Routing Singlecast Frame reached its destination.

Using the information obtained by the exchange of routing frames, a Z-Wave primary Controller builds and maintains a routing table. Besides, every time a new node is added to the network, the primary Controller inquires it about its neighbour nodes, allowing it to keep track of the network topology, and therefore build a more reliable routing table. An example of a routing table and its equivalent network topology is given in figure 2.4 [66]. The ‘1’ in the table indicates a direct connection between two nodes (for example, there is a direct connection between nodes 1 and 2, as the table intersection between 1 and 2 is marked with a ‘1’). The ‘0’ means the corresponding association of nodes is not directly connected.



(a) Network topology

Nodes	1	2	3	4
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1
4	0	0	1	0

(b) Routing table

Figure 2.4: Example of a Z-Wave network topology (a) and its corresponding routing table (b)

Transfer Layer

The transfer layer in Z-Wave controls data transmissions between two nodes. Its typical tasks are acknowledgment sending, data retransmission and checksum verification [66]. Although transfer layer frames can be differentiated into four types, the same basic structure is shared between them.

There are four basic formats of transfer layer frames: Singlecast, Acknowledgements, Multicast and Broadcast. Singlecast frames are intended for one destination node only. This type of frame is indicated for reliable communication schemes since it is followed by an Acknowledgement frame. If a Singlecast or Acknowledgement frame is lost or corrupted, the initial Singlecast frame is retransmitted. The Multicast and Broadcast types do not support acknowledgement, and therefore are not indicated for reliable communication.

Medium Access Control (MAC) Layer

The Z-Waves MAC layer defines the way data streams are exchanged in the RF medium. Z-Wave RF data is transmitted in blocks of 8 bits, MSB to LSB. A Z-Wave data stream is composed by a preamble block, a start of frame (SOF) block, the upper layer payload blocks and a end of frame (EOF) block, all 8-bit long. It also employs a collision avoidance scheme based on CSMA/CA.

Physical (PHY) Layer

Z-Wave operation frequency are 868 MHz (EU) or 916 MHz (USA), working in unlicensed industrial, scientific and medical (ISM) bands. Z-Wave uses Frequency Shift Keying (FSK) modulation with Manchester channel encoding [66]. The current version throughput is 40 kbps a rather low value when compared to other sets of wireless standards such as Bluetooth or Wi-Fi. The latter is an understandable fact however, since the protocol was purposely designed for low data rate applications. Originally, Z-Wave devices provided a much lower throughput, 9.6 kbps. Measures have been taken in order to allow the coexistence of the earliest and the current version, being the latter backwards compatible with the first.

Table 2.13: Comparison between Z-Wave transceiver ICs

Model	Interfaces	Power consumption	Package	Price
Zensys ZW0102	Z-Wave UART SPI GPIO pins	21 mA[1] 35 mA @ $P_t = 5$ dBm[2] 30 μA[3]	52 pin TQFP Length: 10 mm Width: 10 mm	3,46 € [39]
Zensys ZW0201	Z-Wave UART SPI GPIO pins	23 mA[1] 36 mA @ $P_t = 0$ dBm[2] 2.5 μA[3]	32 pin QFN Length: 5 mm Width: 5 mm	-
Zensys ZM3102N	Z-Wave UART SPI GPIO pins	23 mA[1] 36 mA @ $P_t = 0$ dBm[2] 2.5 μA[3]	18 notch PCB Length: 13.6 mm Width: 12.5 mm	5,00 € [39]

[1] Receive mode.
[2] Transmit mode at maximum RF transmit power (P_t).
[3] Sleep mode.

2.3.3.2 Z-Wave Transceivers

Following the same logic as the presentation of ZigBee, after a brief description of the Z-Wave protocol, the discussion will now address hardware issues related to the implementation of a Z-Wave enabled device. In a similar way to section 2.2, several Z-Wave transceiver ICs are presented and compared according to such parameters as output interfaces, power consumption or price.

Unlike ZigBee, Z-Wave works over a proprietary specification, resulting in a narrower variety of market available solutions. As in the previous case, they consist of ICs that act as Z-Wave transceivers. Some of these solutions are presented on table 2.13. The prices presented in table 2.13 are unitary, presented in Euros, corresponding to the same order quantity of more than 1000 units.

The solutions presented on table 2.13 are different from each other. The Zensys ZW0102 is a simple Z-Wave single chip transceiver, including an integrated 8-bit 8051 MCU core [68], eliminating the need for additional major components. Its power consumption characteristics are affected by this fact. The Zensys ZW0201 is an improved version of the ZW0102, including an optimized version of the 8051 MCU core and presenting a much smaller size. The ZM3102N, also developed by Zensys, is a small PCB that joins the ZW0301 Z-Wave single chip (similar to the ZW0201), a system crystal and RF front-end circuitry. This solution attempts to reduce the design complexity of such RF applications presenting itself as a good option for this project.

2.3.4 Comparing ZigBee and Z-Wave

After presenting brief overviews of the two technologies under discussion, this section explores the differences between them in order to perform a suitable comparison. Table 2.14 [50] provides a summary and simple comparison between ZigBee and Z-Wave.

Table 2.14: Comparison between ZigBee and Z-Wave

Characteristics	**ZigBee**	**Z-Wave**
Frequencies	916 MHz (USA) 868 MHz (EU) 2.4 GHz (USA/EU)	916 MHz (USA) 868 MHz (EU)
Maximum data rates	20 kbps (868 MHz) 40 kbps (916 MHz) 250 kbps (2.4 GHz)	40 kbps (916/868 MHz)
Nodes per network	Up to 2^{16}	Up to 232
Topologies	Mesh, star, tree	Mesh
Device types	3	2
Max. num. of message hops	32	4
Routing	Tree or mesh routing	Source routing
Intended applications	Large commercial/industrial networks Home automation	Home automation

Z-Wave supports both 868 MHz and 916 MHz frequency bands, and achieves raw data rates of 40 kbps while operating at both of them. Since ZigBee's PHY and MAC layers are based on the IEEE 802.15.4 standard, it also supports 2.4 GHz, achieving 250 kbps under this condition. The addressing specification defined by the Z-Wave protocol stack only allows networks as large as 232 nodes. Z-Wave uses a simple routing scheme based on routing tables (source routing) with a maximum size of the routing table to 232×232 and a maximum number of 4 message hops (which limits indoor transmission ranges to a maximum of 100 meters [55]). ZigBee employs more complex mesh routing schemes (simplified AODV [57, 55]) while also supporting more simple tree-based mechanisms. Technically, ZigBee presents several advantages over Z-Wave, which is related to the intended application scope of both technologies: while Z-Wave is clearly focused on simple residential applications, ZigBee attempts to cover both residential and large commercial/industrial purposes. However, the main difference between the two technologies is not so related with technical aspects, as explained in the next paragraphs.

Z-Wave is proprietary, meaning the technology is only supplied and maintained by the company that owns its rights, Zensys [64], and OEMs (Original Equipment Manufacturer) interested in developing Z-Wave products must acquire the technology from Zensys. OEMs depend on Zensys for maintenance and development efforts of the protocol. On the other hand, a single source for Z-Wave specification and hardware helped Z-Wave products from different OEMs to achieve high interoperability [69]. In fact, being a proprietary technology did not stop the Z-Wave Alliance from growing (including partnership with Intel [64]) and achieve a top position in the Home Automation market, with more than 300 products from leading consumer brands [70].

ZigBee is an open standard regulated by the ZigBee Alliance, and OEMs are allowed to follow the ZigBee specification and implement its own version of the protocol stack (eg.: Texas Instruments [60], Microchip [63], Freescale [62] and Control4 [69]). However, the final objective of the OEM is to profit from their own ZigBee implementation, so they usually end up 'protecting' their

own version [69]. Among other things, this results in poor interoperability between ZigBee devices produced by different OEMs, as admitted by several members of the ZigBee Alliance [69]. However, some steps have been taken by members of the ZigBee Alliance to define standards for certain application profiles [69] and explore new markets, such as the standardized specification for radio frequency-based remote controls ZigBee RF4CE (Radio Frequency for Consumer Electronics) [71].

2.4 Remote Communication

As specified by the project objectives (see section 1.3), the information collected at the residential environment should be sent to a remote location. This way the system can be monitored from virtually everywhere and re-allocates data processing resources to the remote location. The system requires a technology designed for the exchange of messages in distributed environments such as the Internet. Several solutions implement the latter such as TCP/IP sockets, endpoints of communication schemes made across IP networks [72]. Besides being able to act as receivers or transmitters in such type of networks, TCP/IP sockets implement a mechanism for delivering the received data to application processes [72]. Other mechanisms include CORBA (Common Object Requesting Broker Architecture) [73], DCOM (Distributed Component Object Model) [73] or Web Services [74]. The latter consists of web applications sets accessible using standard Internet protocols [75], a concept that matches the established requirements, and has been successfully employed in similar projects [25]. This subchapter provides a brief description of Web Services, particularly the protocol used for exchanging information over the Internet.

2.4.1 Web Services

A Web Service basically consists in a set of programmable application functions that can be accessed through the Internet. Instead of a whole application, a Web Service can be viewed as a building block of a web application [76]. The architecture of a Web Service is generally divided into three entities [74]:

- Service provider, responsible for creating, registering and publishing the existence of a specific Web Service;
- Service broker, the entity that maintains the registry of a several published Web Services;
- Service requester, the element that interacts with the two previously mentioned entities by searching the desired Web Service on a service broker and then bind their applications to the respective service providers.

The functional core of Web Services is formed by three technologies, Web Services Description Language (WSDL), Universal Description Discovery & Integration (UDDI) and Simple Object Access Protocol (SOAP) [76]. Figure 2.5 depicts the architecture of a Web Service, including its technologies [74].

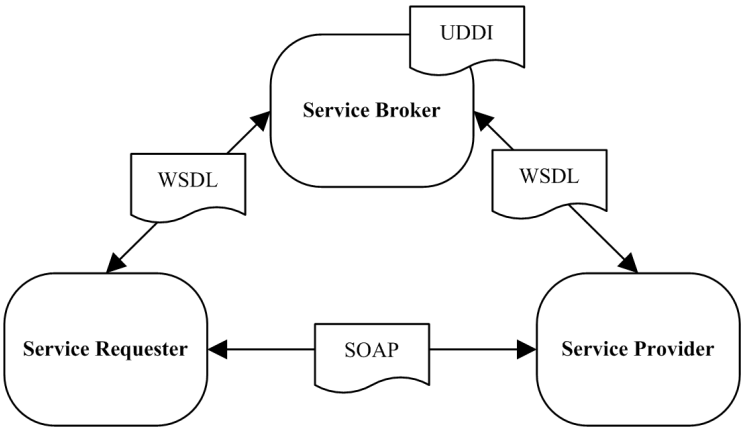


Figure 2.5: Architecture of a Web Service

WSDL is based in XML (Extensible Markup Language) and used to describe the programmatic interface of a Web Service [74]. The parameters involved in this kind of description can be definition of data types, the functions performed by a specific Web Service, message formats, among others. UDDI plays a major role in the action of a service requester. It provides a mechanism for registering and searching Web Services [74]. SOAP is the protocol used by Web Services to exchange data with client applications through the Internet. The description of this technology is particularly interesting for this project and it is discussed in more detail over the next section.

2.4.2 Simple Object Access Protocol

SOAP is a simple protocol for communications over the Internet, based on XML. It is used by client applications to request operations from Web Services. It was created under the supervision of the World Wide Web Consortium [74].

SOAP allows the carriage of XML encoded messages over higher-level and commonly used protocols like HTTP (HyperText Transfer Protocol) and SMTP (Simple Mail Transport Protocol) [74]. During a normal operation scheme, a client application sends a SOAP request to a particular Web Service, the latter performs the requested method and sends a SOAP response back to the client.

The next text section shows a SOAP request sent over HTTP to a Web Service example. This service simply takes a FEUP login and returns the department the login belongs to.

```
1  POST /mySOAPserver.php HTTP/1.0
2  Host: paginas.fe.up.pt
3  Content-Type: text/xml; charset="utf-8"
4  Content-Length: xxx
5  SOAPAction: "http://www.mySOAPserver.pt/mySOAPserver.php"
6
7  <SOAP-ENV:Envelope
8  (...)
9   <SOAP-ENV:Body>
10   <m:getMyDepartment xmlns:m="urn:FEUPdepartment"
```

```
11      <myLogin>ee04010</myLogin>
12     </m:getMyDepartment>
13    </SOAP-ENV:Body>
14   </SOAP-ENV:Envelope>
```

The first part of the text (lines 1 to 5) corresponds to the HTTP section of the request. The actual SOAP request is enclosed within a SOAP envelope (lines 7 to 14), which contains a SOAP body (lines 9 to 13), the element that carries the XML message to be transmitted [74]. The Web Service is requested to perform the method `getMyDepartment` (line 10) that takes as an argument the variable `myLogin` (line 11). After performing the requested operation, the Web Service responds with the following message:

```
1   HTTP/1.1 200 OK
2   Content-Type: text/xml; charset="utf-8"
3   Content-Length: xxx
4
5   <SOAP-ENV:Envelope
6    (...)
7    <SOAP-ENV:Body>
8     <m:getMyDepartmentResponse xmlns:m="urn:FEUPdepartment">
9      <yourDepartment>DEEC</item>
10    </m:setMeasuredValueResponse>
11   </SOAP-ENV:Body>
12  </SOAP-ENV:Envelope>
```

Since it is a SOAP message, the response is also initiated with a HTTP section. Just as the previous example, a SOAP body is involved by a SOAP envelope. After performing the requested method, the Web Service sends the results back to the client. In this case, the result is "DEEC", the value of the variable `yourDepartment` (line 9).

2.5 Summary

The survey of existent solutions and suitable technologies conducted in the initial phase of the project is documented in this chapter.

The first part, a research of related work, was essential to understand the characteristics of both commercial and academic solutions for electricity metering systems. The selection of relevant aspects from each one of them delivered a collection of concepts that proved to be useful for this project. The second part established the first contact with specific technologies such as electricity metering ICs, wireless communication solutions and tools used for remote communication. Furthermore, a comparison between different solutions inside the same technology scope allowed for a better understanding of their functionalities.

Chapter 3

System Specification

This chapter specifies the system designed to answer the challenges mentioned in chapter 1, without implementation details. First, a structural and functional overview of the complete system will be given, specifying the designation, position and function of its building blocks as well as the relations they establish between each other. The first part is followed by several sections that individually document the specifications of the several building blocks.

3.1 General System Specification

The general architecture of the system is depicted in figure 3.1. The system consists in three main blocks: *Home*, *Backoffice* and *User Interface*. While the first two are sets of physical components, the *User Interface* consists of both graphical data representations and the physical devices that enable access to them. The components interact with each other through different interfaces identified in figure 3.1 by letters. First, a functional description of the system components will be given followed by the specification of the interfaces that connect the components.

3.1.1 Components

This subsection will cover the general specifications of each one of the components identified in figure 3.1. The individual descriptions will be presented by paragraph, following figure 3.1 from left to right.

Electricity Meter

The *electricity meter* is the component that measures the electricity consumption of a single electrical appliance, i.e., for each appliance to be monitored there is an independent measurement. It encloses two submodules: one performs the consumption calculation, the *measuring module*, and

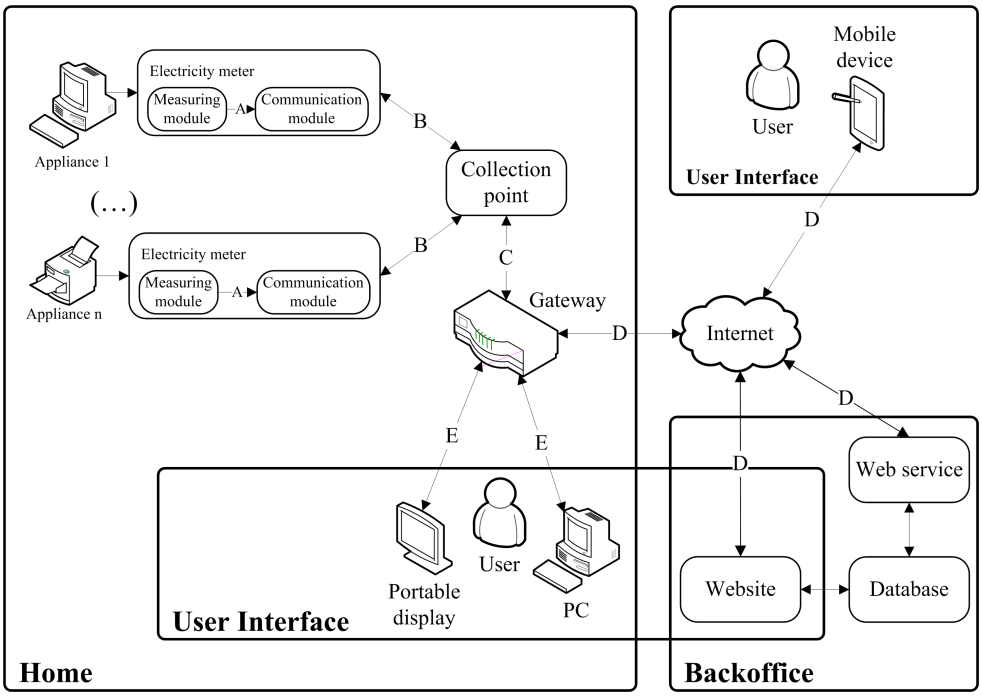


Figure 3.1: General system architecture

another, *communication module*, transmits the calculation result to a reception point, the *collection point*.

Collection Point

The *collection point* is a single component responsible for receiving data which is transmitted by one or more *electricity meters* and forwarding the information to another system component, the *gateway*. It is important to notice that this architecture clearly specifies the existence of only one *collection point* and many *electricity meters*, being the last controlled by the first.

Gateway

The *gateway* receives data from the *collection point*, processes it and then handles its transmission to a remote location, the *Backoffice* environment, in this case through the Internet. The *collection point* and the *gateway* may not necessarily be two separate devices and could be integrated into a single component.

Website, Web Service and Database

While not a component, the *Backoffice* holds the remaining system components, a server running a *web service*, a *database* and a *website*. The first is related to the communication of data between the *Home* and *Backoffice* sides, the second is responsible for storing the received data in a structured way while the third acts as the endpoint for user interaction with the system.

The *website*, mentioned in the previous paragraph, is also part of the third building block of the system, the *User Interface*. The *website* is responsible for the high-level processing of the measured data, generating and displaying visually appealing information, while interacting with the system *database*. Besides the *website*, the third main system block includes the physical devices that can access it. Examples of such devices are mobile devices, photo frame displays or regular PCs with browser support.

3.1.2 Interfaces

This subsection will cover the general specifications of each one of the system interfaces, identified in figure 3.1 by the use of letters. The individual descriptions will be presented by paragraph, following figure 3.1 from left to right.

Interface A

Interface A connects the two submodules included in the *electricity meter*, the *measuring* and *communication* modules. This interface should rely on a technology indicated for the communication within embedded circuits, since the *electricity meter* will consist in such a solution.

Interface B

Interface B consists of a wireless network formed by the *communication modules* included in one or more *electricity meters* and one *collection point*. It follows a master-slave architecture, where the master role is played by the collection point and the slaves by the *electricity meters*. The wireless technology to be used should be suitable to be used in residential environments and employed in home automation applications, such as ZigBee or Z-Wave, discussed in section 2.3.1.

Interface C

The connection between the *collection point* and the *gateway* is made through interface C. This should be based on USB, Ethernet or other technology that allows for automatic connection and transparent usability, i.e., the end-user of the system should only be concerned with connecting the two components, leaving the connection configuration to be automatically handled by them.

Interface D and E

Interface D corresponds to the Internet. Besides handling the communication channel between the *Home* and *Backoffice* sides, it also acts as a connection path to devices able to access the main Graphical User Interface (GUI) of the system, the *website*. These devices are then able to access it whether they are inside or outside the *Home* network. Interface E corresponds to a LAN (Local Area Network) or WLAN (Wireless Local Area Network) located within the *Home* environment, which allows the residentially located devices to reach the Internet, and therefore, the *website*.

The system follows a client-server architecture where the *Home* can be viewed as the client side and the *Backoffice* as the server side. It should be noticed that the relation between the number of *Home* and *Backoffice* sides is N:1, i.e., several *Home* sides can be served by only one *Backoffice*. The use of a remote location avoids the necessity of an *Home* machine capable of constant high-level processing of data, saving the end-user unnecessary material and power consumption costs. Since most of the present day households are now equipped with fast Internet connections, the system takes advantage of an already existent infrastructure, which virtually results in no additional costs for the end-user.

The terms formatted in italic (ex.: *Home*, *Backoffice* or *collection point*) are used throughout this document to identify the different system components. Figure 3.1 will be frequently referenced along this chapter, due to its importance on understanding the position and function of each individual block. The specification of each one of the modules and interfaces identified in this section will be covered in detail along the remaining parts of this chapter.

3.2 Home Environment

The *Home* environment is a main block of the system, as depicted on figure 3.1. It includes the components which are visible to the end-user, the *electricity meters*, the *collection point* and the *gateway*.

This subchapter provides brief specifications for each of the components included in the *Home* environment, starting with the *electricity meter*, followed by the *Home* wireless network (where the *collection point* is included) and finishing on the other edge, at the *gateway*.

3.2.1 Electricity Meter

The *electricity meter* is the physical component that calculates the electricity consumption of a given appliance. As seen on figure 3.1, the *electricity meter* encloses two modules: the *communication module* and the *measuring module*. Figure 3.2 provides a high level design of the *electricity meter*, with its functional blocks:

- Power supply: a block of circuitry that is capable of delivering power to both the *communication module* and the *measuring module*;
- Current sensor: a device capable of sensing the current flowing through an electric cable;
- Power metering IC: an integrated circuit equipped with resources to perform a electric energy calculation and output its value, given a current and a voltage as inputs (see section 2.2);
- Communication module: a slave node of the wireless network consisting of a wireless transceiver IC and RF front-end and the necessary firmware;

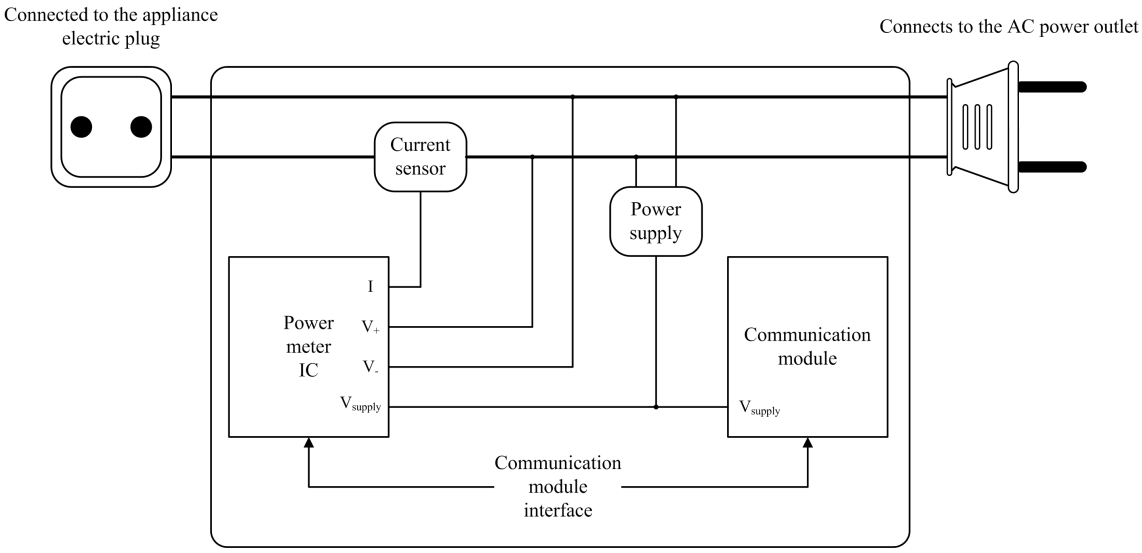


Figure 3.2: High level design of the *electricity meter*

- Communication module interface: a physical connection that allows the *communication module* to get data from the *measuring module*.

The *electricity meter* sits between the AC power outlet and the electrical appliance to be monitored. The *electricity meter* is then able to collect the power to operate, feed the current sensor with the current that flows through the appliance as well as to retrieve the applied voltage and communicating its value to the power metering IC. It should also be easily installed, i.e., its addition to the *Home* wireless network and registry within the system should be as easy and transparent as possible for the end-user.

3.2.2 Home Wireless Network

The communication between the components of the *Home* environment is done trough a wireless network, identified in figure 3.1 as interface B. The reason for this option is simple, since a wireless network is more flexible and requires fewer resources than a wired network.

The concept of flexibility includes the ease of adaptation to different kinds of household configurations, providing reliable communications despite of the number of floors, division sizes or shapes, etc. Mesh networking technologies [50], such as Z-Wave and ZigBee (discussed in section 2.3), achieve advanced levels of adaptation. Z-Wave was chosen since it is specifically designed for residential applications. Besides, Z-Wave is more advanced in terms of product interoperability than ZigBee, i.e., Z-Wave products from different manufacturers are, almost in all cases, compatible between each other, a characteristic not usually found on ZigBee certified products at the moment, as admitted by several ZigBee Alliance members [50, 69]. This is an important advantage as it leaves the door open for further integration possibilities.

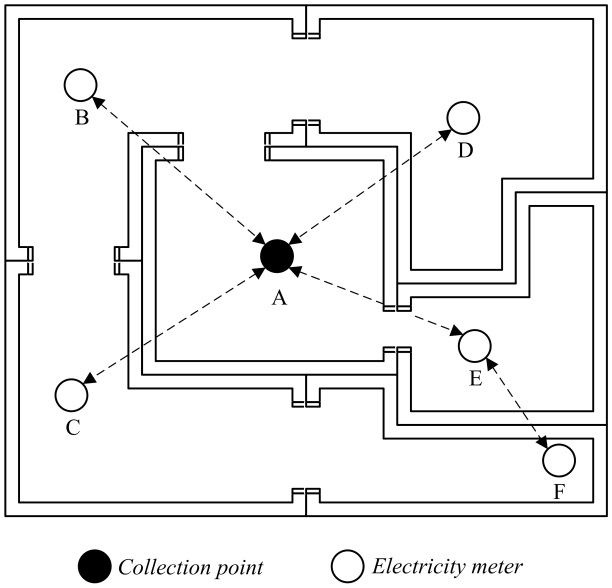


Figure 3.3: Typical layout of the *Home* wireless network

The wireless network is formed by several nodes scattered around the *Home* environment. The network follows a master-slave model: with the exception of the *collection point*, which behaves as the network master, all the nodes are *electricity meters* playing the role of slaves. Figure 3.3 shows a typical layout of the *Home* network, a mixture of star and mesh topologies [50]. The *collection point* is responsible for managing the inclusion and exclusion of other nodes as well as periodically requesting measurement values from *electricity meters*. Besides it should collect the answers to such requests and send them in an adequate way to the *gateway*. The *electricity meters* simply respond to commands sent by the *collection point*.

The *collection point* and the *electricity meters* communicate directly between each other, as far as possible, i.e., if the *collection point* cannot directly reach another node, it establishes a connection via other nodes operating as routers (refer to section 2.3.3 for further details about Z-Wave routing functionalities). All *electricity meters* should have the capability of routing frames to nearby nodes, so a meter is disconnected only if there is no other within its wireless range.

3.2.3 Gateway

As shown in figure 3.1, the *gateway* is the point that interfaces the *Home* environment and the Internet. The *gateway* hosts an application that consists of two main parts, one responsible for receiving data from the *collection point* (through interface C) and another part that should send the data to the *Backoffice* via Internet.

The *gateway* application should be capable of handling the reception and transmission of multiple data blocks, without bottlenecks in any of its two parts. It should run on a PC or other type of Internet interface, such as a router. The latter shows a big advantage as a *gateway* application host since it is a device that generally is always running, unlike a PC. This way, the *Home* side could

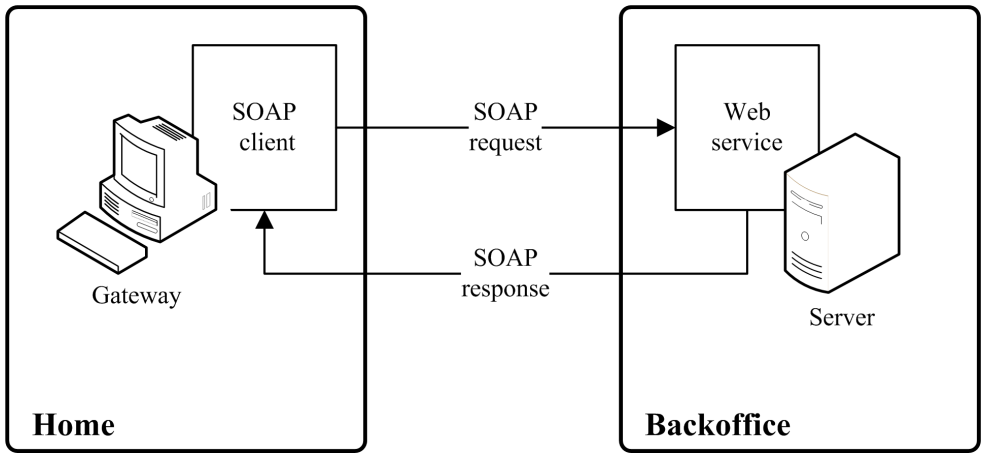


Figure 3.4: Description of the communications between the *Home* and *Backoffice* sides

constantly send consumption information to the *Backoffice* without the expenditure of unnecessary resources and power.

3.3 Backoffice Environment

As previously stated, the consumption data must be sent to a remote location in order to be fully processed and prepared for displaying.

This subchapter starts by describing several aspects concerned with the communication between the *Backoffice* and the *Home* environment, including security issues. The final part is reserved to the specification of the system *database*, resident at the *Backoffice*, which stores the measurement data in a structured way. Despite being physically a part of the *Backoffice*, the *website* specification is not covered in this section, as it belongs to the *User Interface*.

3.3.1 Web Services and SOAP

The communication between the *Backoffice* and *Home* sides is based on Web Services [73]. As described in section 2.4, Web Services are sets of web applications that can communicate using SOAP, an XML-based protocol designed for the exchange of messages in a distributed environment. SOAP presents another advantage as it features multi-platform interoperability, i.e., it allows for normal operation between objects of different platforms and/or languages. Other advantages include the re-use of existing routines and libraries which translates into an ease of development.

The description of the communication process is depicted on figure 3.4. A SOAP client application, running on the *gateway* sends SOAP requests to a remote server that hosts a custom Web Service, specified to execute different operations, all of them involving interactions with the main *database*:

- Save consumption data in a structured way;

- Register new *electricity meters*, when activated by a specific end-user;

3.3.2 Security in Remote Communications

To account for the dangers of communication of data over the Internet, namely attacks on data confidentiality, data integrity and end-to-end authentication (both client and server), two security mechanisms are used:

- SSL (Secure Sockets Layer) protocol [54]: used for server-side authentication and data encryption, providing data confidentiality and integrity;
- A *username & password* combination: used for client-side authentication, providing controlled access.

Without the use of SSL, the SOAP message parameters would be easily accessible to a third party, by inspection of the HTTP request over which the SOAP message is sent. With the SSL protocol, since it runs beneath the application layer (and, therefore, HTTP and SOAP), the SOAP parameters are encrypted and cannot be easily inspected by a third party [54]. In addition, SSL provides server-side authentication through the validation of the server's certificate by the client-side [54].

There is still the possibility for a third party to send wrong information to the server, assuming the identity of a legitimate client, given that it possesses the server certificate on his trusted certificates list. The use of a *username & password* combination as a parameter on a SOAP request implements a simple and safe mechanism for client-side authentication, since the client-side credentials would be also encrypted due to the use of SSL. Furthermore, the information will only be correctly accessed and processed if the credentials are consistent with those resident at the server.

3.3.3 Database

A database is used to store the measurement data in a structured way, following the entity-relationship model [77] shown on figure 3.5. The *database* stores information about the end-users of the system, the *electricity meters* owned by the various end-users and the measurements made by the *electricity meters* over the time. It is based on a simple model formed by three main entities:

- DEVICE: It works as the *database* representation of an *electricity meter*. It is uniquely identified by the attribute *DEVICE ID*. Another identifier, *HOME ID*, corresponds to its identification within the *Home* environment. The attribute *DESIGNATION* allows the end-user to optionally associate an *electricity meter* with an appliance, i.e., if the *electricity meter* would be monitoring a living room lamp, the *DESIGNATION* field could be marked as 'laptop' or 'table lamp';
- CUSTOMER: The entity that represents a end-user of the system. It is uniquely identified by the attribute *CUSTOMER ID* and is associated with several *DEVICE* entities, i.e., *electricity meters*;

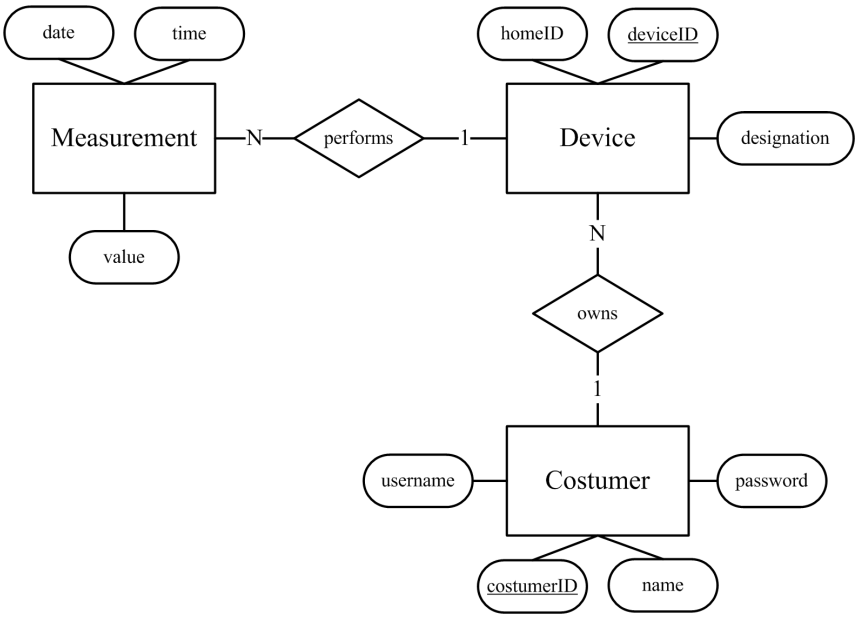


Figure 3.5: Entity-relationship model for the system database

- MEASUREMENT: Represents a measurement made by an *electricity meter*. A *DEVICE* can be associated with several *MEASUREMENTS*. A *MEASUREMENT* is made in a given *DATE* and *TIME*, holding a specific *VALUE*. In addition, a *MEASUREMENT* can be differentiated by the *DEVICE ID* of the *electricity meter* that performed it as well as by the *DESIGNATION* the *electricity meter* was holding at the moment of the action.

3.4 User Interface

The *User Interface* groups the application responsible for processing the consumption data and generating the visual information to be seen by the end-user and the renderer software - a common browser - that runs on devices used for displaying that same visual information, such as portable displays, mobile devices or a regular PC.

This subchapter presents the specification of the *User Interface*, including not only general technological aspects but also user interaction guidelines.

3.4.1 Concept

Figure 3.1 shows the position the *User Interface* block occupies within the whole system as its components. The visual output is presented in the form of a *website*, hosted in a web server running on the *Backoffice*. The reasons that sustain the web approach are the lack of need for a *Home* machine capable of constant high-level processing of data and the fact that it can be accessed from any location where an Internet connection is supported or from any platform that supports a browser. It also implies ease of development by re-utilizing existing routines and resources.

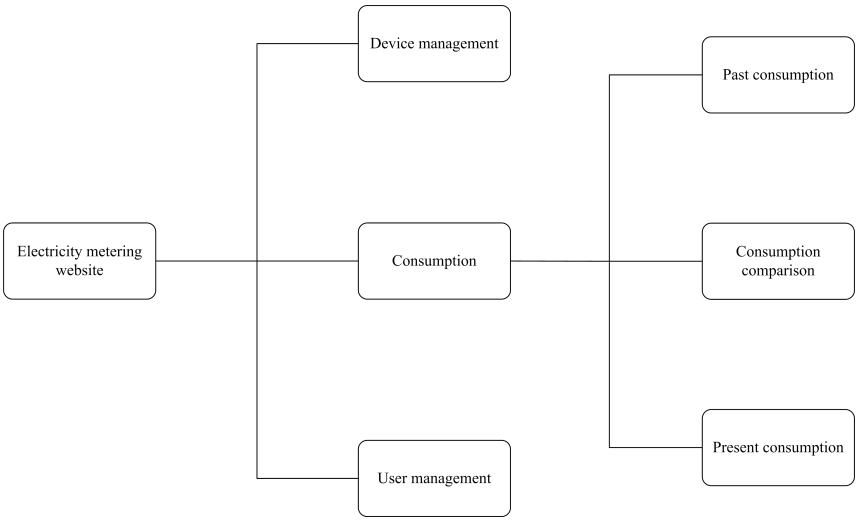


Figure 3.6: *Website* structure overview

In terms of user interaction, the main idea behind the *website* is to *inform* the end-user, displaying the electricity consumption information using visually appealing items such as charts and graphs. The *website* is also oriented to *instruct* the end-user, providing advice about how to reduce consumption costs. Overall, the following design guidelines drove the design of the *website* and visualization charts, in order to enhance the efficiency of the presented information [1]:

- Presentation of vivid and personalized information;
- Translation of energy units into more intuitive measures to enhance public motivation, for example, monetary losses or gains;
- Saving the end-user of complex information processing efforts, leaving those efforts to the *website*, for example, displaying future electricity bill estimations.

Besides the aspects directly related to the monitoring of electricity consumption, the *website* allows the end-user to manage owned *electricity meters*. The next sections will introduce a brief overview of the *website* structure by identifying and describing the contents of its main sections.

3.4.2 Structure Overview

The *website* is divided into three main areas, as shown on figure 3.6. The first, *user management*, includes functionalities specifically related to end-users such as *website* registration and account maintenance. The *consumption information* area covers the presentation of electricity consumption. It makes use of graphs, charts and other visual information. It is divided into three minor areas:

- Current (as in *real-time*) consumption: a section that shows the end-user the ongoing consumption pattern, showing the active *electricity meters* and their readings at the instant the end-user is accessing the information;

- Historic of consumption: a repository of consumption information on past periods;
- Comparison: a section that allows the user to compare its electricity consumption with national or regional averages, over current or past periods of time.

The consumption information is shown in a quantitative way using energy, equivalent CO_2 emissions and monetary units. Power or energy related values (W or Wh) can be directly displayed without previous processing, since this is the specified format for saved data in the system's database. However, that does not happen with the other unit types. The quantification of CO_2 emissions is indirectly related to the production of the consumed electricity. Specifically its calculation based on a factor, expressed in kg per kWh, that indicates the released amount of CO_2 during the production of 1 kWh in a certain country [78]. For Portugal the factor is approximately 0.55 kg/kWh, as taken from recent data collected by CARMA (Carbon Monitoring for Action) [78]. The monetary cost units follow the same principle, being the calculation based on a factor expressed in currency per kWh, established by the electricity provider service.

Considering the impact of the scaling units, consumption costs are considered to be more effective than bare energy units, being a strong motivator for logic realizations by the end-users [1]. This way, the default visualizations should be scaled in monetary units.

Eventual advices should be embedded along the *consumption information* area, helping the end-user to link the tips with the numbers, an option that helps them to make correct and informed decisions [1]. The third main part, *device management* is related to the control of the *electricity meters* owned by the end-user. The user can manually add or remove a device or change its designation (see section 3.3.3), as well as set or unset alarms or specify the periodicity of the measurements.

This ends the brief specification of the *User Interface* block of the system, to be implemented in the form of a *website*. The details about its specific functionalities and graphical aspect of the implemented version are discussed in the next chapter.

3.5 Summary

Besides the implementation phase, one of the main objectives of this project is to specify a solution for an electricity metering system with a user-friendly interface.

This chapter documents that same specification, dividing the system architecture into three main physical environments, *Home*, *Backoffice* and *User Interface*, the first located at the end-users household and the second at a remote location. The communication between the *Home* and *Backoffice* has been specified to be done over the Internet. Each of the two sides includes its own individual components and interfaces, all of them specified in this chapter. The *User Interface* comprises the visualization part of the system, which is specified to be a *website*. This solution makes the information reachable in any geographical location, as long as an Internet connection is available.

Chapter 4

System Implementation

The system under discussion involves several technologies that significantly differ from each other. Along with the problem of understanding each individual technology, there is the challenge of integrating them, in order to provide a complete and functional system.

This chapter introduces detailed implementation descriptions of the same parts discussed in the specification. The same structure of chapter 3 will be adopted, starting with the components included in the *Home* environment, followed by explanations of the *Backoffice* and *User Interface*.

4.1 Home Environment

This subchapter describes the implementation details of the components and interfaces which are present in the *Home* environment. The first component to be addressed will be the *electricity meter*, followed by the *Home* wireless network (where the *collection point* and *communication module* are included), finishing with the description of the *gateway*.

4.1.1 Electricity Meter

The *electricity meter* is the component that calculates the electricity consumption of a given appliance. Given the brief general specification of the energy meter in section 3.2.1, its implementation is now described. The next sections will address several design aspects about each one of the functional blocks identified in section 3.2.1.

4.1.1.1 Power Metering IC

From all the solutions presented in section 2.2, two alternatives were left to be chosen: the Analog Devices ADE71xx and the Teridian 71M6511. The two are similar to each other, both offering a similar range of functionalities and development tools. The solution from Teridian offers a slight advantage in terms of cost (refer back to section 2.2.5). The final choice was the Teridian 71M6511

Table 4.1: Comparison between shunt resistors and current-sense transformers.

Sensor type	Advantages	Disadvantages
Shunt resistor	Passive device, therefore easy to apply.	Expensive as an off-the-shelf component. Low precision. Large size.
Current-sense transformer	Cheaper than shunt resistors. Easy to apply in the final circuit.	Accuracy problems for low frequencies.

due to a question of availability and the impact of this factor on the time constraints of this project. Once the power metering IC was chosen, the necessary surrounding circuit design started to be done:

- Voltage and current input circuits[1];
- Power supply design;
- Interface with the *communication module*;
- Other circuitry.

4.1.1.2 Current Sensor

The current sensor is the device that senses the current consumed by the monitored electrical appliance. The chosen power metering IC, the Teridian 71M6511, supports the following current sensor types:

- Shunt resistors;
- Current-sense transformers;
- Rogowski coils.

Rogowski coils are the most expensive of the three types, leaving the final choice to either the shunt resistor or the current sense transformer. Table 4.1 provides a brief comparison between the two. Since the main objectives of this project are to create a low-cost solution within a relatively short time frame, the chosen sensor type was the current transformer, specifically the CSE187-L by Triad Magnetics, an off-the-shelf (OTS) component designed to monitor current in low frequency applications [80]. It is able to monitor currents from 100 mA to 30 A at frequencies from 50 to 400 Hz and presents a turn ratio (primary to sense) of 1:500 [80]. Its unit price can go down to 1.74 € [39], for very large quantities.

Besides the choice of the appropriate current sensor, there is the problem of correctly interfacing it to the power metering IC. Figure 4.1 depicts the basic circuit for the current input pins of the Teridian 71M6511, when using a current-sense transformer [79].

[1] The Teridian 71M6511 performs the necessary computations to accurately measure real and apparent power: offset cancellation, phase compensation, frequency detection, VAR calculation, peak detection, voltage phase measurement, etc. [79].

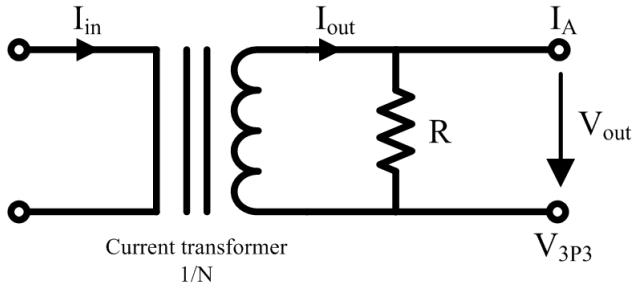


Figure 4.1: Current-sense transformer interface circuit for the Teridian 71M6511

The Teridian 71M6511 specifies that the voltage at pins I_A and I_B must be within $\pm 250\,\text{mV}$ [79]. By inspection of the circuit on figure 4.1, that means the current I_{out}, flowing through resistor R, must create a voltage drop that should not exceed the $\pm 250\,\text{mV}$ range. The necessary dimensioning effort is then confined to finding a suitable value for R. Considering that $I_{in} \leq 208\,\text{A}$, as recommended by Teridian [79], the value of R can be calculated through equations 4.1 and 4.2:

$$V_{out} = R \times I_{out}, \tag{4.1}$$

the Ohm's law.

$$I_{out} = \frac{I_{in}}{N}, \tag{4.2}$$

where N is the transformer winding ratio, 500:1 in this case.

Replacing I_{out} in equation 4.1 by 4.2, results in equation 4.3:

$$V_{out} = R \times \frac{I_{in}}{N} \Leftrightarrow R = \frac{N \times V_{out}}{I_{in}} \tag{4.3}$$

Considering that for $I_{in} = 208\,\text{A}$, $V_{out} = 250\,\text{mV}$ and $N = 500$, the resulting value for R is approximately $425\,\text{m}\Omega$. With calculation of R the dimensioning of the current input circuit is finished. More details about the final circuit can be found further ahead in this chapter.

4.1.1.3 Power Supply

The design of a low-cost and suitable power supply for the *electricity meter* was one of the main steps of the overall circuit design. The power supply must provide the necessary power for the *measuring* and *communication modules* of the *electricity meter* to operate at their peak consumptions.

The main requirements for the design are related to the load it must feed [81], which is mainly composed by the power metering IC, the Teridian 71M6511, and the Z-Wave transceiver IC, the Zensys ZM3102N. Their DC characteristics, specifically required supply currents and voltages are given in table 4.2(a) [35, 68]. By analysing the values, the specification present in table 4.2(b) can be defined for the power supply.

Table 4.2: Definition of the power supply general specification

(a) Values of supply currents and voltages for the Teridian and Zensys chips [35, 68]

Module	Parameter	Minimum	Typical	Maximum
Teridian	Current	-	6.4 mA	9.5 mA
	Voltage	3.0 V	3.3 V	3.6 V
Zensys	Current	2.5 μA	23 mA[1]	36 mA[2]
	Voltage	2.1 V	3.3 V	3.6 V

[1] Zensys in receive mode with the MCU running [68].
[2] Zensys in transmit mode ($P_t = -2$ dBm) with the MCU running [68].

(b) Power supply general specification

Description	Parameter	Minimum	Typical	Maximum
Input	Voltage	85 VAC	-	265 VAC
	Frequency	47 Hz	50/60 Hz	64 Hz
Output	Voltage	3.0 V	3.3 V	3.6 V
	Current	6.5 mA[1]	29.4 mA[2]	45.5 mA[3]

[1] Zensys on sleep mode, Teridian on normal operation (2.5 μA + 6.4 mA).
[2] Zensys on receive mode, Teridian on normal operation (23 mA + 6.4 mA).
[3] Zensys on transmit mode, Teridian under max. current (36 mA + 9.5 mA).

Capacitive Power Supply

The first version of the power supply followed the suggestion made by the manufacturers of the Teridian power metering IC [79]. A simplified but illustrative version is depicted in figure 4.2, a capacitive power supply [82], which does not make the use of transformers, making it more cost effective[82]. The last fact also makes it non-isolated, which introduces electrocution hazards.

The input current, I_{in}, charges the capacitor C_2 until the voltage at its terminals reaches $V_z - V_d$, i.e., the breakdown voltage of the Zener diode, D_1, minus the forward voltage drop at D_2. The output voltage, V_{out}, is then $V_z - V_d$. As the desired output current, I_{out}, rises, the capacitor C_2 takes a longer time to charge up completely, and, if the output current rises over a certain value, the output voltage will not reach $V_z - V_d$. This happens because a significant part of I_{in} is being drawn by the load and not only charging C_2. The original circuit suggested by Teridian includes a voltage regulator which maintains the output voltage at 3.3 V, the required supply voltage for the *measuring* and *communication* modules. The circuit of figure 4.2 also presents circuit protection elements, specifically a MOV (Metal Oxide Varistor), VR_1, and a fuse resistor, R_1:

- The MOV provides transient protection to the circuit;
- The fuse resistor, protects the circuit against excessive current.

Several simulations have been conducted in NI Multisim, a circuit design program developed by the National Instruments Electronics Workbench Group [83]. In order to simulate a load which would behave as the combination of the Teridian 71M6511 and the Zensys ZM3102N ICs,

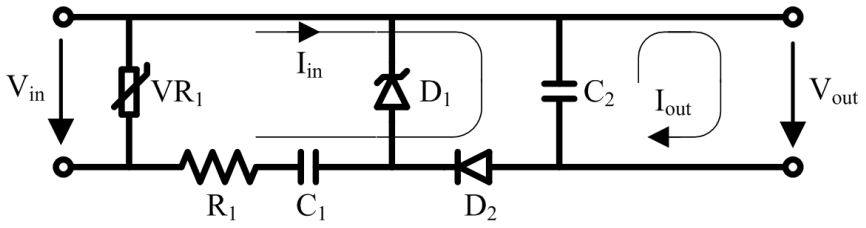


Figure 4.2: Capacitive power supply circuit

the equivalent resistances have been calculated for each one of them. Following the Ohm's law, $V = R \times I$, and according to the values on table 4.2, the equivalent resistance values are approximately 360 Ω for $R_{Teridian}$ and 91 Ω for R_{Zensys}. The two resistances were applied in parallel to the terminals of the power supply circuit and the output current and voltage behaviour was analysed.

The results concluded that the capacitive power supply design was not able to supply enough current to both ICs. The reduction of the combined impedance formed by the input stage (formed by C_1 and R_1) produced more satisfactory results. However the excessive output ripple and poor power efficiency of the circuit led to the consideration of a new design, described in the next section.

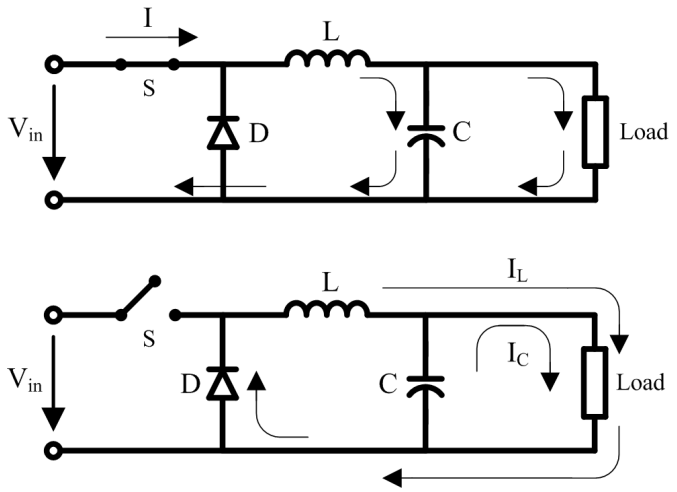
Switched-Mode Power Supply: Buck Converter

The second and final design was based on a non-isolated switched-mode power supply (SMPS) [84], as shown on figure 4.3(a). It's main function is to convert the 220 V AC from a normal wall outlet to a lower voltage, 3.3 V DC in this case, in order to correctly supply voltage and current to the *electricity meter*. Before describing the final design in detail (shown on figure 4.4), the concept of a SMPS is explained.

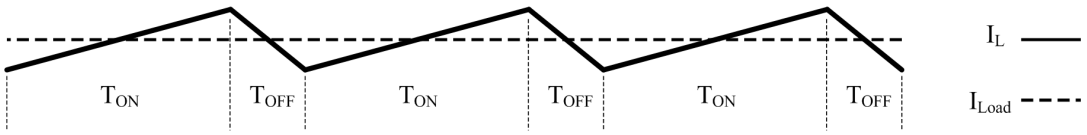
Figure 4.3(a) presents the basic design of a buck converter [84], a commonly used type of SMPS. It is used to convert a DC voltage, V_{in}, to a lower DC voltage, V_{out}. Its key components are identified in figure 4.3(a): a switch S which cycles through ON/OFF states with a given frequency and alternately connects/disconnects V_{in} to the circuit, a diode D, an inductor L, and an output capacitor C. The arrows on figure 4.3(a) explain how current flows through the circuit during the ON/OFF switching cycles. During the ON cycle, the voltage $V_L = V_{in} - V_{out}$ is sensed across L, which causes the inductor current, I_L, to increase according to:

$$V_L = L_o \frac{\partial i_L}{\partial t}, \tag{4.4}$$

where L_o is the inductance of the inductor L in Henry (H). The inductor current then flows to the load and the capacitor C, charging it. Diode D does not conduct since it is reverse-biased. During the OFF cycle V_{in} is disconnected from the circuit, the voltage across L is $V_L = -V_{out}$ (disregarding the voltage drop across D), making the diode to be forward-biased. Since I_L cannot change instantly, its initial value will be equal immediately before and after the switching instant. However, due to the negative value of V_L, the value of I_L starts to decrease with time, which results



(a) Modes of operation of the buck converter (ON/OFF switching cycles) [84]



(b) Inductor current, I_L, and equivalent load current, I_{Load}, during the ON/OFF switching cycles [84]

Figure 4.3: Basic aspects of a SMPS type, the buck converter

in a pattern similar to that shown on figure 4.3(b). The decreasing I_L flows trough the load, as well as the current from the discharging capacitor C, making $I_{Load} = I_L + I_C$.

The main advantages of using a SMPS, when compared to other types (resistive, capacitive, etc.) are [84]:

- Higher power efficiency;
- Lower heat generation due to higher power efficiency;
- Smaller size and lighter weight (elimination of transformers and other large components);
- Lower cost (elimination of transformers and other large components).

After the brief introduction to a SMPS, the next section describes the final design of the power supply used in the *electricity meter*.

Final Design

The final design is shown on figure 4.4. It is similar to the basic design shown on 4.3(a) with the addition of an input stage to provide circuit protection, AC rectification and EMI (Electromagnetic Interference) reduction [84, 85], and an output stage that provide regulation to the 3.3 V output.

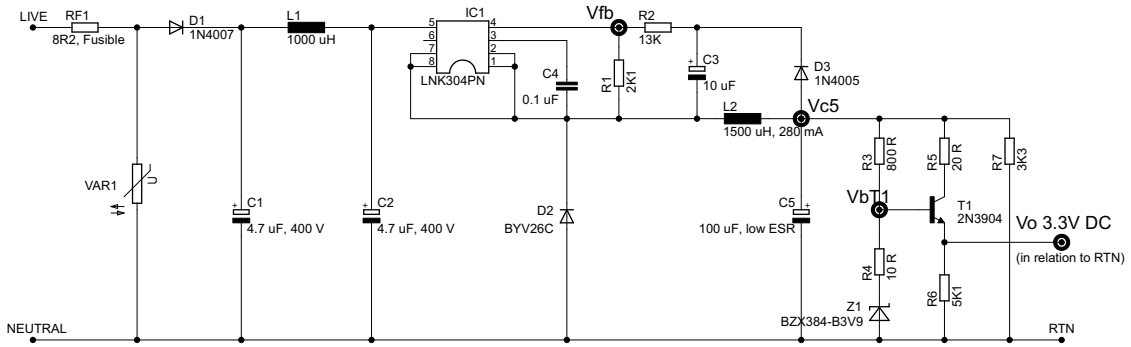


Figure 4.4: Final design of the buck converter used to supply the *electricity meter*

IC_1, the LNK304PN developed by Power Integrations [86], acts as the buck converter's switch, operating as an high voltage switched current source (switching frequency of 66 kHz [86]). The regulation of the voltage that supplies the output stage, V_{C_5}, is accomplished by adjusting the ratio of enable/disable switching cycles of the LNK304PN's internal current source, depending on the value of the feedback current sensed at pin 4 of IC_1, I_{FB} [86]. The voltage divider formed by resistors R_1 and R_2 on the feedback loop is defined such that $V_{FB} = 1.65\,\text{V}$ when the voltage supplied to the output stage, V_{C_5} (equivalently, the voltage across C_3), is equal to 12 V. $V_{FB} = 1.65\,\text{V}$ is specified by IC_1 for $I_{FB} = 49\,\mu\text{A}$, and according to the value of I_{FB}, it behaves in the following manner:

- If $I_{FB} > 49\,\mu\text{A}$, it means the supplied current is sufficient, therefore the current source switches off for the next cycle;
- If $I_{FB} < 49\,\mu\text{A}$, it means the load is demanding current, therefore the current source switches on for the next cycle.

Besides the switch, the other typical buck converter components are present, the inductor L_2, the diode D_2 and the output capacitor C_5, which was chosen to hold a low ESR value (Equivalent Series Resistance) [84, 85] in order to reduce output ripple [86]. Its functionality was already addressed in the previous section, with a difference: instead of switching ON/OFF a voltage source, a switching current source is used, forcing a time-varying current through the inductor L_2, and therefore creating a voltage across it. The values of L_2 as well as the model to be used for D_2 were defined according to the guidelines provided on [86], an application note provided by Power Integrations.

The input stage presents protective circuitry (a topic addressed on the capacitive power supply section), the diode D_1 is used to rectify the 220 V AC input voltage and a π-filter - formed by C_1, C_2 and L_1 - reduces EMI.

The output stage follows an emitter follower configuration [85], with T_1 (a npn BJT [85]) working in the active region (therefore $V_{CE_{T_1}} \geq 0.3\,\text{V}$), and a Zener diode, Z_1, which are used

together used to provide good regulation to the 3.3 V output. Z_1, when biased by an appropriate current, generates a 3.9 V voltage across it[2]. Together with the voltage drop created by R_4, results in $V_{B_{T1}} \approx 4$ V. Since T_1 is a npn BJT, one can consider $V_{BE_{T_1}} \approx 0.7$ V. Then the output voltage becomes:

$$V_o = V_{B_{T1}} - V_{BE_{T_1}} \approx 4 - 0.7 \approx 3.3\,\text{V} \tag{4.5}$$

As mentioned above, $V_{C_5} = 12$ V, and R_3 is used to set a 10 mA bias current for Z_1, therefore:

$$R_3 = \frac{V_{C_5} - V_{B_{T_1}}}{I_{Z_1}} \approx \frac{12-4}{10 \times 10^{-3}} \approx 800\,\Omega \tag{4.6}$$

R_5 is used to keep the collector current of within T_1's maximum ratings[3]. The evaluation results of the design's performance can be consulted in section 5.1.1, which include a testing prototype built on a PCB (see appendix B for details on the schematic and PCB layout). In the end, a power supply with the following general characteristics was achieved (more details of its performance on section 5.1.1):

- Input voltage: Supports multiple input line voltages ranges, 85 VAC to 265 VAC, as well as different line frequencies, 50 and 60 Hz;
- Output voltage, V_{out}: 3.3 V;
- Output current, I_{out}: Up to 110 mA;
- Low cost due to the lack of an expensive transformer (more details on cost section 5.1.1);

4.1.1.4 Communication Module Interface

The *measuring* and *communication modules* specification previewed the exchange of data between each other through the UARTs on each one of the module ICs. However, there are communication protocols which are more suited to the communication between microcontrollers such as I^2C [33]. I^2C is simple and requires low manufacturing costs when compared to other solutions such as SPI or RS-232 [33]:

- It consists of a multi-master serial computer bus developed by Philips;
- Makes use of only two bidirectional lines, one for data communication, SDA (Serial Data), and another one for the synchronization of the data transfer , SCL (Serial Clock);
- The required hardware consists of two pull-up resistors;
- It can be implemented by software, requiring only two multi-use pins on each one of the microcontrollers, available both in the Teridian 71M6511 and the Zensys ZM3102N [35, 68].

[2]The model used for Z_1, BZX384-B3V9, presents a tolerance of 2% [87].
[3]The maximum value for I_C is 200 mA [88].

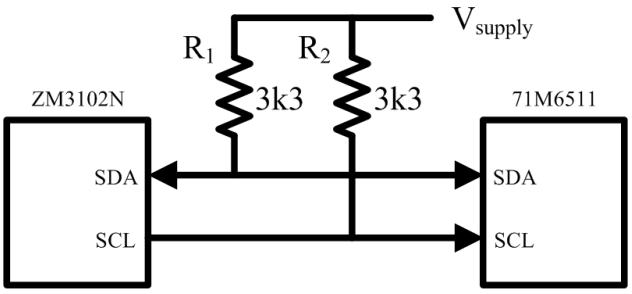


Figure 4.5: Simple circuit for the I^{2}C interface between the *measuring* and *communication* modules

An illustrative version of the interface circuitry between the Teridian 71M6511 and the Zensys ZM3102N is depicted in figure 4.5. The first already has dedicated pins for a I^{2}C interface which is used to drive a serial EEPROM (Electrically Erasable Programmable Read-Only Memory). Since I^{2}C is a multi-master method, the Zensys ZM3102 can also be included in the I^{2}C circuit and assume a master role. The pull-up resistors, R_1 and R_2, where dimensioned according to the guidelines present on [35]. The low hardware effort required by I^{2}C is compensated by the necessary software implementations on both ends of the communication channel. The software implementation of I^{2}C in microcontrollers is rather standardized [33], consisting of simple methods that control the operation of the I/O pins defined as SDA and SCL [33].

4.1.1.5 Overall Circuit Design

The individual design efforts documented all over the previous sections have been combined into the schematic shown on appendix A. Besides the parts discussed over sections 4.1.1.1 to 4.1.1.4, several other minor additions were made, based on the design guidelines of the Teridian 71M6511 demonstration board [79]. The different parts are separated in the schematic shown in appendix A by numbers. Their description is listed below (except for the components already addressed on sections 4.1.1.1 to 4.1.1.4):

1. Power supply;

2. Current input circuit;

3. Voltage input circuit: dimensioned by following the guidelines present in [79]. It consists of a resistor ladder, R_{28}, R_{30}, R_{31} and R_{32}, with decreasing values, from line to the Teridian 71M6511. Together with capacitor C_{20}, a low pass filter is formed. The first resistor, R_{28}, presents the highest value, 2 MΩ, which results in the higher voltage drop. The voltage drops at R_{30} to R_{32} will be in a safer range, and care about leakage or arching effects need only be taken for R_{28} [79];

4. *Communication module* interface: (a) a simple connector for a Z-Wave module based on the Zensys ZM3102N chip (see sections 2.3.3.2 and 4.1.2.2 for further details), (b) a simple LED driving circuit that allows to implement the signalling functionalities described in table 4.3, (c) the I^2C interface connections;

5. Connection to the debug board: Since this is a schematic of a prototype version, the connection to the debug board has been included, which can be useful for eventual debugging issues [79];

6. Connection to a flash programmer [79]: once again, due to the prototype nature of the circuit, the connection to a flash programmer has been included;

7. Reset switch circuitry: Although this is not a major requirement for a functional energy meter, it is useful for the prototype phase.

Although the schematic has been completed, the same did not happen with the physical implementation of it, due to time and logistic constraints. The integration of a functional *electricity meter* in the remaining parts of the system was accomplished with the Teridian 71M6511 Demonstration Board, as documented in the next section.

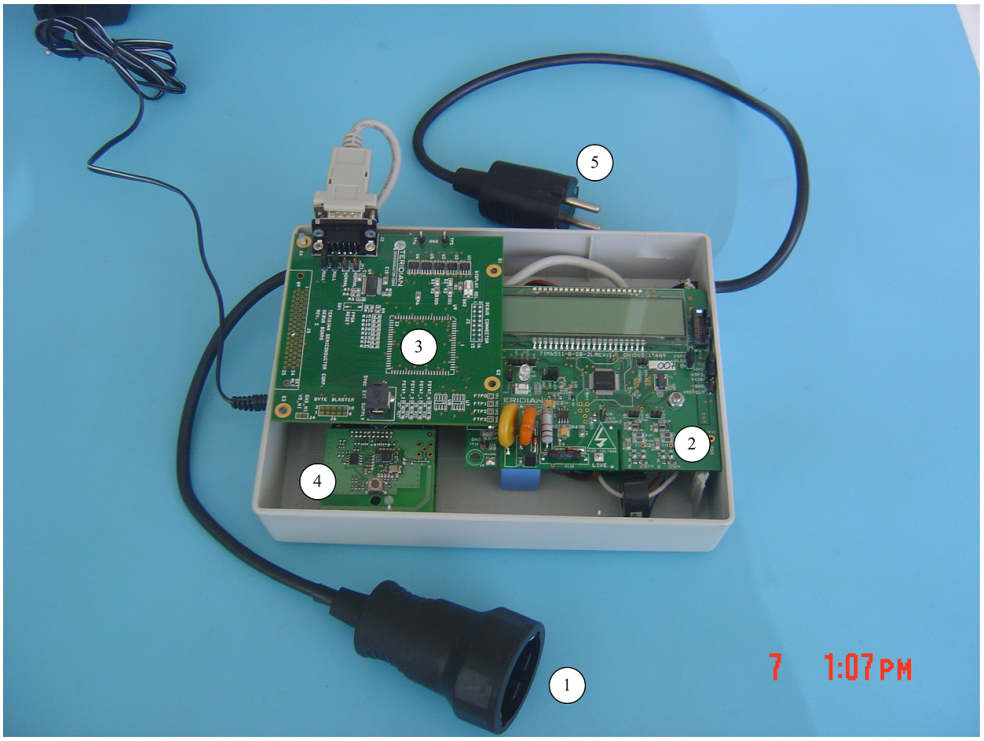
4.1.1.6 Teridian 71M6511 Demonstration Board

The practical integration of the *measuring* and *communication modules*, as specified in section 3.2.1, was accomplished with the 71M6511 demonstration board provided by Teridian [79]. Among others, this development tool includes the following features:

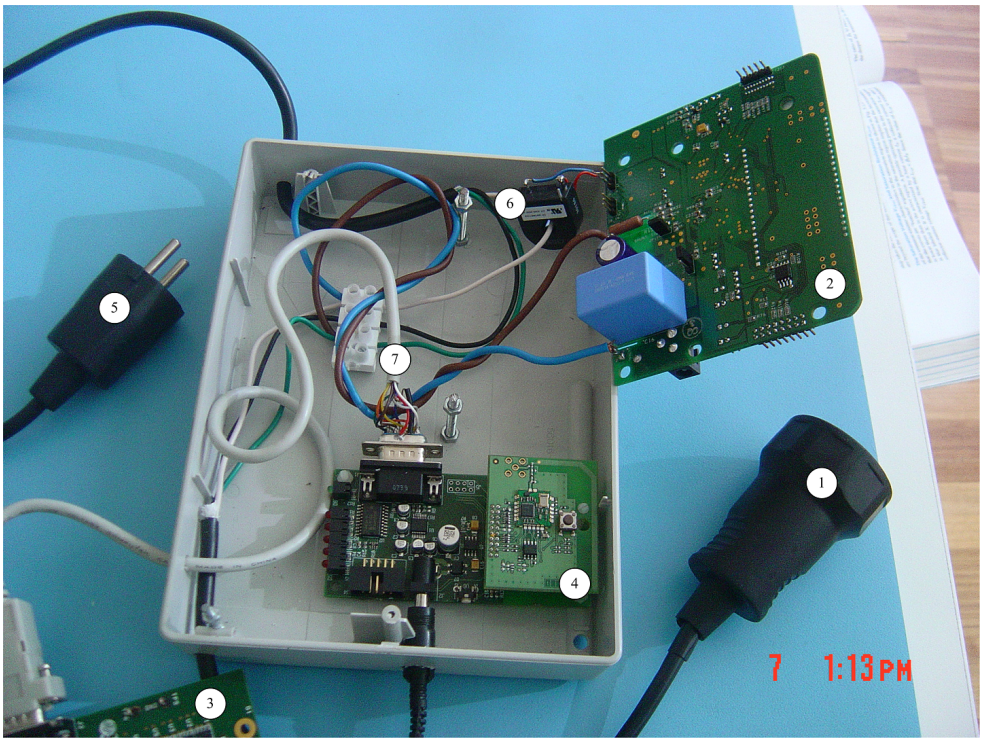
- A ready-to-use PCB with a pre-programmed 71M6511 IC and necessary peripheral circuitry;
- A LCD display, used to show consumption data;
- On board capacitive power supply (see section 4.1.1.3);
- A RS-232 serial interface;
- Useful documentation that includes hardware and software design guidelines.

The first step of the board setup was to prepare it for monitoring an electrical appliance, i.e., to assemble a module similar to figure 3.2. The board already contains a circuit to interface a current sense-transformer with a 2000:1 winding ratio [79], therefore another current-sense transformer was needed that not the CSE187-L, mentioned in section 4.1.1.2, (see number 6 on figure 4.6(b)). The final result is shown in figure 4.6. The components have been accommodated and fixed into a rectangular plastic box, in order to make it suitable for easy handling.

The connection with the *communication module* was made by through a RS-232 cable connected to the interfaces of both the demonstration board and a Z-Wave hardware module (see



(a) Top view: (1) monitored electrical appliance plug (2) Teridian 71M6511 Demo board (3) Teridian Debug board (4) Z-Wave *communication module* (5) *electricity meter* AC power plug



(b) Bottom view: (6) current-sense transformer (7) RS-232 serial interface between the *communication* and *measuring* modules

Figure 4.6: The *electricity meter* prototype achieved by using the Teridian 71M6511 Demo Board

number 7 on figure 4.6(b)). The next phase was focused on understanding the integration of the *communication module* with the demonstration board. The latter is pre-programmed with a code that performs calculations of instantaneous values such as V_{RMS}, I_{RMS}, instantaneous power values (in W, VA), and energy values (in Wh, VARh and VAh). The code can also interpret commands received by the 71M6511 UART, some of them answered with variables that hold calculated values. The accumulation values, such as the accumulated energy, in Wh, are stored in 64 bit variables. According to the demonstration board specifications, each additional bit of the variable corresponds to an increment of 8.35×10^{-8} Wh [79], meaning that the accumulator will hold up to $2^{64}\times8.35\times10^{-8}$ Wh which results in 1.54×10^{12} Wh. Other details about the actual communication of the values hold by these variables are addressed in section 4.1.2.4. In this case, the *communication module* retrieves a value in Wh from the *measuring module*.

The Teridian 71M6511 demonstration board was essential to the project as it allowed to achieve a first version of the complete system, even without the finished hardware prototype of the *electricity meter*, whose design has been discussed along the previous sections of this subchapter.

4.1.2 Home Wireless Network

This section describes the implementation details of the communication between the *electricity meters* and the *collection point* inside the *Home* environment. Firstly, a few paragraphs will address pertinent aspects about the adopted implementation tools. The remaining sections will focus on the implementation of specific components of the *Home* wireless network.

4.1.2.1 Implementation Tools

As stated in section 3.2.2, the adopted wireless technology is Z-Wave, a proprietary protocol of the Danish company Zensys. Zensys provides a development kit [89] which was the main tool used on the implementation of the *Home* wireless network. The development kit includes RF modules based on the Zensys ZW3102N (see section 2.3), that can be programmed to operate as Slaves or Controllers. Two kinds of RF modules were used, a simple model to implement Slave devices and another for Controllers:

- Slave model: Equipped with 5 controllable LEDs (Light Emitting Diode), 1 push-button and a RS-232 serial interface, among other features;
- Controller model: Equipped with 6 controllable push-buttons, 7 controllable LEDs and a RS-232 serial interface, among other features.

The necessary software and hardware resources to program the RF modules are also included in the set. Finally, the kit provides software and hardware tools to analyse the network traffic exchanged within a Z-Wave network. Appendix D provides some images of the hardware tools provided by the development kit.

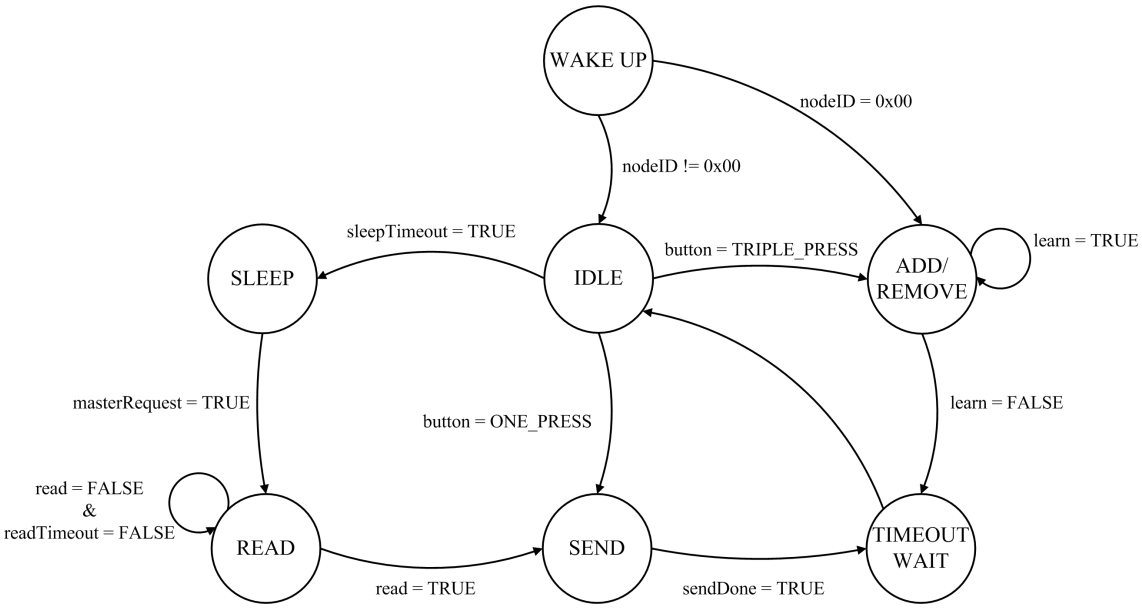


Figure 4.7: State diagram that describes the *communication module* state machine

4.1.2.2 Communication Module

The *communication module* is a building block of the *electricity meter* (see section 3.1). The main function of the *communication module* is to retrieve consumption data from the *measuring module*, via a serial connection, upon request from the *collection point*. It then packages the information into Z-Wave protocol frames, and transmits them back to the *collection point*. It can also route Z-Wave frames to other nodes, if the direct connection between a neighbouring *electricity meter* and the *collection point* is not available. Therefore it is defined as a Routing Slave device. The 8051 MCU core of the Z-Wave chip included in the *communication module* was programmed to implement the state machine described by the diagram shown on figure 4.7, and explained in detail in the next paragraphs.

Initialization Process

When powered up, its state is defined to WAKE UP. When it does it for the first time, the *communication module* is not yet included in the *Home* wireless network and therefore it has not been given a node identification by the *collection point*. In this case, the module changes its state to ADD/REMOVE, waiting for the *collection point* to include it in the network. When the inclusion process is finished, a flag value is changed (LEARN), making the state change into TIMEOUT WAIT, triggering a sleep timer. The IDLE state follows immediately. If the sleep timer is allowed to reach the end of its countdown while in IDLE state, the Z-Wave chip is powered down and left in SLEEP state. The exclusion and further inclusion processes follow the same sequence of states as the initial inclusion with the difference of being triggered by pressing the push-button twice.

Table 4.3: Interpretation of the *communication module* LED combinations

LED 1	LED 2	LED 3	Description
Off	Off	Off	The *communication module* is powered down.
Off	Off	On	The module is waiting for the reception of data over the RS-232 serial interface.
Off	On	Off	The module is idle, waiting for a sleep timeout.
On	On	Off	The module is waiting to be added to the network.

Power Saving Method

In order to save power, the *communication module* is powered down for most of the time, operating in the *frequently listening* mode[4]. While in this mode, the whole Z-Wave chip is turned down with the exception of the RF transceiver, which is powered up every 1 second in order to check for *collection point* requests.

Reading Consumption Values

When a *collection point* request is received, the Z-Wave chip is powered up and enters into READ state. While in this state, the *communication module* sends a command to the *electricity meter's measuring module* through the RS-232 interface, requesting the last reading stored in memory. It waits for a response for 2 seconds after changing the state to SEND. The received value is transmitted to the *collection point*, and the module goes into TIMEOUT WAIT state after the transmission is completed. After that, the module follows the same sequence of events as described above. The operation of the communication module can be partially followed for debugging purposes, by interpreting its LEDs, as shown on table 4.3.

4.1.2.3 Collection Point

The *collection point* is the central element of the *Home* wireless network, behaving as a Z-Wave Static Controller device. Besides managing the inclusion and exclusion of other nodes or maintaining the network routing tables, its main purpose is to request measurements from the *communication modules* with a periodicity of 60 seconds, collect all the responses and send them to the *gateway* through a RS-232 serial connection. The 8051 MCU (Microcontroller Unit) core of the Z-Wave chip included in the *collection point* was programmed to implement the state machine shown on figure 4.8, and explained in detail over the next paragraphs.

Initialization

The *collection point* is implemented as a Z-Wave development kit controller model [89] and makes use of two buttons to account for user interaction. The *collection point* starts at the POWER UP

[4]Specific Z-Wave term. The Z-Wave radio on the transceiver powers up for a short period every 1 second and if it receives a command, the whole module powers up [66].

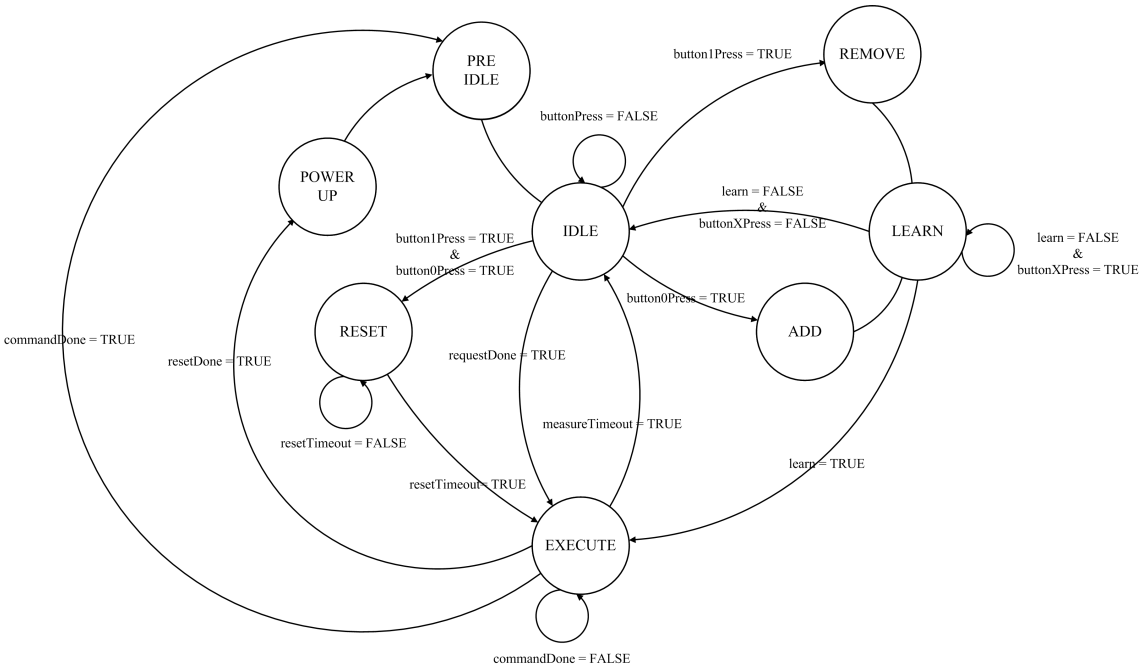


Figure 4.8: State diagram that describes the *collection point* state machine

state, where the list of nodes included in the controlled network is updated and fetched from an external memory. The PRE IDLE state follows immediately, handling the start of a timer of 60 seconds, used to trigger the measure requests to the *electricity meters*. After this, the system falls into the IDLE state, and it remains there until a button is pressed. While in this state, the user can choose three different actions: add a new node, remove an existing node or reset the *collection point*.

Add and Remove Functionalities

By pressing one of two buttons the user can instruct the *collection point* to handle the inclusion or exclusion of a *communication module*. Upon that action, the system falls from IDLE into ADD or REMOVE state where the node inclusion or exclusion process is initiated. The system them falls into LEARN state, i.e., the *collection point* waits for a *communication module* to announce its intention to be included or excluded, which happens if the latter is in the ADD/REMOVE state (see section 4.1.2.2). If a *communication module* does not respond while the controller is in the LEARN state, the user can cancel the action by releasing one of the buttons. If a response is detected, the system falls into EXECUTE state where no user action can cancel the process until it is completed. The measure timer is also stopped in order not to interfere with the process. When finished, the system falls into PRE IDLE state, and follows its normal flow.

Reset Mode

The reset action is required for clearing the complete list of nodes included in the network controlled by the *collection point*. This action is achieved by pressing the two interaction buttons at the same time. When the reset action is requested, the system falls into RESET state. Upon reaching this state, a timer of 5 seconds is started, giving the user a chance to cancel the action by simply releasing one of the two buttons, leaving the system into IDLE state again. If the timer ends, the system falls into EXECUTE state, where no user action can cancel the reset. Once again, the measurement timer is also stopped in order to prevent interference. When the reset is completed, the system goes into POWER UP state, and then follows its normal course of action. Since the user cannot start to press the two buttons exactly at the same time, every time a user presses one of the two buttons, and then falls into the ADD or REMOVE state, the system checks if the two buttons are actually being pressed. If that is the case, the state is immediately changed into RESET state.

Requesting Measurements

The measurement requests are triggered by a timer of 60 seconds. This timer runs independently from the main state machine of the system. Every time the timer reaches 60 seconds, the timer calls a subroutine that collects the identifications of the *communication modules* controlled by the *collection point* and sends a multicast message that instructs the first to get out of sleep mode and collect a value from the associated *measurement module*. After completing the request transmission, the timer is restarted and the system falls into IDLE state. The system handles the responses from the *communication modules* in a independent routine from the system state machine, referred by the Z-Wave software specifications as the Application Command Handler, which starts its action whenever a Z-Wave command is received [66]. Whenever a measurement command frame is received (see section 4.1.2.4 for more details), the *collection point* retrieves the source node identification and the measurement value from it. It then initiates a serial communication over its UART, in order to share that same information with the *gateway* application. It is important to notice that while the measurement requests are sent as a multicast message to the various *communication modules*, the request responses are independently handled by the *collection point*.

4.1.2.4 Application Layer Frame

The last two sections were exclusively dedicated to Z-Wave application software implementations of the *communication module* and *collection point*. However, since the *Home* wireless network is being used for the transmission of specific data formats, some work had to be done around the application layer of the Z-Wave protocol stack.

As discussed in section 4.1.1.6, the *electricity meter's measuring module* stores the accumulated electricity consumption values in a 64 bit variable [79]. The Z-Wave application running in the *communication module* can retrieve the value of the variable by sending a command through the serial connection that interfaces it with the *measuring module*. If the latter correctly interprets

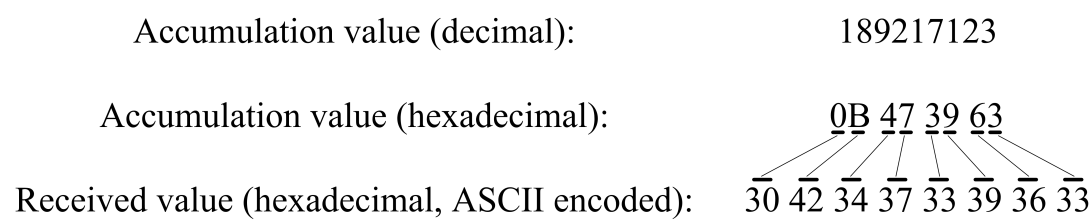


Figure 4.9: Explanation of the data format changes that occur during the exchange of data between the *measuring* and *communication* modules

the command, it sends back the variable's value. The *communication module* then receives a data block of 16 bytes, not 8 bytes (64 bits) as one would expect. Each received byte is ASCII encoded and is numerical digit of an hexadecimal representation of a single byte from the 64 bit (8 byte) variable. Figure 4.9 shows a practical and more explicit example of the problem. If no further data processing effort is made at the *communication module*, the Z-Wave application frame to be transmitted has to support a payload size of 16 bytes. The basic Z-Wave REPORT frames support no more than 1 byte of payload data. A new custom command class (see section 2.3.3 for details about Z-Wave command classes) had to be specified in order to support a 16 byte payload size. Table 4.4 shows the details about the specified commands and frames.

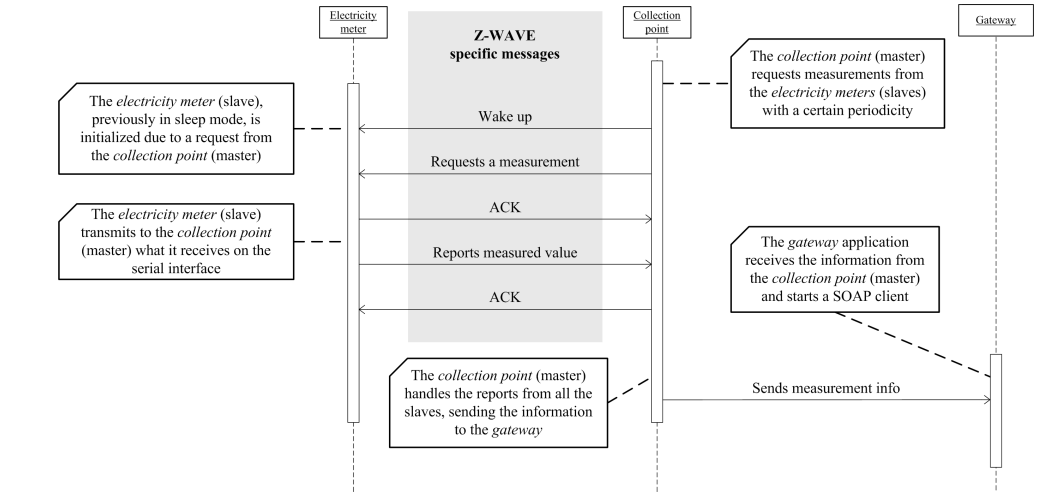
4.1.2.5 Communication Scheme

Given the descriptions of the Z-Wave application software and the work developed around the application layer of the Z-Wave protocol stack, it is now possible to build and understand a scheme of the different communication messages exchanged in the *Home* wireless network. Figures 4.10(a) and 4.10(b) show the communication profiles for different network operations in the form of sequence diagrams.

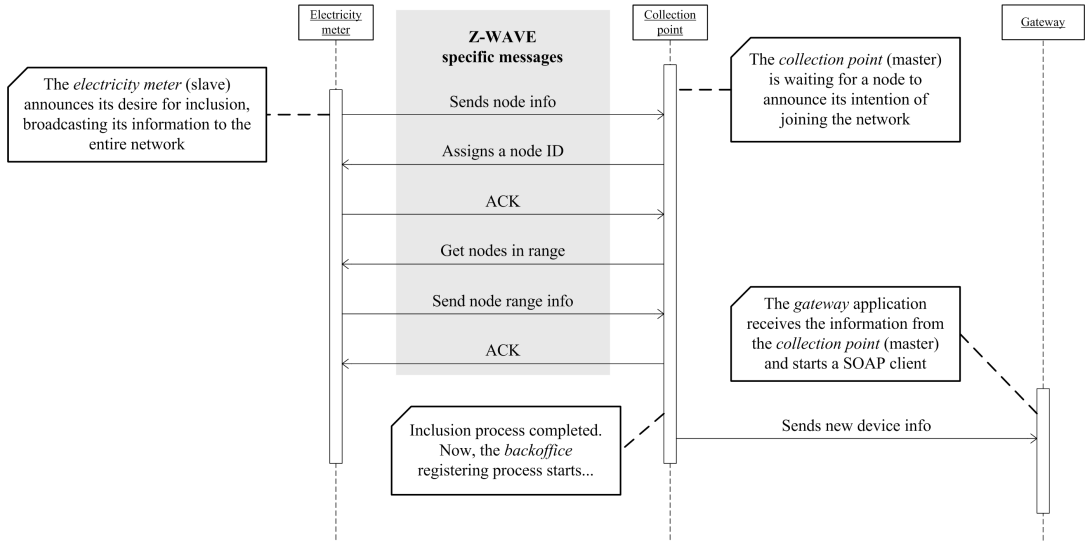
The diagram shown in figure 4.10(a) documents the normal operation of the Z-Wave network, i.e., the sequence of communications started when the *collection point* (master) requests a measure from a *communication module* (slave).

Figure 4.10(b) documents the sequence of exchanged messages when a new node is added to the network. Note the communication of the new node information to the *gateway*, in order to register the new *electricity meter* in the *Backoffice*.

This concludes the implementation description of the *Home* wireless network. It should be noticed that the work was centred in software rather than hardware development. However, the hardware modules to be developed in the future can be based on simplified versions of the RF units provided by the used Z-Wave development kit. The software development consisted in the programming of the MCU cores of the Z-Wave modules provided by the Zensys development tools. The latter were programmed in C language [90] and the used development platform was the uVision 3 Integrated Development Environment, distributed by Keil Software [91].



(a) Communication scheme during the normal operation of the wireless network (master requests and slave responses)



(b) Communication scheme during the addition of a new node to the wireless network

Figure 4.10: Communication scheme during the normal operation of the wireless network (a) and the addition of a new node to the wireless network (b)

Table 4.4: Details of the commands and frames specified by the new EMA command class

Designation	Description
EMA Get	Command used to request a measurement value from a *communication module*.
EMA Report	Command used by the *communication module* as a response to a measurement value request.
EMA Get Frame	The frame reserves 2 bytes to specify the command class it belongs to (EMA command class) and the command that the frame is supposed to carry (EMA Get).
EMA Report Frame	Besides the 2 bytes that specify the command class and carried command, this frame type reserves a 16 byte array for carrying measurement values.

4.1.3 Gateway Application

The *gateway* acts as an interface between the *Home* environment and the Internet, running an application that consists of two main parts: one responsible for receiving data from the *collection point* and another one that is supposed to send that same data to the *Backoffice*, via Internet.

The exchange of data with the *collection point* is accomplished with serial communication, specifically the RS-232 standard [33], a suitable solution due to the characteristics of the *collection point* hardware (section 4.1.2.1). The second part consists in a SOAP client. The implementation of the two main parts of the application is discussed over the next sections.

4.1.3.1 Serial Communication

The activity diagram shown in figure 4.11 describes the *gateway* application section responsible for receiving data in RS-232 format. The program consists of an event-driven serial port reader [92]. The main process starts by simply opening the COM port [92] on which the serial data is supposed to be received. The port is initialized with the following parameters:

- Character encoding: ASCII (American Standard Code for Information Interchange);
- Baud rate: 9600 bps (bits per second);
- No parity;
- 8 data bits;
- 1 stop bit.

If the initialization is successful, the main process will then wait for a COM port event, i.e., for a contiguous stream of serial data, terminated by the NULL character. When such an event occurs, an independent and parallel sub-process is created, responsible for storing the data stream into a buffer, verifying the validity of the received stream, and translating the ASCII encoded data into a suitable format to be used as arguments by the SOAP client. The SOAP client is called and,

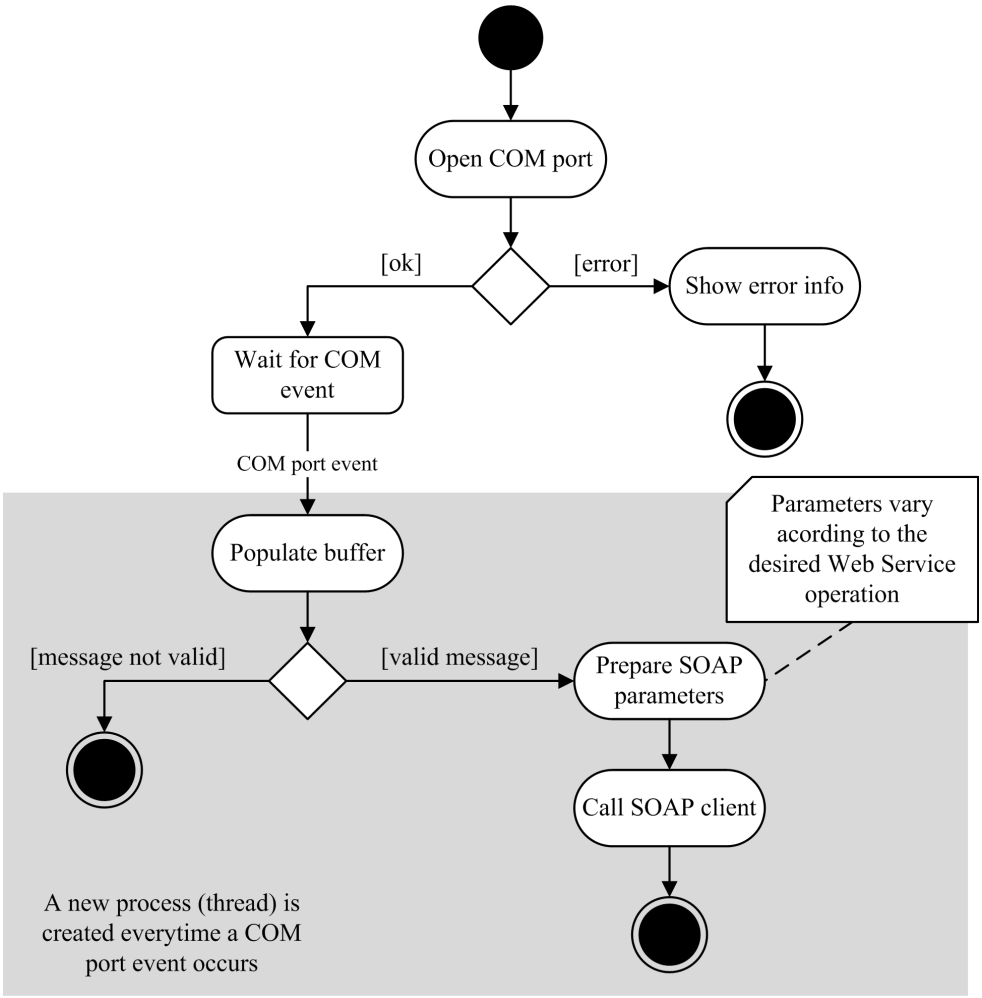


Figure 4.11: Description of the serial communication section for the *gateway* application

after the SOAP response is received, the sub-process is terminated. Meanwhile, the main process will keep listening for other COM port events.

4.1.3.2 SOAP Client

The second part of the *gateway* application consists in SOAP client that generates requests to be answered by the custom Web Service running at the *Backoffice*. The activity diagram on figure 4.12 shows the overall flow of the SOAP client operation. A SOAP request is generated by calling a method which takes the arguments listed below:

- URL that points the location of the Web Service;
- The *username & password* combination of the respective end-user of the system;
- Time and date of the request;

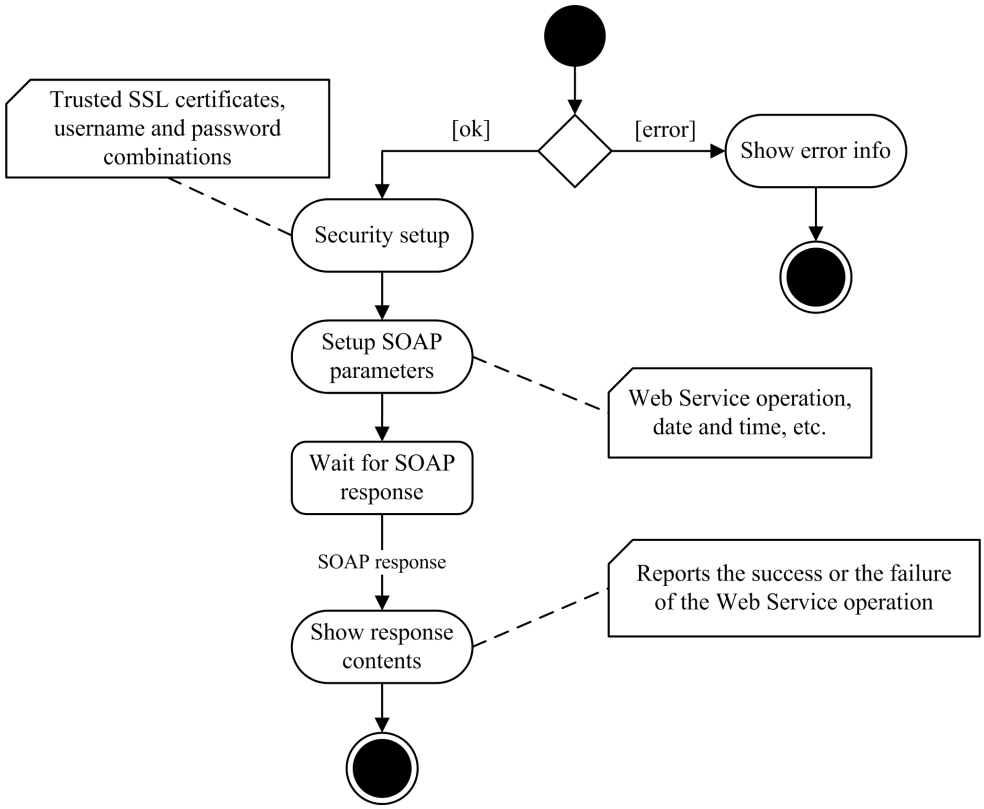


Figure 4.12: Activity diagram for the SOAP client section of the *gateway* application

- Desired operation to be executed by the Web Service (see section 4.2.1 for details about the available Web Service operations);
- Identification of the *electricity meter* that triggers the SOAP action and a measurement value (the SOAP request parameters).

If no initialization errors occur, the process starts with the SSL security set up. This includes the specification of the server certificates that should be trusted by the client, for SSL security, and other credentials, for client-side authentication (see section 3.3.2 for further details). A small stop should be made at this point to introduce some details about the implementation of the SSL security mechanisms.

The *gateway* application is developed in Java, and makes use of the Java Secure Socket Extension (JSSE) [93], a complete SSL API (Application Programming Interface) that allows SSL to be implemented in Java applications. JSSE defines a special type of files, *truststores*, which are databases of certificates that are used to verify the identities of other clients and servers [93]. During the setup of a SSL session, the developed Java application loads a specific *truststore* file which contains the certificates of the valid servers. This file is provided with the application and stored in the client machine during its installation process. After packaging the necessary arguments into a SOAP request, the application sends it to the Web Service URL. The process waits for a SOAP

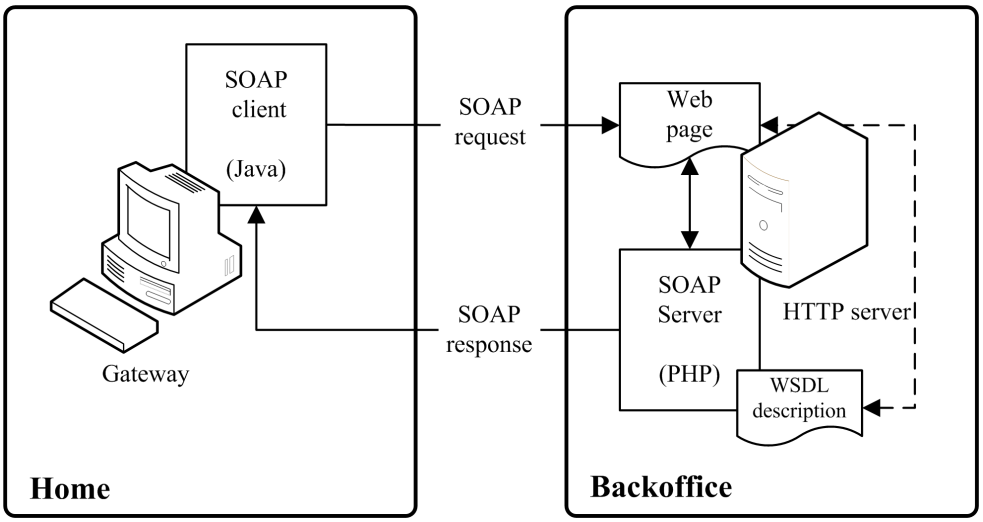


Figure 4.13: Description of the communications between the *Home* and *Backoffice* sides, with implementation details using NuSOAP

response and terminates after its reception.

On this section, the main implementation aspects of the gateway application were addressed. Both the serial communication and SOAP client parts were implemented in Java programming language [92, 93, 94]. Several external libraries were used, specifically RXTX [95] for the serial communication and Apache SOAP for Java [94] for the SOAP client. The gateway application is designed to be constantly running on a PC connected to the Internet. This fact goes against one of the reasons for the use of a remote location: why using a remote location to perform high-level data processing when the resources to do so are already present and constantly available at home? The ideal solution would be for the application to be running on another type of gateway, such as a router, which is normally in constant operation, however, due to the project's time frame, the implementation of a PC application appeared as a faster development option.

4.2 Backoffice Environment

This subchapter describes the implementation details of the remote communication infrastructure and the *Backoffice database*, following the specifications given in section 3.3.

4.2.1 Web Services and SOAP

The communication between the *Home* and *Backoffice* sides is based on Web Services, which uses the SOAP protocol. Figure 4.13 describes the communication method introducing its implementation details.

The *Home* side runs a SOAP client, developed in Java programming language [94]. When necessary, it builds up a SOAP request and sends it over HTTP to a custom Web Service, running

at the *Backoffice*. The Web Service is implemented in the form of a PHP script (Hypertext Preprocessor), using the NuSOAP library [96]. This library consists in a combination of PHP classes specially created to implement SOAP based Web Services [96]. When called, the PHP script generates a specific WSDL document that describes the custom Web Service as well as a SOAP server object that handles the SOAP request sent by the client application. The WSDL document specifies two Web Service operations:

- SET MEASURED VALUE: Upon this operation, a new measurement value, from a given *electricity meter*, is inserted into the *Backoffice database* (see section 3.3.3 for details about the *database* specification);
- SET NEW DEVICE: Inserts a new *electricity meter* entry into the *Backoffice* database. This operation is related to the automatic registration of an *electricity meter* in the system, described in section 3.1.

After the requested function is finished, the SOAP server packages the Web Service response into a SOAP message and sends it back to the SOAP client, over HTTP. Since the methods described above do not require specific return values, the content of the responses are simple notifications that indicate if the operations were concluded with or without errors.

This section was mostly focused on the implementation of the remote communication infrastructure on the remote side of the system, specifically the definitions of the custom Web Service. The SOAP client application runs on the *Home* side, being addressed further ahead in the chapter, in section 4.1.3.

4.2.2 Security in Remote Communications

The communication of data to the *Backoffice* is made through the Internet, which potentially introduces loss of confidentiality and integrity. The system attempts to solve the problem by the introduction of the SSL protocol. The security portion is implemented on the client side by the use of the Java SE Security API [93].

In this case, the client-side application is provided with a list of server certificates to be trusted. When sending a SOAP request to the server, the client sends it over HTTPS (Hypertext Transfer Protocol Secure), a combination of HTTP and SSL [54]. If, during the initial SSL handshake, the announced server certificate is listed on the client-side trust list, the transmission is carried on [93, 54]. This way the client is sure that it is communicating with the right server. As stated in section 4.1.3.2, this implies the installation of the server certificate on the client side.

4.2.3 Database

The implementation of the *database* consists in the choice of technologies to deploy the previously specified model and in performing simple configurations for it to be correctly accessed by

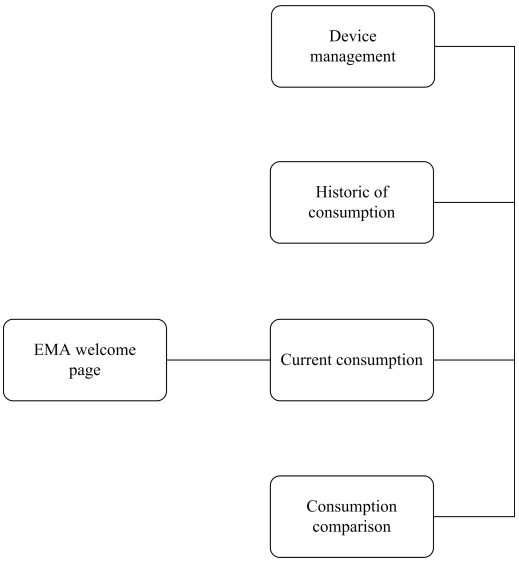


Figure 4.14: Final *website* structure specification

the system *User Interface*. In this case, the implementation phase does not add much more relevant information when compared to its specification. In order to store the measurement data in a structured way, a simple database has been developed, following the model shown on figure 3.5, section 3.3.3. It is based in SQL (Structured Query Language) [96] and is built on PostgreSQL, a free licensed ORDBMS (Object-Relational Database System) [96].

4.3 User Interface

The main graphical interface of the system is implemented as a *website* which that can be accessed by multiple devices equipped with Internet connections. Besides making it available virtually everywhere, the use of such devices as interaction tools have the potential for a complete user-friendly experience.

The implementation process of the system's *website* is described in this subchapter. First, a definition of its structure will be given, followed by a description of the implemented pages as well as of the technologies used in the implementation. The final section documents the implementation of a modified version (with reduced interactivity) of the *website* in a Wi-Fi enabled digital photo frame.

4.3.1 Website Structure

Although the *website* structure had already been defined in the specification phase (see section 3.4.2), a final scheme has been specified in the implementation phase, as shown in figure 4.14. Its building units (web pages) are briefly described in the following list:

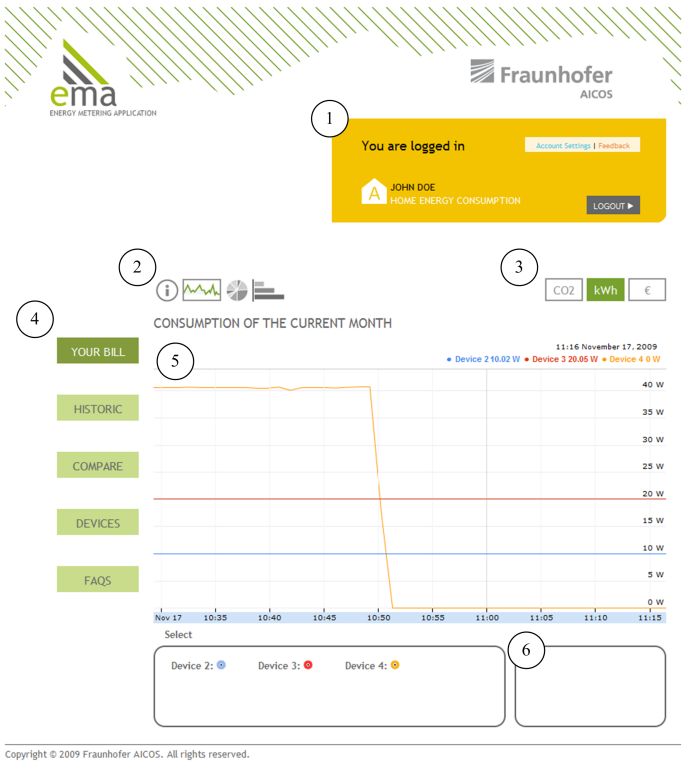
- Welcome page: The starting page of the *website*. It allows a user to login the system and access its information;
- Current consumption page: Shows the consumption of the current period (in this case, the current month). Information is shown in the form of different types of charts (bar, pie and line charts) and in three different units (monetary costs, equivalent CO_2 emissions and energy units). The details about the different kinds of charts and units are discussed in the next section;
- Historic of consumption page: Makes use of the same graphical mechanisms of the current consumption section to show information about past information;
- Compare page: Compares the users electricity consumption with national or regional averages. Despite being part of the final specification, this page has not been implemented;
- Devices page: Allows the user to manage its owned *electricity meters*, allowing them to verify when was it first added, associate a tag with it, among other operations.

4.3.2 Web Page Description

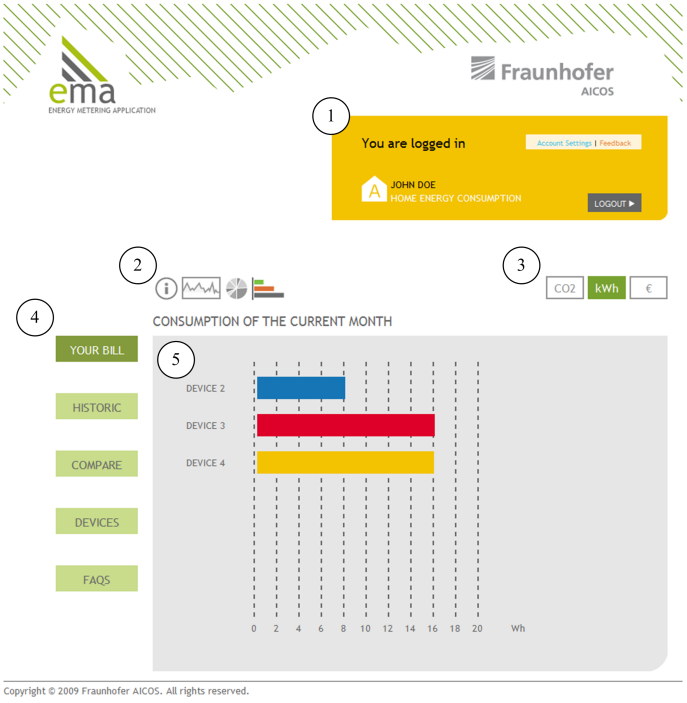
Although several pages have been defined in final specification given in section 4.3.1, only the basic visualization pages have been implemented, specifically the Home page and the Current Consumption page, since these pages alone can deliver the most basic visualization functionalities, and then complete the developed electricity metering system. Figure 4.15 depicts two views of the Current Consumption page. The following list names their contents according to the numbers shown in the figure:

1. Session box: Allows the end-user to check its session status and access personal information. The colour should change according to the efficiency that can be inferred from the collected consumption data. Although being part of the final specification, this particular functionality is not implemented;
2. Chart type icons: These icons allow the end-user to change the type of chart that displays the consumption information. There are three available chart types: an interactive[5] line chart that shows consumption over time (in W), a bar chart that displays information based on the energy accumulated by the different *electricity meters* (in Wh) and a pie chart, that shows relative consumptions between the *electricity meters* owned by the end-user;
3. Unit type icons: Three icons that allow the user to change the units in which the consumption data is displayed: monetary cost (€), energy units (W or Wh) and equivalent CO_2 emissions (kg/Wh);
4. Side menu: A section of buttons that allow the user to navigate through the different sections of the *website*;

[5]The line chart allows for the user to scroll along the time axis and zoom in a specific time window.



(a) Timeline chart



(b) Bar chart

Figure 4.15: Two views of the *website* current consumption pages: (a) timeline chart (b) bar chart

5. Chart area: The area that includes the actual chart to be displayed. Its size and additional features change from type to type. For example, the line chart type includes a selection box that allows the user to show or hide the consumption line of a specific device;

6. Alert windows: These containers show small text messages that inform the user about any abnormality or advice related to his consumption policies. Although being part of the final specification, the generation of such messages has not been implemented;

As stated in section 4.1.1.6, the electricity consumption values are stored in the system database in energy units (Wh), which has implications in the calculations of the values as presented by the *website*, specifically the current consumption page. The type and complexity of the calculations varies with the type of chart to be presented (bar or line), the order of magnitude of the values shown in the chart (mW, W, kW, etc.) and the unit type of those values (electrical energy/power, monetary cost and CO_2 emissions).

While the values shown in unit types that are power or energy related (W or Wh) can be displayed without performing unit conversion, those related to monetary cost or equivalent CO_2 emissions demand previous calculations. The display of data in CO_2 or monetary units is made by conversion of power or energy values based on given factors. These have been mentioned in section 3.4.2. The values of the factors are stored in the database[6], and are called when the webpage performs such a conversion.

The values also need to be displayed in an adequate way. First, there is the need to determine a suitable scale. The number of scale divisions is set to be always 10, with the determination of the scale division size based on the maximum value to be represented. Second, if the maximum value to be represented is 10^{-2} W, 10^{-2} kg or 10^{-2} €, it is more practical to convert it to 10 mW, 10 g or 1 cent. So, besides determining the scale division, the webpage is programmed to adapt its magnitude to achieve an easier representation

Finally, the difference between bar and line charts also implies different calculation efforts. While the values for the bar chart can be taken directly from the database (accumulated electric energy, as stored in the database), the line chart shows the consumption over time, i.e. power[7] (W) instead of energy (Wh). This is accomplished through a simple calculation, also performed by the webpage, using both energy and time differences between two consecutive measurements.

In order to build the *website*, several technologies were used, specifically languages such as HTML (Hypertext Mark-Up Language) [97], CSS (Cascading Style Sheets) [97], PHP and JavaScript [98]. The line chart, previously described above, makes use of the Google Visualization API, specifically the Annotated Time Line type [99]. This type of resource consists in a JavaScript visualization that is rendered within the browser using Flash [99].

[6] The factor values require frequent updates. An automatic update of these values has not been implemented.
[7] It can also show the rate of CO_2 emissions or cost over time (kg/hour or €/hour).

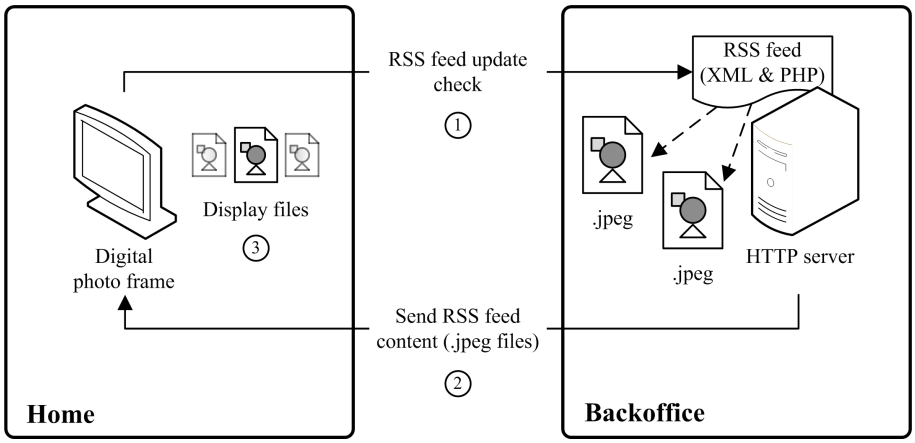


Figure 4.16: Description of the communication scheme between the digital photo frame and the *Backoffice*

4.3.3 Digital Photo Frame

In addition to the *website* which can be accessed via a Internet-enabled PC or mobile device (see figure 3.1), another form of user interaction was implemented, accessible through a digital photo frame equipped with a IEEE 802.11 b/g wireless connection.

The use of a wireless digital photo frame allows the user to quickly verify the his electricity consumption, without having to actively log in to the *website*. The information provided by the digital photo frame is based on the same contents provided by the *website* (same unit types and graphical appearance). However, since the frame is mostly 'passive', i.e. does not allow the same level of user interaction provided by a PC or mobile device, a special application had to be developed for the digital photo frame.

As described in figure 4.16, the digital photo frame acts as an RSS reader (Really Simple Syndication) [100] which frequently checks for updates of an RSS document, an XML file, hosted by the HTTP server located at the *Backoffice*. The RSS document contains links to JPEG (Joint Photographic Experts Group) [101] images which consist in modified screenshots of the current consumption page of the *website*. At every update check, i.e every time the RSS feed URI is called, a PHP script, described by the diagram shown in figure 4.17, creates JPEG pictures optimized for presentation on the 800×600 resolution digital photo frame (the final result is shown on section 5.1.4). The JPEG images are then transferred and displayed in the photo frame.

Figure 4.17 shows how the image of a chart for a photo frame is built by the PHP script as it execution progresses. The first part of the script (data section) is similar to those used by the *website* to build the consumption pages, specifically the definition of the measurement axis scale (the horizontal axis for the bar chart, the vertical for the line chart). The difference is the addition of the time scale definition on the line chart, since its construction is not handled by the Google Visualization API [99] in this case:

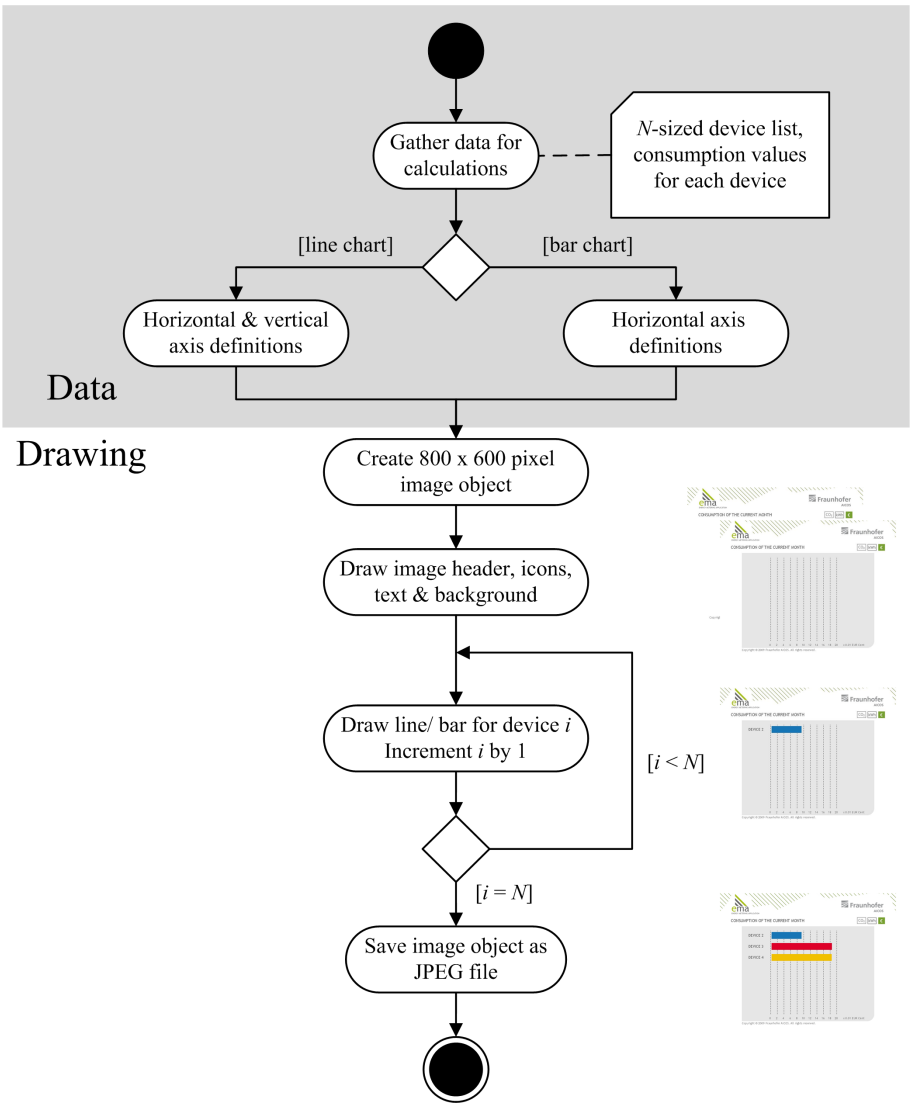


Figure 4.17: Description of the PHP script responsible for generating the images to be shown at the digital photo frame

1. For easy visualization on the digital photo frame, the maximum number of time intervals has been set to 12;

2. The script then chooses an adequate time scale interval based on the time difference, T_D, between the first and last measurements to be represented. The scale interval is chosen from a set of pre-defined values, T_I^i, (1 min, 5 min, 15 min, 30 min, 60 min, and others) as soon as $T_I^i \times 12 \geq T_D$;

3. The times of the first and last measurements are then rounded so that the values shown on the scale are multiples of T_I away from the start of a given hour. For example, if the first measurement was made at 11:11, for $T_I = 15$ min, the values on the scale should be 11:15, 11:30, 11:45, etc. Due to this fact, the lines of the chart start/end slightly before/after the scale vertical lines, as verified on figure 5.8(c).

After these steps, the script creates a 800×600 pixel image object, builds the header, the text as well as the actual chart, and finally saves the image object as a JPEG file. The graphical appearance of the photo frame images is shown in figure 5.8.

The wireless digital photo frame used in the project was the Samsung SPF-83V [102]. As previously said, the images are requested/sent to the photo frame using a RSS feed mechanism. The RSS feed URI is simply introduced in the digital photo frame, using its specific configuration interface. In order to create the JPEG images, PHP scripts are employed, using special methods from the GD Graphics Library [103].

4.4 Summary

This chapter documented the implementation details of the developed electricity metering system. The structure of this chapter is similar to the one used for chapter 3 in order to allow a direct comparison between specification and implementation. For each one of the system components the methodologies and technologies used in the implementation process have been identified.

The main difference between the specification and implementation phases is related to the *electricity meters*. Although a final schematic for the implementation of a PCB that would combine the *measuring* and *communication* modules was achieved, the physical implementation of such solution could not be completed. However, a physical implementation of an *electricity meter* prototype was accomplished by using the development tools provided by Teridian and Zensys, the manufactures of the electricity metering IC and Z-Wave transceiver, respectively. This allowed the effective creation of a functional prototype of the complete wireless metering system.

Chapter 5

Evaluation and Validation

After the development phase of the project, a functional system has been obtained, involving several sections, each one built around different types of technologies. When put together, the latter intend to deliver the functionalities of an electricity metering system. The existence of multiple system blocks also introduces multiple points of possible failure, so there is the need for a distributed evaluation of the system's performance.

In this chapter, the performance of the developed system is analyzed and validated. The first section introduces the subchapter by explaining evaluation methodologies. The following sections present the results of each of the performed tests. Finally, and since it is a relevant issue for this project, a brief evaluation of manufacturing and production costs is given.

5.1 Evaluation Scenario

The system was subjected to a series of tests that simulate its normal operation, i.e., the use cases it has been designed to cover. Those will consist in the electricity monitoring of multiple electrical appliances made by several *electricity meters* and the correct visualization of their output in the *website*. Other functionalities involve the quick addition of new *electricity meters*, as explained in section 4.1.2.3. While performing these operations, the performance of each individual block of the system is analyzed.

The following test scenario has been set up:

- 4 *communication modules* (in the form of slave hardware units from the Z-Wave development kit), spread all over a specific household configuration (see figure 5.1), one of them connected to a *measuring module*;
- The *measuring module* together with a *communication module* make up an *electricity meter*, as explained in section 4.1.1. It is metering the electricity consumption of a simple lamp, equipped with a incandescent light bulb of 40 W;

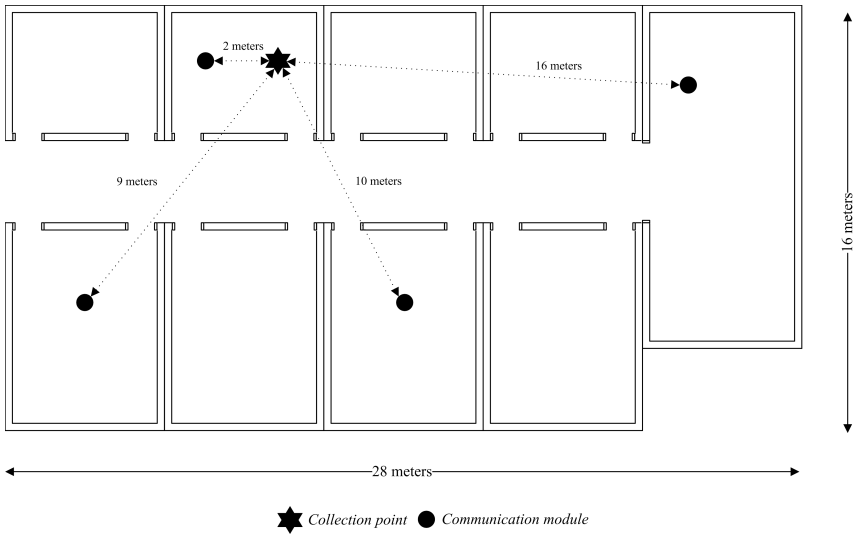


Figure 5.1: Disposition of the modules during the test sessions

- 1 *collection point* connected trough an RS-232 cable to a serial port of a PC running the *gateway* application;
- 1 remote HTTP server running a Web Service and a *website* that constitutes the main visualization interface of the system.

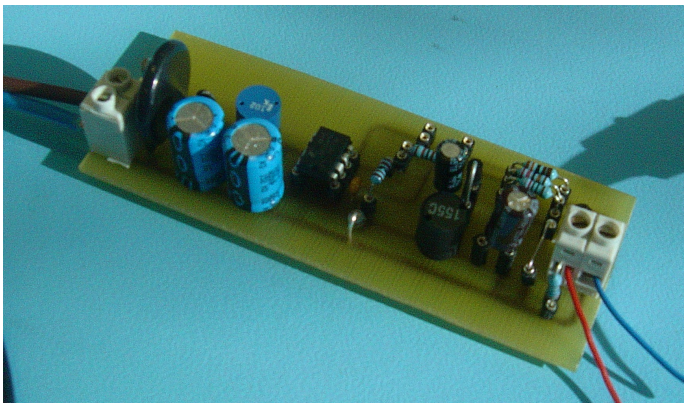
First, the analysis of the issues related to the *electricity meters* are presented, followed by the remaining components of the *Home* environment. After that, the next proof-of-concept process concerns the security in the communications between the *Home* and *Backoffice* sides. The last aspect to be verified is the display of consumption data in the *User Interface*.

5.1.1 Electricity Meter

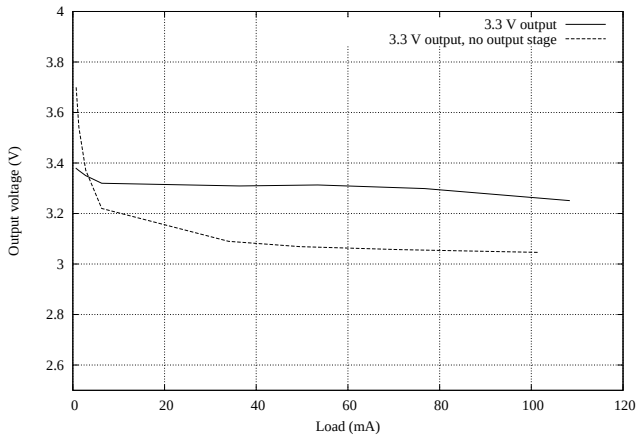
The first important issue to notice about the *electricity meter* is the existence of only one *measuring module*, implemented by the only available Teridian 71M6511 Demonstration Board, as documented in section 4.1.1.6. Although a final hardware implementation of the *electricity meter* has not been achieved, the specification plans were completed and can be consulted in section 4.1.1.5. In the global test, the other *electricity meters* are replaced by slave hardware models provided by the Zensys Z-Wave development kit (see section 4.1.2.1), that generate and transmit fictitious measurements.

5.1.1.1 Switched-Mode Power Supply

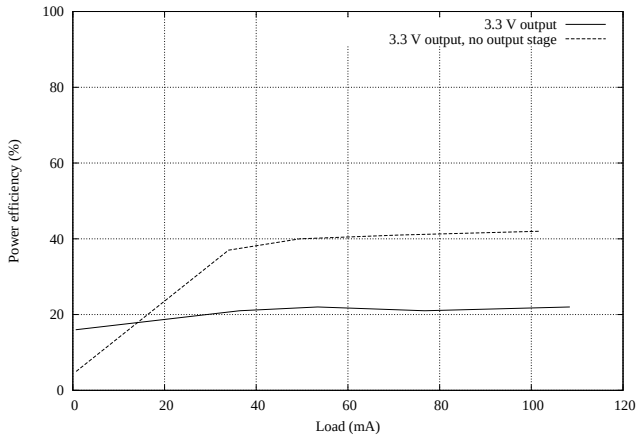
In this section the SMPS shown in figure 4.4 is evaluated. A PCB prototype of the final design has been created to conduct performance tests , as shown in figure 5.2(a) (see appendix B for details on the schematic and PCB layout). The results are shown on figures 5.2 and 5.3.



(a) PCB prototype of the SMPS design shown in figure 4.4

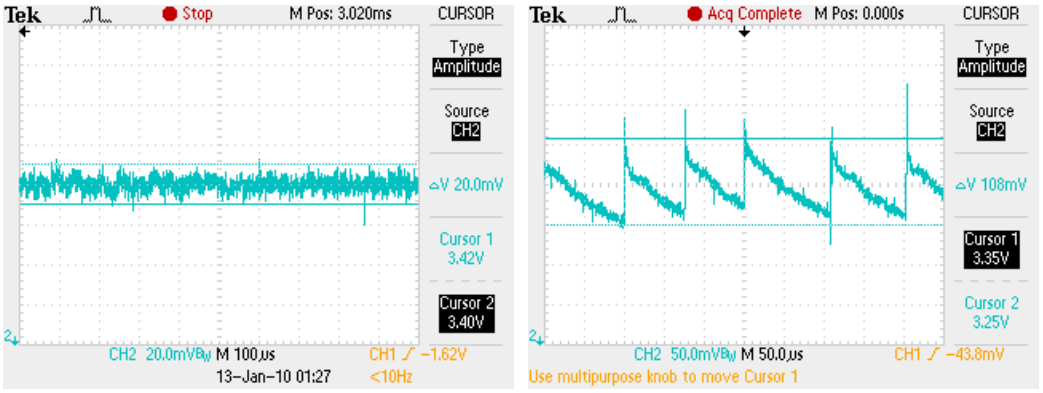


(b) Output voltage regulation vs. output load current



(c) Power efficiency of the SMPS design vs. output load current

Figure 5.2: SMPS evaluation results



(a) Output voltage profile of the output stage design (b) Output voltage profile of the design without an output stage

Figure 5.3: SMPS evaluation results (cont.)

The graph on figure 5.2(b) shows how the output voltage (3.3 V) varies as the load current changes between several values, including those shown on table 4.2(b). The SMPS shown on figure 4.4 is compared with a similar design, only without the output stage composed by T_1, Z_1 and resistors from R_3 to R_7[1]. The chart shows how the regulation of V_{out} has benefited from the introduction of the output stage, even for low current values, due to the use of a 2% tolerance Zener diode and its emitter follower configuration.

Figure 5.2(c) compares the variation of efficiency (P_{out}/P_{in}) with load current between the designs with/without an output stage. The reduction of efficiency caused by the output stage is evident (40% vs. 20% for current values over 40 mA), since the SMPS actually transforms the 220 VAC input to 12 V DC that supply the emitter follower stage. 10 mA are also used to bias the Zener diode, Z_1. The joint analysis of figures 5.2(b) and 5.2(c) identifies a tradeoff between regulation/efficiency.

Finally, figure 5.3 and 5.3(b) show fast variations of output voltage, collected on a oscilloscope. Figure 5.3(b) refers to the design with an output stage (load current of 76.72 mA), while figure 5.3(b) refers to a design with no output stage (load current of 1.31 mA, in order to achieve a output voltage, V_{out}, of approximately 3.3 V). In the first case, the amplitude of the fast variations is near 20 mV as opposed to the 100 mV of the second case. Therefore, the use of the output stage results in better regulation as well as in a reduction of fast output voltage variations at the expense of a lower power efficiency. Both figures present voltage spikes that occur at the switching instants of the LNK304PN. This is usual in SMPSs, due to fast current changes at the switching instants. This issue is relevant for this project, since the *electricity meter* includes the use of ICs which are sensitive to supply voltage fluctuations. The results of this test seem to indicate an additional design effort in order to make the SMPS a suitable solution for the *electricity meter*. However, no real conclusions can be made since a final implementation of the *electricity meter* could not be

[1] In order to achieve a 3.3 V output for the design without an output stage, the resistor R_2 had to be changed to 2.1 kΩ (see section 4.1.1.3 for details on the determination of this value).

Time	Speed	Delta	Source	Dest.	HomeId	Data	Application
25-13:52:42	9.6Kbit/s	992	001	255	EBF2D139	BroadCast	Transfer Presentation
25-13:52:42	40Kbit/s	16	000	001	EEEF4095	SingleCast	Node Info
25-13:52:42	9.6Kbit/s	19	001	000	EEEF4095	SingleCast	Assign Ids
25-13:52:42	9.6Kbit/s	27	000	001	EEEF4095	Ack	
25-13:52:42	9.6Kbit/s	21	001	005	EBF2D139	SingleCast	NOP
25-13:52:42	9.6Kbit/s	21	005	001	EBF2D139	Ack	
25-13:52:42	40Kbit/s	31	001	005	EBF2D139	SingleCast	Find Nodes In Range
25-13:52:42	40Kbit/s	7	005	001	EBF2D139	Ack	
25-13:52:42	40Kbit/s	7		2		WakeUpBeam	
25-13:52:43	40Kbit/s	1150		2		WakeUpBeam.End (251)	
25-13:52:43	40Kbit/s	2	005	002	EBF2D139	SingleCast	NOP Power
25-13:52:43	40Kbit/s	7	002	005	EBF2D139	Ack	
25-13:52:43	40Kbit/s	9		3		WakeUpBeam	
25-13:52:44	40Kbit/s	1150		3		WakeUpBeam.End (251)	
25-13:52:44	40Kbit/s	2	005	003	EBF2D139	SingleCast	NOP Power
25-13:52:44	40Kbit/s	7	003	005	EBF2D139	Ack	
25-13:52:44	40Kbit/s	9		4		WakeUpBeam	
25-13:52:45	40Kbit/s	1150		4		WakeUpBeam.End (251)	
25-13:52:45	40Kbit/s	2	005	004	EBF2D139	SingleCast	NOP Power
25-13:52:45	40Kbit/s	8	004	005	EBF2D139	Ack	
25-13:52:45	40Kbit/s	6	005	001	EBF2D139	SingleCast	Command Complete
25-13:52:45	40Kbit/s	7	001	005	EBF2D139	Ack	
25-13:52:45	40Kbit/s	5	001	005	EBF2D139	SingleCast	Get Nodes In Range
25-13:52:45	40Kbit/s	7	005	001	EBF2D139	Ack	
25-13:52:45	40Kbit/s	5	005	001	EBF2D139	SingleCast	Node Range Info
25-13:52:45	40Kbit/s	7	001	005	EBF2D139	Ack	

HomeID: EEEF4095 SrcNodeID: 001 DstNodeID: 000
Header Type: WakeUp Beam Speed: none SingleCast Request ACK Sequence Number: 7 Length: 17 bytes
Application Layer: Assign Ids
New node ID : 5
New Home ID : 0xEBF2D139

(a) Z-Wave log

id_device	costumer	designation	entrydate	entrytime
2	1	Device 2	2009-07-25	13:52:33.172562
3	1	Device 3	2009-07-25	13:52:34.350924
4	1	Device 4	2009-07-25	13:52:41.24962
5	1	Device 5	2009-07-25	13:52:48.977707

(b) Confirmation of the node addition in the system database

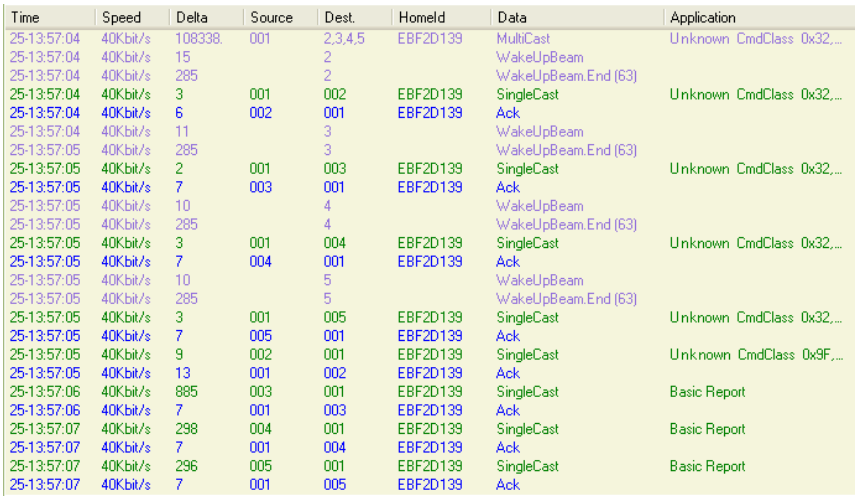
Figure 5.4: Messages exchanged within the *Home* wireless network during the node addition tests

made, due to the project's time constraints. Nevertheless, the study of different power supply types provided knowledge about the design of such kind of hardware parts, specifically transformerless power supplies.

The results shown in this section fit into the parameters defined in table 4.2(b), which indicates a successful design of the SMPS for the *electricity meter*.

5.1.2 Home Wireless Network

As mentioned in section 5.1, the four *electricity meters* used in the test sessions were positioned on different divisions of a specific indoor configuration, as depicted in figure 5.1. The objective of this test is to evaluate the reliability of the Z-Wave wireless network by capturing the exchanged data between the *communication modules* and the *collection point* over a period of time and then analyse the communication logs. The data logs have been obtained through a set of Z-Wave network scanning tools provided by the Z-Wave development kit [89].Two types of network operations were tested, the addition of a new node and the normal electricity monitoring scheme.



Time	Speed	Delta	Source	Dest.	HomeId	Data	Application
25-13:57:04	40Kbit/s	108338.	001	2,3,4,5	EBF2D139	MultiCast	Unknown CmdClass 0x32,...
25-13:57:04	40Kbit/s	15		2		WakeUpBeam	
25-13:57:04	40Kbit/s	285		2		WakeUpBeam.End (63)	
25-13:57:04	40Kbit/s	3	001	002	EBF2D139	SingleCast	Unknown CmdClass 0x32,...
25-13:57:04	40Kbit/s	6	002	001	EBF2D139	Ack	
25-13:57:04	40Kbit/s	11		3		WakeUpBeam	
25-13:57:05	40Kbit/s	285		3		WakeUpBeam.End (63)	
25-13:57:05	40Kbit/s	2	001	003	EBF2D139	SingleCast	Unknown CmdClass 0x32,...
25-13:57:05	40Kbit/s	7	003	001	EBF2D139	Ack	
25-13:57:05	40Kbit/s	10		4		WakeUpBeam	
25-13:57:05	40Kbit/s	285		4		WakeUpBeam.End (63)	
25-13:57:05	40Kbit/s	3	001	004	EBF2D139	SingleCast	Unknown CmdClass 0x32,...
25-13:57:05	40Kbit/s	7	004	001	EBF2D139	Ack	
25-13:57:05	40Kbit/s	10		5		WakeUpBeam	
25-13:57:05	40Kbit/s	285		5		WakeUpBeam.End (63)	
25-13:57:05	40Kbit/s	3	001	005	EBF2D139	SingleCast	Unknown CmdClass 0x32,...
25-13:57:05	40Kbit/s	7	005	001	EBF2D139	Ack	
25-13:57:05	40Kbit/s	9	002	001	EBF2D139	SingleCast	Unknown CmdClass 0x9F,...
25-13:57:05	40Kbit/s	13	001	002	EBF2D139	Ack	
25-13:57:06	40Kbit/s	885	003	001	EBF2D139	SingleCast	Basic Report
25-13:57:06	40Kbit/s	7	001	003	EBF2D139	Ack	
25-13:57:07	40Kbit/s	298	004	001	EBF2D139	SingleCast	Basic Report
25-13:57:07	40Kbit/s	7	001	004	EBF2D139	Ack	
25-13:57:07	40Kbit/s	296	005	001	EBF2D139	SingleCast	Basic Report
25-13:57:07	40Kbit/s	7	001	005	EBF2D139	Ack	

Figure 5.5: Message exchanged within the *Home* wireless network during the normal operation tests

Figure 5.4(a) shows a portion of a communication log which presents the exchanged Z-Wave frames during the addition of a new node. The communication scheme as presented in figure 4.10(b) can be confirmed by inspection of the log. Each line of figure 5.4(a) corresponds to an exchanged frame, and its parameters are located along the top. The most relevant parameters for this explanation are:

- Source: The identification of the node which sent the given frame;
- Dest.: The identification of the node to which the frame has been sent;
- Data: Transfer layer information for the given frame (see section 2.3.3.1;
- Application: Application layer information (see section 2.3.3.1).

The process starts with *collection point* (node 001) entering into learn mode, i.e., waiting for a new node to announce its intention of joining the network. That happens when the user presses one of the interface buttons in the *collection point's* hardware module (see section 4.1.2.3). The new node sends a frame with its information (Node Info), still without an assigned identification (the source value is 000). The *collection point* then assigns an identification to the *communication module* (node 005, in this case) and requests it to find its neighbour nodes. The *collection point* sends wake up beams to the *communication modules* that are already included in the network, allowing the recent added node to exchange information with them. Figure 5.4(b) depicts the information about the added *electricity meters* in the *Backoffice* database. It can be seen that the *electricity meter* under inspection, device 5, has been added to the database soon after the Z-Wave inclusion process has been completed.

A portion of the log which documents the normal operation of the wireless network is depicted in figure 5.5. Once again, the communication scheme presented in figure 4.10(a) can be confirmed.

From inspection of the log it can be noticed that the *collection point* sends a multicast wakeup call to all the *communication modules* in the network before making the actual measurement requests, spaced by 2 minutes between each other. In figure 5.5 these request frames carry a specific Application layer information, specifically the 0x32 command class, which corresponds to the EMA Get command class defined in section 4.1.2.4. After the request, each of the *communication modules* then responds with a measurement, finishing a simple cycle of the normal wireless network operation.

No Z-Wave frames have been lost during approximately 4 hours of continuous operation of the system. All of the measurement data has been successfully stored in the database, which means it has also been successfully transmitted to the *gateway* application. The tests subjected to the Z-Wave wireless network portion validated its reliability while performing the most basic use cases that were specified for the system.

5.1.3 Security in Remote Communications

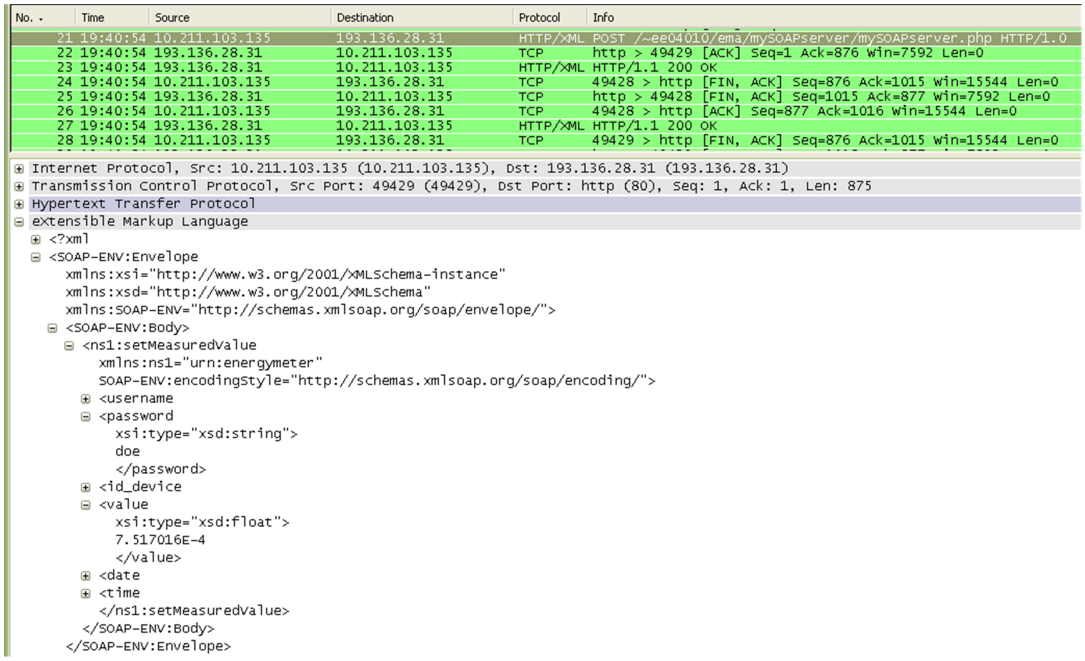
As specified in section 3.3.2, the system makes use of security mechanisms to provide client and server side authentication, data confidentiality and data integrity in the communications between the *Home* and *Backoffice* blocks, made over the Internet. In this section the exchange of information is analysed in order to show the security features of the system claimed in section 3.3.2.

Figure 5.6 shows the Wireshark [104] logs relative to the communication of SOAP messages between the *Home* and *Backoffice* sides, with IP addresses 10.211.103.135 and 193.136.28.31. Figure 5.6(a) shows a SOAP request without employing security mechanisms while in 5.6(b) the SSL protocol is used. It can be seen that in 5.6(a) the SOAP request parameters are easily spotted due to the lack of data encryption. However, in 5.6(b), the SSL protocol is used, and the client and server sides start to establish an SSL session. After exchanging the session keys [54], the two sides exchange data in a secure way.

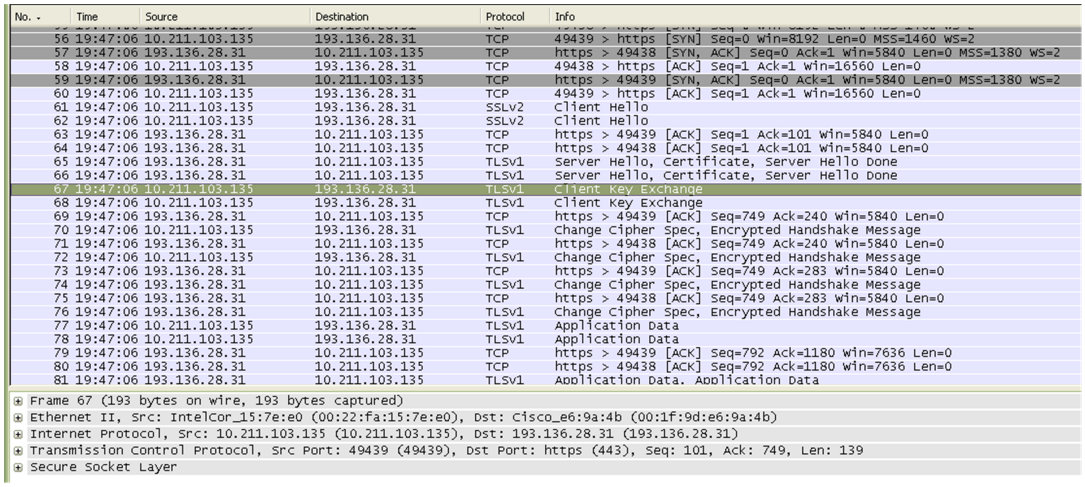
With the evaluation of the security of the communications made over the Internet, the next phase will evaluate the visualization of data in the *website*, the main graphical interface of the system.

5.1.4 User Interface

The visualization interface of the system is implemented as a *website* and another version of it, specially modified to be shown on a digital photo frame. The final specification and implementation details for both have been given in sections 4.3.1 and 4.3.3, however not all of the parts of the defined structure and functionalities could be implemented due to time constraints.



(a) No security mechanism



(b) With security mechanism (SSL)

Figure 5.6: Wireshark logs for the evaluation of the security in remote communications

5.1.4.1 Website

The implementation of the *website* was confined to simple visualization functionalities, as depicted in figure 5.7, which shows some screenshots from the *website*, taken during the test scenario.

Figure 5.7(a) shows the accumulated energy consumption calculated by each of the four *electricity meters* in the form of a bar chart. This screenshot was taken after approximately 1 hour of operation and it can be seen that device 2, the one that is monitoring the lamp with a 40 W light bulb, indicates approximately 40 Wh. This is a reasonable value since the lamp had been in constant operation, drawing a power of approximately 40 W for 1 hour, which according to the formula for energy, $E = P \times \Delta t$, leads to 40 Wh of energy consumption. Figure 5.7(b) shows the instantaneous power consumptions readings in the form of a time-line chart. The line chart displays the results taken over a section of time of the whole test. It can be seen that device 2, the *electricity meter* that is monitoring the lamp equipped with a light bulb of 40 W, shows a instantaneous consumption value of 39.85 W. The actual RMS current drawn by the lamp has been measured with a multimeter to be approximately 175 mA. Following the formula for average power, $P_{avg} = V_{RMS} \times I_{RMS}$, considering V_{RMS} to be 230 V [105], it results in an instantaneous power of 40.25 W. The error is then in the order of 1.0%. This error may be related to calibration issues of the Teridian 71M6511 Demo Board. However, a deep study and evaluation of the measurement precision is not an objective of this project.

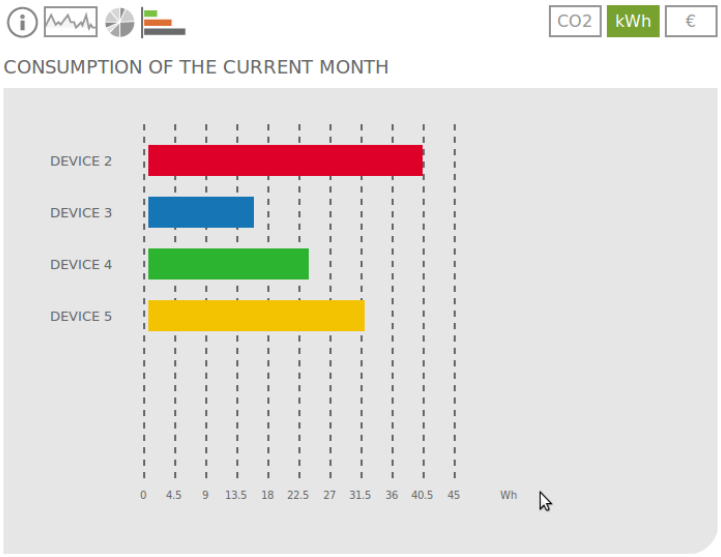
5.1.4.2 Digital Photo Frame

As stated in section 4.3.3, the images displayed in the digital photo frame are a modified version of the *website*, specially modified to fit the low interaction and 800×600 resolution of the Samsung SPF-83V [102] digital photo frame. Figure 5.8(a) shows an actual digital photo frame displaying consumption information. The photo frame simply displays a sequence of bar and line charts. Specifically, the displayed charts are:

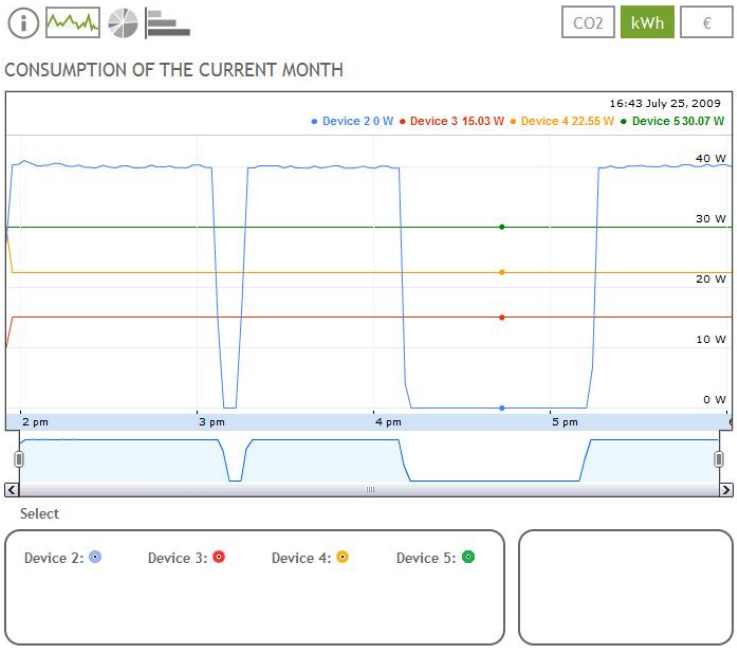
- 3 bar charts, displaying information in three different unit types: energy, monetary cost and CO_2 emssions;
- 3 line charts, displaying information in the same unit types as above;

Figures 5.8(b) and 5.8(c) show examples of JPEG images for a bar and a line chart, respectively. These charts are similar to those presented in the *website*, the differences are mainly related to the method used for image creation (see section 4.3.3), specifically in the line chart case. The photo frame evaluation has been performed at a different time than the test procedure for the *website*, documented in section 5.1.4.1, therefore a direct comparison between figures 5.8 and 5.7 is misleading. Appendix C provides *website* screenshots taken at the same occasion as those in figure 5.8.

The use of a digital photo frame enables a new use case, in which energy consumption can be consulted without requiring the user to log in to the *website*. The user can simply turn on the



(a) Bar chart

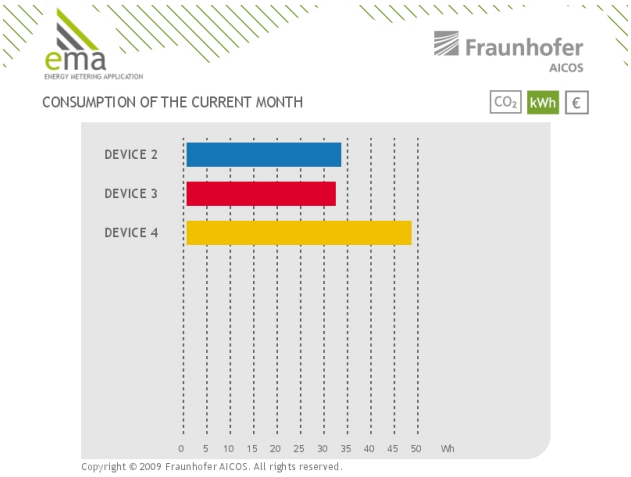


(b) Timeline chart

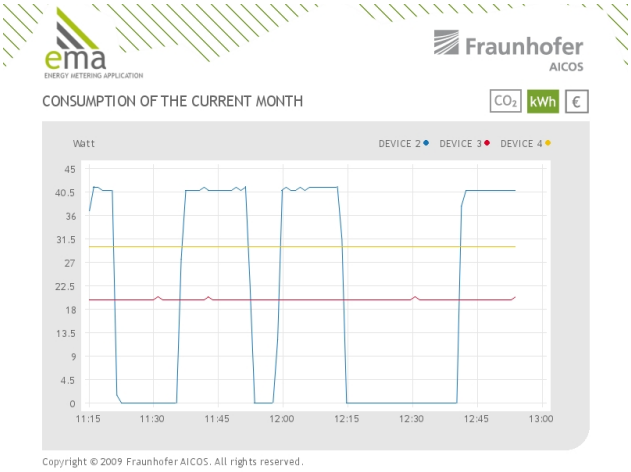
Figure 5.7: Screenshots of the *website*, the main graphical interface of the system: (a) bar chart (b) timeline chart



(a) Samsung SPF-83V digital photo frame



(b) Photo frame version of a bar chart



(c) Photo frame version of a line chart

Figure 5.8: Different perspectives over the digital photo frame application

photo frame, leave it always on, and check it from time to time, whenever the frame catches his attention.

5.2 Cost Evaluation

The low cost of the solution was an important objective for this project, and it was a factor taken into account in many of the decisions. The division of the system architecture in two separated environments, with the communication between them made via Internet is an example of such a decision. The system makes use of an infrastructure that already exists, since most of today's households are equipped with fast Internet connections, and transfers the processing effort from the user house to a single external point, avoiding the necessity of developing hardware or elaborated software to be locally installed. Hence, the main cost of the solution comes from the electricity meters.

The main focus of this section is the production cost evaluation of a single *electricity meter*, whose design was described in section 4.1.1. An estimate of its cost was calculated by adding the price of its main components, taken from major electronic part distributors, such as DigiKey [39], Farnell [41] and Mouser Electronics [40]. The prices are relative to high volume orders, i.e., orders over 1000 units, since this evaluation is focused on mass production costs. The estimate is shown in table 5.1. The total result, even not counting with development costs, is rather high to be considered a 'low-cost' solution. The RF transceiver alone represents approximately 30% of the total cost, and together with the power metering IC, that results in nearly 40%. This means that changes in the individual price may have a large impact on the final cost of such a product.

5.3 Summary

After finishing the test sessions and analysing its results, it can be concluded that a proof-of-concept electricity metering system prototype has been achieved. However some results indicate that some improvements still have to be made.

Firstly a physical implementation of an *electricity meter* could not be completed. The schematics for its completion are a result from this work, even though some improvements to the power supply performance may be required, as verified in section 5.1.1. In spite of these shortcomings, the work developed in this project provided a reliable wireless communications infrastructure to be used for data exchange between the metering devices, as seen in section 5.1.2. The objective of dividing the system into local and remote locations, providing reliable and secure communications, has also been achieved, as documented in section 5.1.3. Section 5.1.4 shows that the basic functionalities of the visual interface were implemented.

Finally, section 5.2 shows a brief compilation of the cost of one of the key components of the work developed for this project, the *electricity meter*.

Table 5.1: Cost of the main components included in the *electricity meter* design

Component	**Price**
Power metering IC	
Teridian 75M6511H-IGT/F	1,93 €
Power supply	
LNK304PN	0,38 €
Varistor 390 V, 6500 A	0,36 €
Inductor 1.1 mH	0,28 €
Inductor 1.5 mH	0,33 €
Capacitor 4.7 μF, 400 V (2 units)	0,28 €
High speed diode (BYV26C)	0,14 €
Zener BZX384-B3V9	0,03 €
BJT 2N3904	0,04 €
Subtotal	1,84 €
Current input circuit	
Current-sense transformer (CSE187-L)	1,74 €
Diode array (BAV99DW)	0,10 €
Subtotal	1,84 €
Other components (measuring module)	
Serial EEPROM (AT24C1024BN)	1,69 €
Connector for *communication module*	0,63 €
Crystal 32.768 kHz (XC488CT)	0,39 €
LED ($\leq$ 1 mA, 4 units)	0,32 €
LED driver (74LS541N)	0,36 €
Subtotal	3,39 €
Z-Wave transceiver IC (plus front-end)	
Zensys ZM3102N	5,00 €
Other components (communication module)	
Other components (estimate)	3,50 €
Total	**17,50 €**

Chapter 6

Conclusions

This chapter presents the main conclusions taken from the development of this particular electricity metering system, specifically the accomplishments of the work, a comparison with the initially specified objectives and the identification of new challenges raised by the developed work as well as potential future developments.

6.1 Achievements

The main achievement of this project is the creation of a proof-of-concept prototype of an electricity metering system and the acknowledgement of the multidisciplinary issues involved in the development of such a solution, including topics such as general hardware and firmware development, wireless sensor and computer networks, computer applications, web applications and even social and behavioural sciences.

6.1.1 General Achievements

The prototype satisfies most the base requirements specified in section 1.2, providing a modular approach for measurement of electricity consumption in buildings, offering user-friendly, platform-independent visualisation of data from any device with Internet access.

The architecture modularity provides easy change/extension of any system block without modifications on other parts. As an example, the use of another technology for the *home* network would not require changes in any other except the *communication modules* and the *collection point*. The same is valid for the addition of other energy-type *measurement modules* or a new Internet communication scheme. For example, if a scheme other than Web Services was to be used, the only change would be on the *gateway* application section that handles Internet communication (see section 4.1.3.2), while every other block could remain unmodified (see section 4.1.3.1). The process of addition of a new *electricity meter* provides ease of deployment, as the *communication modules* were designed to be automatically added to the system upon initialization, i.e., besides

being included in the wireless network, its information is also inserted in the system's *database* (see sections 4.1.2.2 and 4.1.2.5).

The electricity consumption data is presented on a basic, yet user-friendly interface, implemented as a *website*. This way, data visualization can be performed at any device with Internet connection, at home, at work or on the move. Besides the *website*, data visualization is accessible through a digital photo frame equipped with a IEEE 802.11 b/g wireless connection, using a RSS feed mechanism.

The cost evaluation performed in section 5.2 indicated an approximated price of 17,50 € for a single *electricity meter*. Although a much lower value than several market available solutions (see section 2.1), it does not account for development, deployment and other costs. Therefore, a more rigorous analysis must be employed in order to clearly evaluate the solution's total cost.

6.1.2 Specific Achievements

The following are the specific project achievements, which can be compared with the specific project objectives, defined in section 1.3:

- A survey, documented in section 2.1, which showed that the idea of designing electricity metering systems is an interesting product idea for the present day market. Several solutions have been developed over the last years, and a few more are in the verge of being launched, including projects from commercial giants such as Google;

- The initial survey also unveiled interesting academic work in the area, including European projects which started in late 2008/middle 2009. Besides, the analysis of existing solutions was helpful to study suitable technologies and establish project guidelines, including electricity metering, sensor networks and Internet communication schemes;

- Specification of a distributed architecture, that separates the system into two main environments, local and remote, connected via Internet. This strategy moves resources from the local side into the remote side, resulting in reduced production and installations costs and ubiquitous access to visualization;

- Specification of an electricity metering unit, capable of monitoring individual electrical appliances and transmitting the measurement values through an RF medium. This included the understanding/design of specific hardware such as power metering ICs, current sensing circuitry, a switched-mode power supply (SMPS), and microcontroller communication;

- Implementation of a prototype for the electricity metering unit, achieved by integrating development tools provided by the suppliers of the technologies employed in the design mentioned above, specifically the Teridian 71M6511 IC and the Zensys ZM3102N;

- Integration of a sensor network infrastructure, based on Z-Wave, that allows for the exchange of data within a residential environment, in a reliable way. This included the implementation of an application that is capable of periodically requesting, receiving and interpreting consumption data exchanged within the sensor network and send it to a remote location, using the Internet as a communication channel;
- Integration of security mechanisms on the Internet communication;
- Specification and implementation of a basic, yet user-friendly interface, implemented as a website, which allows the system users to visualize the electricity consumption information. Due to this approach, the user interface can be accessed from any location and from any device equipped with and Internet connection. Besides the website, data visualization is accessible through a digital photo frame equipped with a IEEE 802.11 b/g wireless connection, using a RSS feed mechanism;
- Evaluation of the electricity metering system prototype functionalities, including the SMPS, the sensor network and the user interface.

In addition to the achievements listed above, during the course of this work a paper has been submitted to *BuildSys 2009 First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings* [106]. The paper can be consulted in appendix E, and although it has not been accepted, its reviews provided helpful guidelines for future developments.

The objectives defined in sections 1.2 and 1.3 have been achieved. However, the work lacked the hardware implementation of an individual electricity metering unit in a Printed Circuit Board (PCB) (although the design is finished), which is one of many possibilities for future developments. In conclusion, this project was based on the specification, implementation and integration of multiple systems and technologies, such as embedded systems design, sensor networks, software programming, communications over the Internet and web development, all of which are incorporated in a modern electrical and computer engineering program such as the one this work been made for.

6.2 Future Work

As stated in section 6.1, there are several possibilities for future developments, including the implementation of a an individual electricity metering unit in a PCB or the integration of a Z-Wave *collection point* with a *gateway*. The developed solution deserves a comprehensive scalability study in order to understand the limits of the suggested distributed architecture. Besides, further usability functionalities and studies should be applied to improve its GUI.

Some of the above possibilities are already under development. The developed work was the starting point for a master thesis at Universidade do Minho and Fraunhofer Portugal AICOS, under the title '*Small Displays in the Home Environment*' [107], currently under development. The thesis

will study user interfaces at home environments, using this work, specifically the developed digital photo frame interface as one of the subjects under analysis.

References

[1] Saul Greenberg Helen Ai He. Motivating Sustainable Energy Consumption in the Home. In *ACM CSCW Workshop. CHI 2008, ACM Press*, 2008.

[2] Wang Hongbo and Liu Changbin. Research on Incentive Mechanism to Promote Energy Efficiency in Existing Buildings. In *Industrial Engineering and Engineering Management, 2007 IEEE International Conference on*, pages 1704–1708, Dec. 2007.

[3] E. Elias, E. A. Dekoninck, and S. J. Culley. Assessing User Behaviour for Changes in the Design of Energy Using Domestic Products. In *Electronics and the Environment, 2008. ISEE 2008. IEEE International Symposium on*, pages 1–6, May 2008.

[4] C.C. Stawitz, H.S. McGehee, C.M. Devlin, F. Tan, and S. Wong. Increasing Awareness of Residential Energy Consumption: Data Analysis and Presentation for ecoMOD, a Sustainable Housing Initiative. In *Systems and Information Engineering Design Symposium, 2008. SIEDS 2008. IEEE*, pages 55–59, April 2008.

[5] Marshini Chetty, David Tran, and Rebecca E. Grinter. Getting to Green: Understanding Resource Consumption in the Home. In *Proc. of the 10th Int. Conf. on Ubiquitous computing*, volume 344 of *UbiComp*, pages 242–251, 2008.

[6] 2 Save Energy Ltd. OWL wireless electricity monitor. `http://www.theowl.com`. Last access on 1st July 2009.

[7] Modern Moulds & Tools Ltd. Eco-Eye Real-Time Electricity Monitor. `http://aaaeco.homestead.com/index.html`. Last access on 1st July 2009.

[8] REUK Renewable Energy UK. Eco-Eye review. `http://www.reuk.co.uk/Eco-Eye-Review.htm`. Last access on 1st July 2009.

[9] Maury Wright. Power Monitor Measures Cumulative, Spot Energy Consumption. `http://www.embedded.com/underthehood/212201760?pgno=1`, February 2008. Last access on 1st July 2009.

[10] The Black & Decker Corp. Black & Decker Power Monitor. `http://www.bdonlinestore.com/product_detail.asp?T1=BDT+EM100B&`. Last access on 1st July 2009.

[11] Inc. Tendril Networks. Tendril - Smart energy for life. `http://www.tendrilinc.com/`. Last access on 1st July 2009.

[12] Drew Gislason. *Zigbee Wireless Networking*. Newnes, 2008.

[13] Steven Castle. GE, Tendril Team for Smart Grid Appliances. http://www.electronichouse.com/article/ge_tendril_team_for_smart_grid_appliances/, July 2009. Last access on 20th July 2009.

[14] Martin LaMonica. Tendril: Smart grid meets Zigbee home networks. http://news.cnet.com/8301-11128_3-10001329-54.html, July 2008. Last access on 1st July 2009.

[15] Energy Optimizers Ltd. Plogg: Wireless Energy Management. http://www.plogginternational.com/ploggproducts.html. Last access on 1st July 2009.

[16] C. De Santis H. Labiod, H. Afifi. *Wi-Fi, Bluetooth, Zigbee and WiMax*. Springer, June 2007.

[17] Jan Axelson. *USB Complete: Everything You Need to Develop Custom USB Peripherals*. Lakeview Research, August 2005.

[18] Google Inc. Google PowerMeter. http://www.google.org/powermeter/. Last access on 3rd of March 2010.

[19] Inc. Energy. The Energy Detective. http://www.theenergydetective.com/index.html, March 2010. Last access on 3rd March 2010.

[20] AlertMe. AlertMe - Smart Energy and Monitoring Solutions. http://www.alertme.com/, March 2010. Last access on 3rd March 2010.

[21] Itron Inc. Itron - Knowledge to Shape your Future. http://www.itron.com/, March 2010. Last access on 3rd March 2010.

[22] Google Inc. Google PowerMeter API Introduced for Device Manufacturers. http://googlecode.blogspot.com/2010/03/google-powermeter-api-introduced-for.html, 2010. Last access on 3rd March 2010.

[23] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. Design and Implementation of a High-fidelity AC Metering Network. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 253–264, Washington, DC, USA, 2009. IEEE Computer Society.

[24] P. Levis M. A. Kazandjieva, B. Heller and C. Kozyrakis. Energy Dumpster Diving. In *In Proceedings of HotPower 09*, 2009.

[25] Nissanka B. Priyantha, Aman Kansal, Michel Goraczko, and Feng Zhao. Tiny Web Services: Design and Implementation of Interoperable and Evolvable Sensor Networks. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 253–266, New York, NY, USA, 2008. ACM.

[26] T. W. Yuvraj Agarwal and R. K. Gupta. The Energy Dashboard: Improving the Visibility of Energy Consumption at a Campus-Wide Scale. In *In First ACM Workshop on Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys 09)*, November 2009.

[27] Chiara Buratti, Andrea Conti, Davide Dardari, and Roberto Verdone. An Overview on Wireless Sensor Networks Technology and Evolution. *Sensors*, 9(9):6869–6896, 2009.

[28] IntUBE (Intelligent Use of Buildings Energy Information). "IntUBE Business Models". http://www.intube.eu/wp-content/uploads/2009/03/IntUBE_Whitepaper_BM.pdf, March 2009. White Paper.

[29] A. Di Stefano G. Fiscelli C. Giaconia R. Miceli, D. La Cascia. Impact of Novel Energy Management Actions on Household Appliances for Money Savings and CO2 Emissions Reduction. In *EVER09 Conference, Principato di Monaco*, March 2009.

[30] David Morrison. ICs Ease Adoption of Electronic Energy Metering. *Power Electronics Technology*, 17:60–64, February 2005.

[31] Inc. Cirrus Logic. Cirrus Logic Energy Measurement. http://www.cirrus.com/en/products/pro/techs/T14.html. Last access on 1st July 2009.

[32] Cirrus Logic, Inc. *CS5463 Single Phase, Bi-directional Power/Energy IC*, April 2008. Datasheet.

[33] John Catsoulis. *Designing Embedded Hardware*. O'Reilly Media, 2nd edition, May 2005.

[34] Teridian Semiconductor Corp. Teridian Semiconductor Corp. http://www.teridian.com/. Last access on 1st July 2009.

[35] Teridian Semiconductor Corp. *71M6511/71M6511H Single-Phase Energy Meter IC*, August 2007. Datasheet.

[36] Inc. Analog Devices. Analog Devices - Energy Measurement. http://www.analog.com/en/analog-to-digital-converters/energy-measurement/products/index.html. Last access on 1st July 2009.

[37] Analog Devices, Inc. *Single-Phase Energy Measurement IC with 8052 MCU, RTC and LCD driver*, January 2007. Datasheet.

[38] Inc. Microchip Technology. Microchip Technology - Energy Measurement. http://www.microchip.com/ParamChartSearch/chart.aspx?branchID=11029&mid=11&lang=en&pageId=79. Last access on 1st July 2009.

[39] Digi-Key Corporation. Digi-Key. http://digikey.com/Suppliers/us/Sigma-Designs.page?lang=en. Last access on 1st July 2009.

[40] Mouser Electronics. Mouser Electronics. http://de.mouser.com/. Last access on 1st July 2009.

[41] Premier Farnell. Farnell Portugal. http://pt.farnell.com/. Last access on 1st July 2009.

[42] S. Galli and O. Logvinov. Recent Developments in the Standardization of Power Line Communications within the IEEE. *Communications Magazine, IEEE*, 46(7):64–71, July 2008.

[43] G.N. Srinivasa Prasanna, A. Lakshmi, S. Sumanth, V. Simha, J. Bapat, and G. Koomullil. Data Communication over the Smart Grid. In *Power Line Communications and Its Applications, 2009. ISPLC 2009. IEEE International Symposium on*, pages 273–279, 29 2009-April 1 2009.

[44] HomePlug Powerline Alliance. HomePlug Powerlin Alliance Overview. http://www.homeplug.org/about, 2010. Last access on 21st January 2010.

[45] IEEE Standards Association. IEEE P1901 Draft Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications. http://grouper.ieee.org/groups/1901/, December 2009. Last access on 21st January 2010.

[46] Echelon Corporation. PL 3120/PL 3150/PL 3170 Power Line Smart Transceivers. http://www.echelon.com/products/transceivers/powerline/plsmart.htm, 2010. Last access on 21st January 2010.

[47] Maxim Integrated Products. Powerline Networking Devices. http://para.maxim-ic.com/en/search.mvp?fam=pl_net&tree=powerline, 2009. Last access on 21st January 2010.

[48] Homayoun Nikookar. *Introduction to Ultra Wideband for Wireless Communications (Signals and Communication Technology)*. Springer, 2008.

[49] A. Strba. Operating System Design Challenges for Wireless Embedded Systems Powered by Energy Harvesters. In *Applied Machine Intelligence and Informatics, 2009. SAMI 2009. 7th International Symposium on*, pages 35–40, Jan. 2009.

[50] J. Walko. Home Control. *Computing & Control Engineering Journal*, 17(5):16–19, October-November 2006.

[51] Matthias H and Dirk Timmermann. Introduction Time-Slot-Based Analysis of Bluetooth Energy Consumption for Page and Inquiry. In *9th Euromicro Conference on Digital System Design*, 2008.

[52] EnOcean GmbH. EnOcean - Self-Powered Wireless Sensors. http://www.enocean.com/. Last access on 7th July 2009.

[53] ZigBee Alliance. ZigBee Alliance. http://www.zigbee.org/. Last access on 1st July 2009.

[54] William Stallings. *Cryptography and Network Security Principles and Practices*. Prentice Hall, 2005.

[55] G. Ferrari, P. Medagliani, S. Di Piazza, and M. Martalò. Wireless Sensor Networks: Performance Analysis in Indoor Scenarios. *EURASIP J. Wirel. Commun. Netw.*, 2007(1):41–41, 2007.

[56] Mikhail T. Galeev. Home Networking with Zigbee. http://www.embedded.com/columns/technicalinsights/18902431?_requestid=369144, April 2004. Last access on 1st July 2009.

[57] A. Klein. Performance Comparison and Evaluation of AODV, OLSR, and SBR in Mobile Ad-hoc Networks. In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pages 571–575, May 2008.

[58] M.J. Lee, Jianliang Zheng, Young-Bae Ko, and D.M. Shrestha. Emerging Standards for Wireless Mesh Technology. *Wireless Communications, IEEE*, 13(2):56–63, April 2006.

[59] Fred Eady. *Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers*. Newnes, March 2007.

[60] Texas Instruments, Inc. *CC2520 Datasheet 2.4 GHZ IEEE 802.15.4/ZIGBEE RF Transceiver*, December 2007. Datasheet.

[61] Atmel Corp. *Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications - AT86RF230*, February 2009. Datasheet.

[62] Freescale Semiconductor, Inc. *MC13211/212/213: ZigBee - Compliant Platform - 2.4 GHz Low Power Transceiver for the IEEE 802.15.4 Standard plus Microcontroller*, May 2009. Datasheet.

[63] Microchip Technology, Inc. *MRF24J40 Data Sheet IEEE 802.15.4 2.4 GHz RF Transceiver*, October 2006. Datasheet.

[64] Inc. Sigma Designs. Zensys. `http://www.zen-sys.com/modules/Zensys`. Last access on 1st July 2009.

[65] Z-Wave Alliance. Z-Wave Alliance. `http://www.z-wavealliance.org/modules/start/`. Last access on 1st July 2009.

[66] P.S.H. J.F.R. *Software Design Specification - Z-Wave Protocol Overview*. Zensys, 4th edition, May 2007.

[67] Mikhail T. Galeev. Catching the Z-Wave. `http://www.embedded.com/columns/technicalinsights/193101000?_requestid=367340`, February 2006. Last access on 1st July 2009.

[68] T.C.A. S.D.H. *ZM3102N Z-Wave Module Datasheet*. Zensys, 6th edition, October 2007. Datasheet.

[69] Julie Jacobson. Has Home Automation Standard Finally Arrived? ZigBee Pro with Control4 as Anchor. `http://www.cepro.com/article/has_home_automation_standard_finally_arrived_zigbee_pro_with_control4_as_an/`, May 2005. Last access on 21st January 2010.

[70] Inc. ZWaveWorld. Z-Wave World - Product Reviews. `http://www.zwaveworld.com/reviews.php`, 2008. Last access on 21st January 2010.

[71] Daintree Networks. Introducing ZigBee RF4CE. Technical report, Daintree Networks, ZigBee, Institute of Electrical and Electronics Engineers (IEEE), 2009.

[72] Kenneth L. Calvert Michael J. Donahoo. *TCP/IP Sockets in C*. Morgan Kaufmann, 2009.

[73] Michael Papazoglou. *Web Services: Principles and Technology*. Prentice Hall, 1st edition, September 2007.

[74] J. Roy and A. Ramanujan. Understanding Web Services. *IT Professional*, 3(6):69–73, Nov/Dec 2001.

[75] Jia Zhang and Jen-Yao Chung. A SOAP-Oriented Component-Based Framework Supporting Device-Independent Multimedia Web Services. In *Multimedia Software Engineering, 2002. Proceedings. Fourth International Symposium on*, pages 40–47, 2002.

[76] Arthur Ryman. Simple Object Access Protocol (SOAP) and Web Services. In *ICSE '01: Proceedings of the 23rd International Conference on Software Engineering*, page 689, Washington, DC, USA, 2001. IEEE Computer Society.

[77] Sikha Bagui. *Database Design Using Entity-Relationship Diagrams*. Auerbach Publications, 2003.

[78] Center for Global Development. CARMA - Carbon Monitoring for Action. `http://carma.org/dig/show/world+country`. Last access on 21st July 2009.

[79] Teridian Semiconductor Corp. *71M6511/71M6511H Demo Board*, June 2007. Datasheet.

[80] Triad Magnetics. *CSE187-L: Current Sense, Low Frequency Transformers*, June 2004. Datasheet.

[81] Ed Walker. Design Review: A Step-By-Step Approach to AC Line-Powered Converters. In *Texas Instuments 2004/05 Power Supply Design Seminar*, 2005.

[82] Reston Condit. *Transformerless Power Supplies: Resistive and Capacitive*. Microchip Technology Inc., 2004.

[83] National Instruments Corporation. NI Multisim. `http://www.ni.com/multisim/`, 2009. Last access on 1st July 2009.

[84] Dragan Maksimovic Robert W. Erickson. *Fundamentals of Power Electronics*. Springer, 2001.

[85] Kenneth C. Smith Adel S. Sedra. *Microelectronic Circuits*. Oxford University Press, 2007.

[86] Power Integrtions, Inc. *LinkSwitch-TN Design Guide - Application Note AN-37*, May 2005. Datasheet.

[87] NXP Semiconductors. *BZX384 Series - Voltage Regulator Diodes*, April 2003. Datasheet.

[88] Fairchild Semiconductor. *2N3904 - NPN General Purpose Amplifier*, 2001. Datasheet.

[89] Inc. Zensys. Z-Wave Developer's Kits. `http://web1.zen-sys.com/modules/Products&Techonology/?id=14&chk=5d42873cfd4bdcb73b8551df904965de`, 2007. Last access on 1st July 2009.

[90] Ted Van Sickle. *Programming Microcontrollers in C*. Newnes, 1994.

[91] Inc. Keil Software. C51 Development Tools. `http://www.keil.com/c51/`. Last access on 21st July 2009.

[92] Ian F. Darwin. *Java Cookbook*. O'Reilly Media, 2001.

[93] Scott Oaks. *Java Security*. O'Reilly Media, 2001.

[94] Martin Kalin. *Java Web Services: Up and Running*. O'Reilly Media, 2009.

[95] Keane Jarvi. RXTX: The Prescription for Transmission. `http://users.frii.com/jarvi/rxtx/index.html`, February 2009. Last access on 1st July 2009.

[96] Luke Welling. *PHP and MySQL Web Development*. Addison-Wesley Professional, 2008.

[97] Elizabeth Castro. *HTML, XHTML, and CSS*. Peachpit Press, 2006.

[98] David Flanagan. *JavaScript: The Definitive Guide*. O'Reilly Media, 2006.

[99] Google Inc. Google Visualization API. `http://code.google.com/apis/visualization/documentation/gallery.html`, 2009. Last access on 20th July 2009.

[100] Ben Hammersley. *Developing Feeds with RSS and Atom*. O'Reilly Media, Inc., 2005.

[101] Michael Marcellin David Taubman. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2001.

[102] Darren Murph. Samsung reveals wireless 8 inch SPF-83V digiframe. `http://www.engadget.com/2007/12/13/samsung-reveals-wireless-8-inch-spf-83v-digiframe/`, December 2007. Last access on 1st July 2009.

[103] Matthew Keefe. *Flash and PHP Bible*. John Wiley & Sons, 2008.

[104] Wireshark Foundation. Wireshark. `http://www.wireshark.org/`. Last access on 21st July 2009.

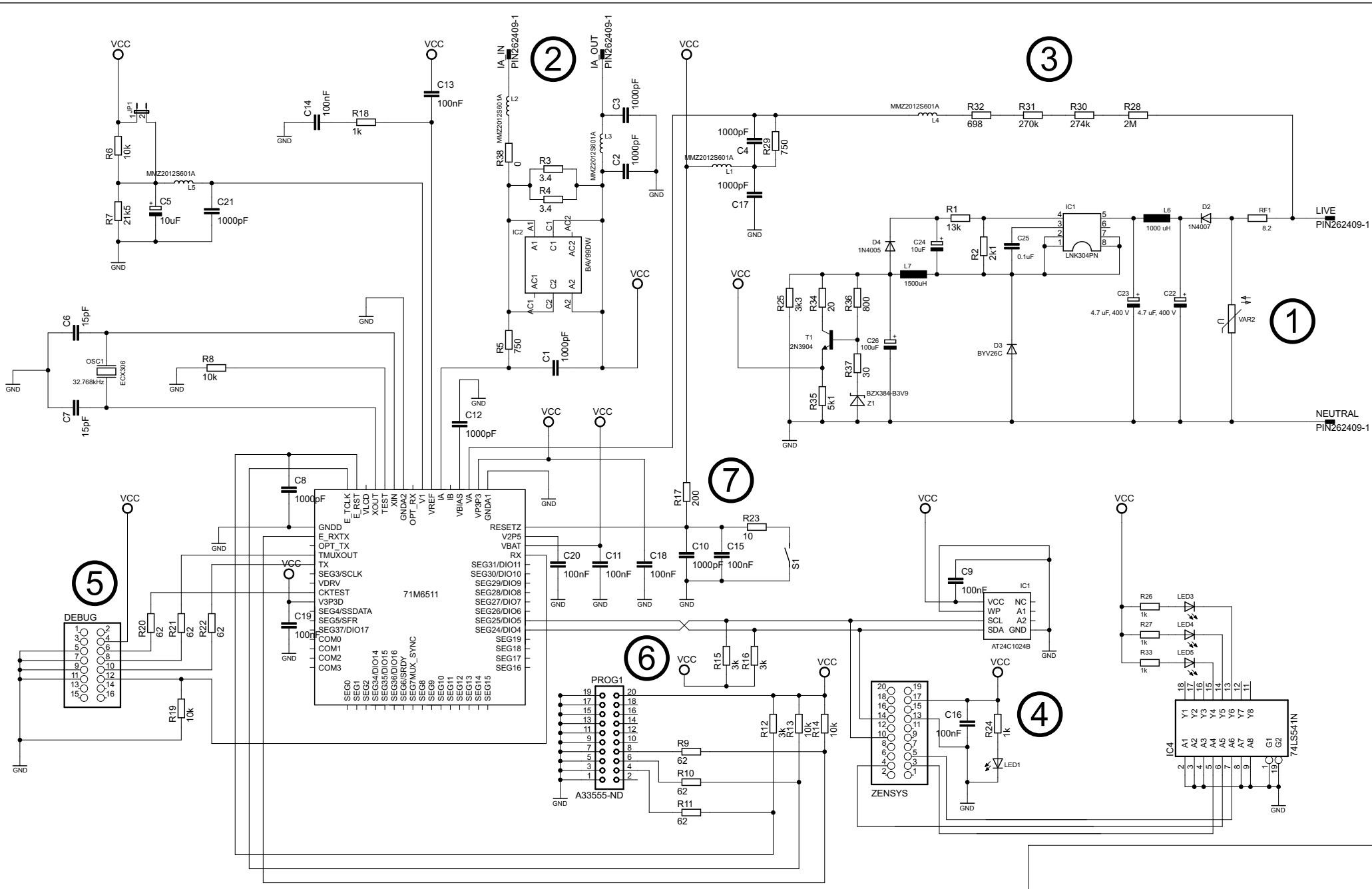
[105] DNT, Direccao de Normalizacao e Tecnologia. Ligacao de Clientes de Baixa Tensao, Solucoes Tecnicas Normalizadas. Technical report, EDP Distribuicao, Energia, S.A., May 2007.

[106] *BuildSys First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, Berkeley, CA, November 2009. In conjunction with ACM SenSys 2009, Berkeley, CA.

[107] Carlos Lopes. Small Displays in the Home Environment. Master's thesis, Universidade do Minho, 2010. Under development.

Appendix A

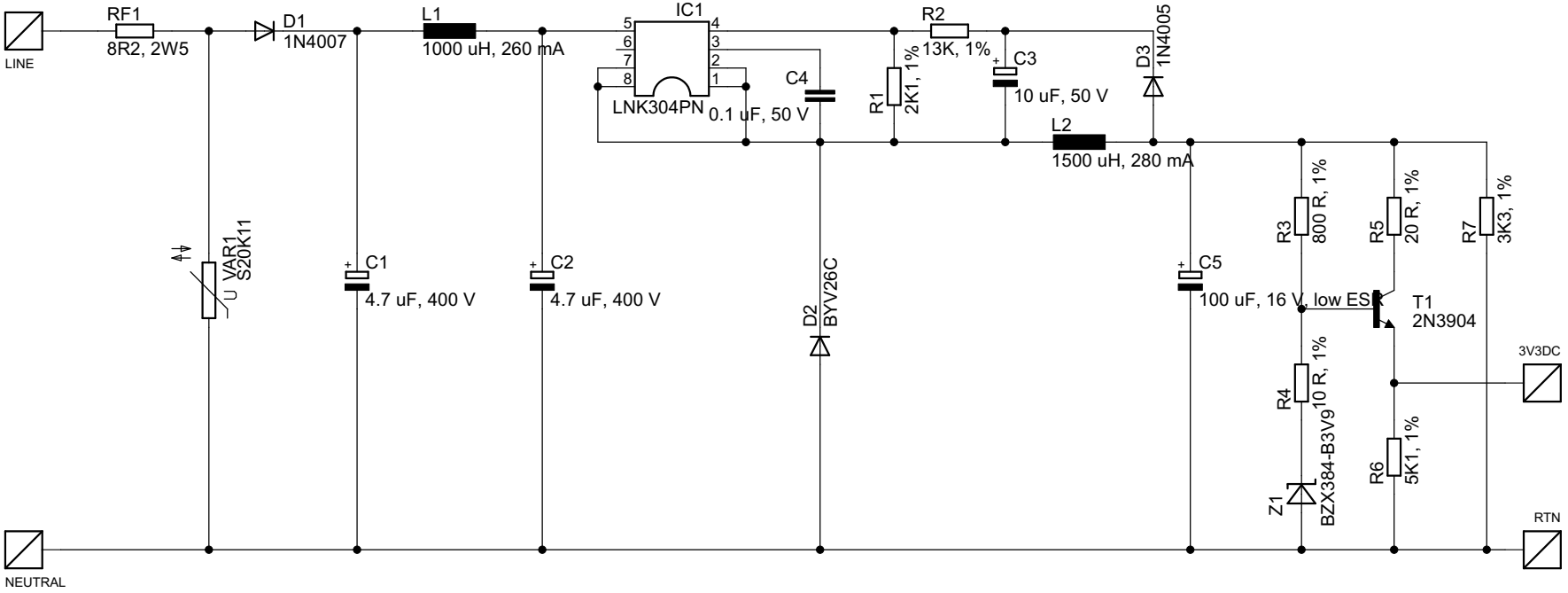
Electricity Meter Schematic



TITLE: energymeter03	
Document Number: ENME03	REV: 03
Date: 06-03-2010 16:30:25	Sheet: 1/1

Appendix B

Power Supply Schematics



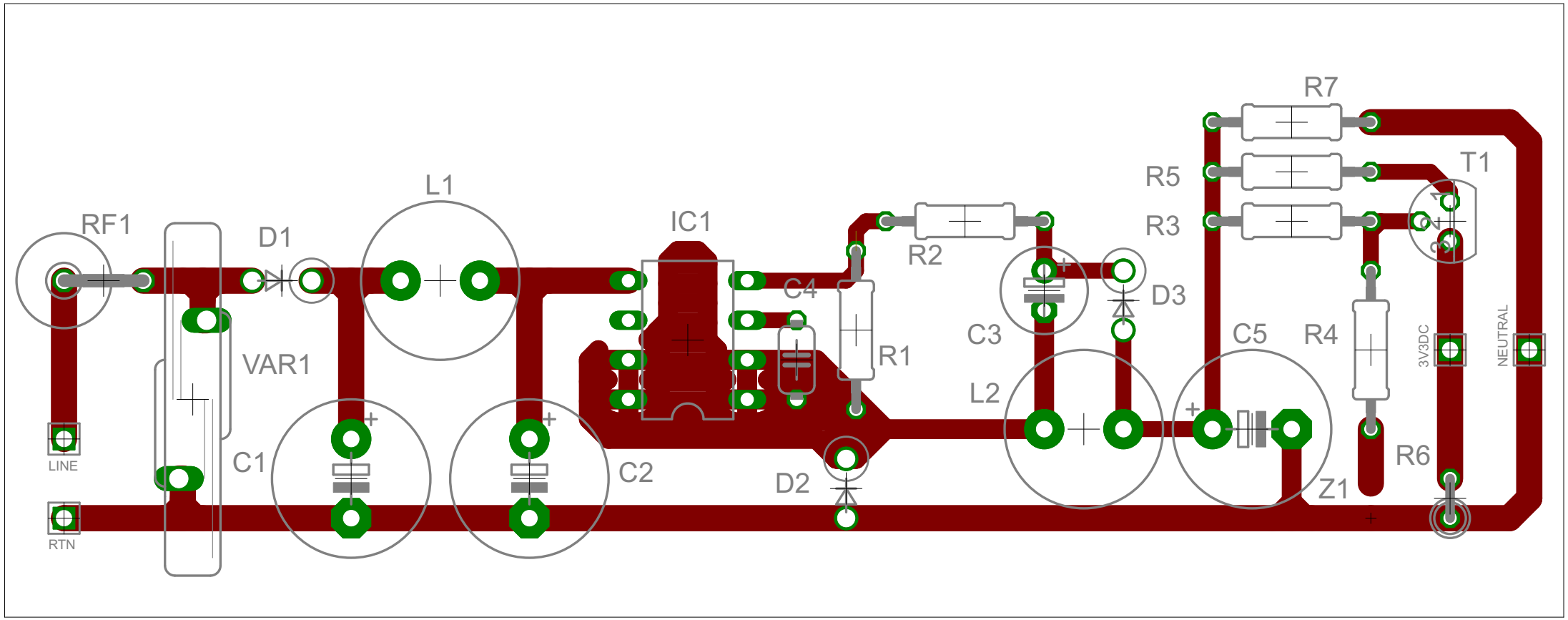
TITLE: pwr_sup_3v3_final05

Document Number: PSU3V3VR05

REV: 05

Date: 20-01-2010 10:10:25

Sheet: 1/1


RF1
D1
L1
IC1
R2
R7
R5
R3
T1
C4
D3
C3
C5
R4
R1
VAR1
L2
3V3DC
NEUTRAL
LINE
C1
C2
D2
R6
Z1
RTN

Appendix C

User Interface Screenshots

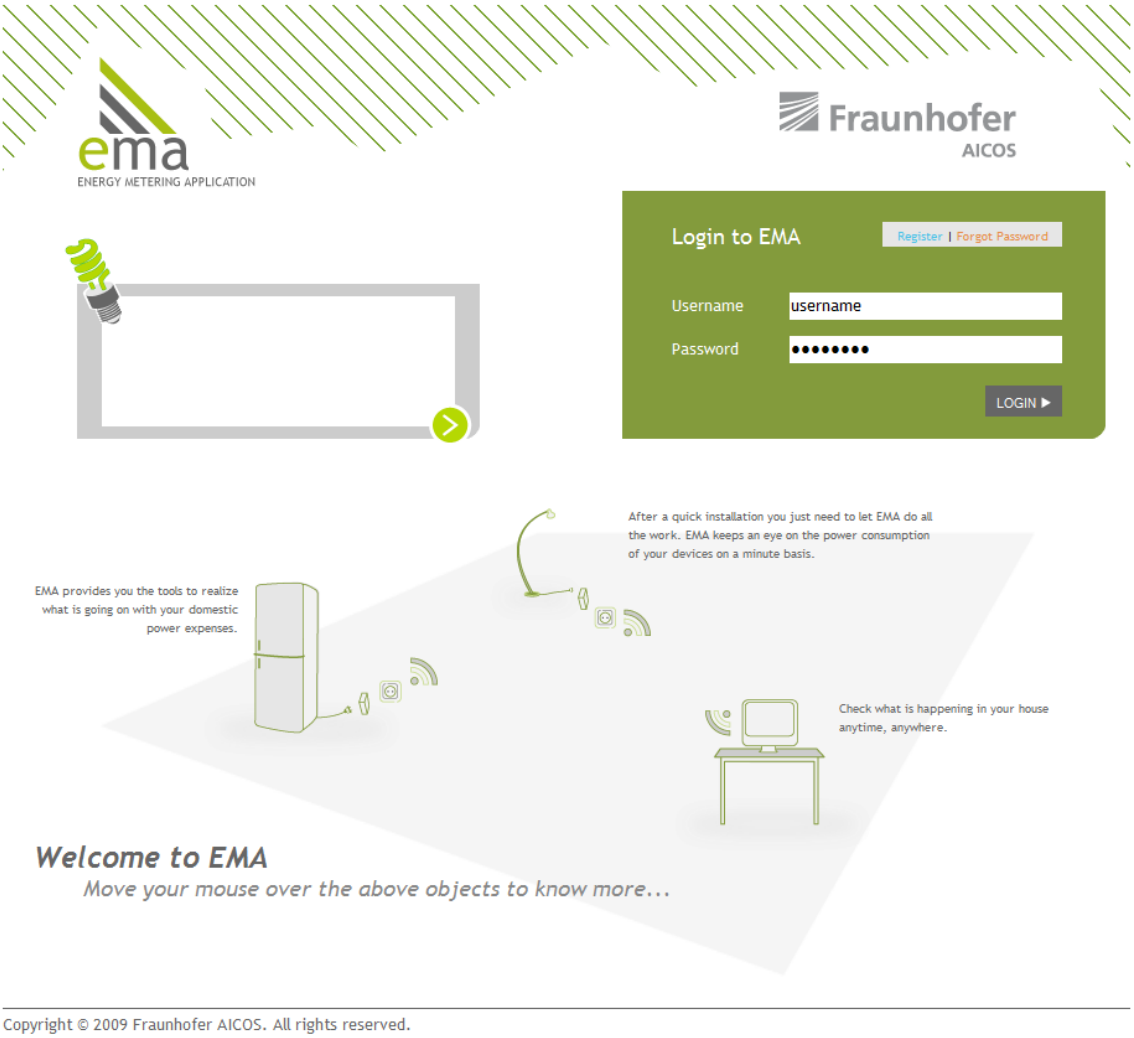


Figure C.1: *Website*'s welcome page

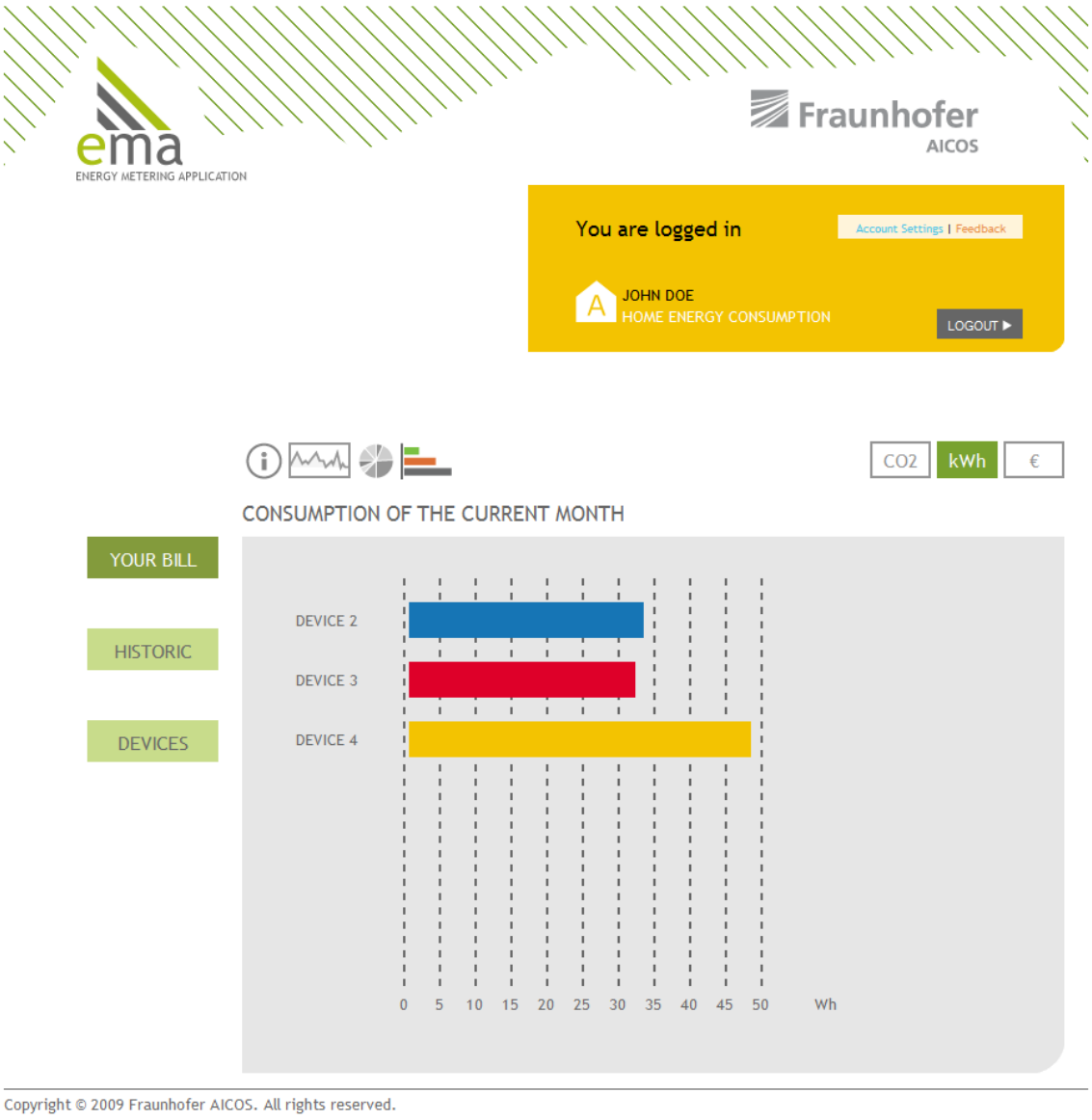


Figure C.2: *Website*'s current consumption page - bar chart



Figure C.3: *Website*'s current consumption page - timeline chart

Appendix D

Evaluation Scenario Screenshots



Figure D.1: An *energy meter* (1) metering the electricity consumption of a simple lamp, equipped with a incandescent light bulb of 40 W (2)

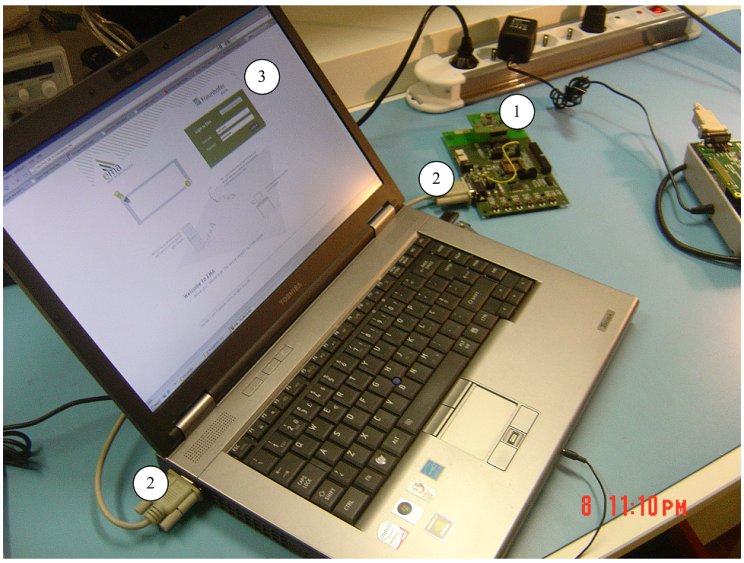


Figure D.2: A *collection point* (1) connected trough an RS-232 cable (2) to a serial port of a laptop (3) running the *gateway* application

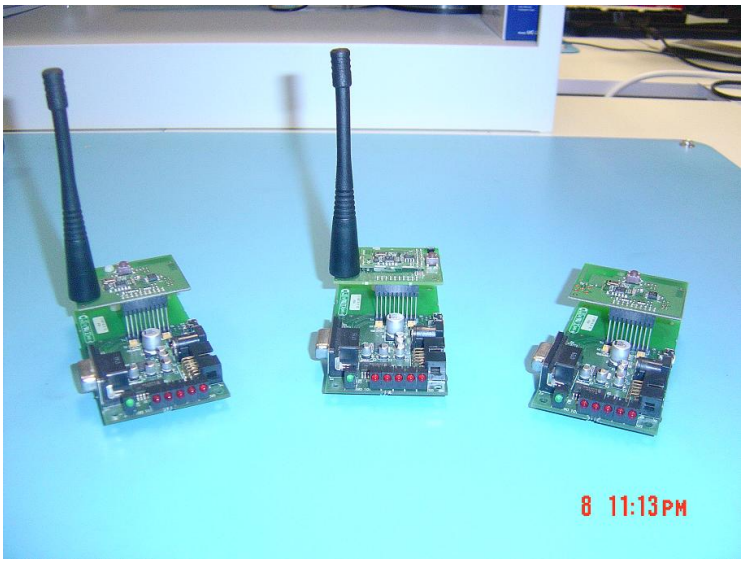


Figure D.3: 3 examples of *communication modules* (in the form of slave hardware units from the Z-Wave development kit)

Appendix E

BuildSys 2009 Paper and Reviews

Open Framework Architecture to Support Awareness of Energy Consumption in Buildings at Low-Cost

A. Rodrigues and A. Aguiar and D. Elias
Fraunhofer Portugal AICOS
{ana.aguiar, antonio.rodrigues,
dirk.elias}@fraunhofer.pt

Abstract

Motivating more efficient domestic energy consumption in developed countries is one of the main approaches towards reducing the carbon emissions, and consequently the human impact in the environment. However, a more efficient energy use for the wide spectrum of population consists in a reduction of the energy consumption with only little or no impact on comfort levels. One way to achieve this is to provide users with information on their consumption that is user-relevant and easy to understand in a time relevant manner. This paper presents an open architecture that gathers energy consumption data for individual domestic appliances in a backoffice and provides its user-friendly visualization in a platform-independent manner from any device connected to the Internet. The system was designed for low cost of the system, user friendliness, interoperability and flexibility to support different business models.

1 Introduction

Excessive energy consumption is becoming a major concern in developed countries due to increasing economical and sustainability issues [1]. In such countries, residential consumption represents 30% to 40% of the total energy demand, and it is predicted to rise whithin the following years [2, 3]. In this context, a more efficient energy use by a wide spectrum of the population is considered to have the potential for the greatest impact on reducing general overconsumption [4].

Recent research in this area defines three ways to make better use of energy in the home context: improving residential building efficiency, appliance efficiency and changing public's energy consumption behavior [3]. The first two strategies are being applied by product manufacturers and governmental institutions in the form of energy labels and efficiency ratings, as an attempt to influence the purchase of energy efficient goods [3]. However, these strategies only provide a partial solution since the general public is not aware of how to use these products in an energy efficient way [1]. Several studies, conducted in different countries, concluded that 26% to 36% of house energy consumption is a result of resident's behavior [3], supporting the importance of this last aspect in the context of residential energy efficiency.

After this last realization, the next step is to understand how to encourage home inhabitants on adopting sustainable behaviors. Recent results show that householders install energy efficiency systems in their homes mostly "for saving money, maintaining a comfortable setting" [5] and that they desire real-time information on their in-home resource consumption [1, 5]. However this may not be sufficient to perform behavioral changes in the residents, since they may not know the course of action to follow upon a certain information set. A more effective approach is to provide consumers with sustainable education while supported by the numbers of their own consumption information [1, 3], as a continuous process, i.e., both before, during and after an energy use case. This way home residents not only have access to their consumption information, but also have the possibility to understand it and decide on the most appropriate action to follow.

This paper presents an open architecture for a system with the potential to encourage sustainable energy usage within residential environments. It enables the collection and storage of energy consumption data from individual electrical appliances. It also provides an user-friendly data visualization in a platform-independent manner, accessible from any device (laptop, PDA, etc.) connected to the Internet. The system was designed for low cost and ease of installation, use of as much existing infrastructure as possible, interoperability and support of different business models. Although the system presented is in many aspects similar to existing solutions, we are unaware of a structured discussion of the issues relevant for the design of the different components and their integration into an architecture as the one proposed here.

Section 2 reviews existing electricity and energy metering systems. The next section elaborates on the design requirements for the system presented and Section 4 presents the novel architecture that we designed and developed as well as details on each component. In Section 5, we discuss the flexibility of the architecture presented and how it fits different deployment scenarios and business models. Finally, we conclude the paper and discuss future work in Section 6.

2 Related Work

Most of the developed countries' households are equipped with metering systems to measure overall electricity consumption and charge accordingly. However, these systems are not appropriate to be used by the inhabitants to understand and control their consumption costs. Recently, several energy metering systems oriented to help end-users understand their energy consumption patterns appeared in the market.

The OWL Wireless Electricity Monitor [6] and Eco-Eye [7] metering system use contactless energy sensors (clips placed around appliance cable) connected through wires to a wireless transmitter and a receiver equipped with a simple display to show instantaneous and accumulated energy consumption and translate it

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Table 1. Energy metering systems feature overview

Features	OWL, Eco-Eye	TREE	Plogg
Sensor type	No contact	Plugged	Plugged
Wireless transmission	Yes	Yes	Yes
Portable display	Yes	Yes	No
Internet access	No	Yes	Yes
Consumption accumulation	No	Yes	Yes
Power supply	Batteries / AC adapter	AC power line	AC power line
Price	34.95£, 49.99/39.99£	N/A	217£

into cost. OWL and Eco-Eye have a transmitter-to-receiver ratio of 1:1 [6], i.e., the display can show data from only one transmitter that supports at limited number of energy sensors wired to it.

TREE (Tendril Residential Energy Ecosystem) [8] uses a lower cost metering unit, consisting of a module to be placed between the device plug and the power outlet. It employs ZigBee [9] for wireless technology and has the capability of sending data through an IP gateway connected to the home's Internet router to a backoffice where it can be visualized in a website. Plogg [10] is a complete metering system similar to TREE that provides the metering devices which different wireless communication standards (such as Bluetooth and ZigBee) and sends the data to a backoffice where it is available for visualisation.

The Google Power Meter [11] and Microsoft Hohm [12] focus on the visualization of consumption data, offering interesting services, like the introduction of active and semi-personalized advices about good energy consumption policies.

Table 1 summarises the features of the covered solutions. The OWL and Eco-Eye energy metering systems support only a very limited amount of energy sensors per display and the user interaction is limited to an LCD display controlled through a set of buttons. Furthermore, since the energy sensors are wired to the wireless transmitter, they must be placed close to each other. TREE and Plogg support several energy meters that can be distributed as wished around the home and provide web interfaces to the consumption data, offering a more user-friendly visualisation of energy consumption. TREE is very similar to the system proposed in this paper, but it is designed to be offered as a service by utilities or energy retailers, so they are obliged to buy the service from them and the price of the service or single components is unknown. Plogg also offers similar functionalities to our system, but at a very high cost: one IP gateway costs 120 and one socket meter 97, so that metering one application alone has installation costs of over 200.

A recent research on understanding resource consumption in the home [5] found out that householders would like detailed real-time information on their in-home energy consumption as well as tips on how to save resources that are relevant to their situation. Although some were aware of the existence of systems similar to Plogg or TREE, these are perceived as "overly expensive or as requiring extensive rewiring" [5].

3 Design Requirements and Guidelines

The goal of this work was to develop a low-cost electricity metering system for residential use, capable of displaying electricity consumption in a way that motivates energy consumption savings by common householders. Although the system described here is specified to monitor electricity, it should easily be extended to meter other types of energy, like gas, water or heating.

To achieve a total low cost, besides the low-cost of the devices, the system should use as much existing infrastructure as possible. Furthermore, it should be easily installed in the home and provide easy to understand real-time information on the consumption of single home appliances. It should also be easy to enhance with other services and features.

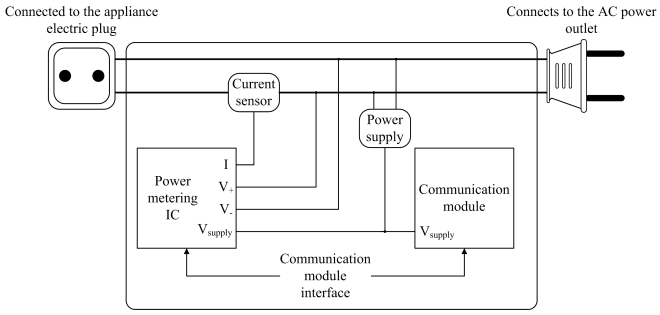
Based on these requirements, the following design guidelines were defined for the system described in the next section:

- **Use a plug (socket) meter** The components required to build a plug meter are less costly than those needed for a clip meter.
- **Keep tasks done in the home to a minimum** The less tasks have to be done in the home, the lower the complexity of the equipment to be installed in the home and, consequently, the costs of that equipment. Furthermore, a lower complexity of the components in the home also implies less chance for failure and less need for maintenance.
- **Rely on continuous Internet connection available in most home nowadays** Since most tasks will be removed from the home and implemented elsewhere, there will be the need to transfer the consumption data from the home. The use of an Internet connection available in most homes nowadays uses existing infrastructure, contributing to keeping low total system costs.
- **Use wireless technology for communication in the home** Besides allowing the free positioning of sensors around the home, this approach also provides communication independently of household configuration, constibuting to ease of installation. The use of wireless network in the home is also advantageous for future expansions to metering other types of energy with the same system.
- **Use a web-based graphical user interface (GUI)** A web-based GUI provides visually appealing items (charts and graphs) and enhanced interactivity with the user, and this can be achieved easily by using a wide range of existing routines and libraries. Furthermore, enhancements to the visualisation and addition of other features, like user-, appliance- and/or context-aware tips on energy consumption can be easily added to the service without the need for any user action (like a software update). Finally, a web-based interface can be easily accessed from any Internet connected device with a browser, inside or outside the home, and information can easily be converted into other forms of distribution, like an RSS feed, all of which can be accessed from devices already owned by the user, so no display device is needed. All these factors contribute to increased interaction and usability of the system at low cost.
- **Modular design and open standards for clearly defined interfaces** A modular design and the use of open standards for communication among system modules provides great flexibility to the system, making it easy to change/enhance any part of the system. This is very important to support different business models.

4 System Architecture

The system architecture proposed following the design guidelines presented above is depicted in figure 1. It consists of three main blocks: **home**, **backoffice** and **user interface**. While the first two are sets of physical components, the user interface consists

Figure 2. Functional blocks of the electricity meter



of both the data visualisation GUI and the physical devices that enable access to it. The components interact with each other through different interfaces identified in figure 1 by letters.

The system follows the typical Internet client-server architecture where the homes are the clients and the backoffice the server. This separation is fundamental for a low cost architecture, making use of existent infrastructure (home gateway, Internet connection, GUI on existing devices) to avoid the necessity of home-located resources capable of constant data processing, saving the end-user unnecessary material and power consumption costs. By moving resources away from the home to the backoffice, maintenance efforts leave the responsibility of the end-users, making the system more user-friendly and easier to manage, improve and extend with enhanced features.

The next sections describe the details of components and interfaces of the main system blocks.

4.1 Home Environment

The home environment handles the measurement and collection of electricity consumption data and encloses the **electricity meters**, the **collection point** and the **gateway**.

4.1.1 Electricity Meter

The electricity meter is the physical component that measures the electricity consumption of a given appliance and is positioned between it and the AC power outlet. The electricity meter is to be placed between the AC power outlet and the plug of the electrical appliance to be monitored, so that not only the current and voltage can be sensed, but also the power to operate collected.

The fulle separation of the measuring and communication modules is a design decision made to provide for the straightforward expansion of the system to other types of metering, since the communication module can then be applied to any kind of energy measuring module.

The power metering IC is an integrated circuit equipped with resources to perform a electric energy calculation and output its value, given a current and a voltage as inputs. Table 2 shows the characteristics of off-the-shelf energy metering ICs currently available on the market that are relevant for this work. It was important to use an IC that can deliver the energy consumption value that can be readily transmitted by the communication module, so that the latter can be re-used with other sensors without the need for any change in firmware. The energy-to-pulse output requires that the pulses be counted, so the cheapes metering IC from Microchip did not suit our purposes. The choice for our proof-of-concept implementation fell upon the Teridian 71M6511 for its good price/features relationship. It measures the accumulated consumed real energy in the period of time between two measurement requests in kWh.

A current sensor is needed to sense the current flowing through the electric cable and prepare it for input to the power metering IC. Several options were available to sense the current: shunt resistors, current-sense transformers and Rogowski coils. The latter are very expensive and thus inadequate for our goals. Shunt resistors are easy to use, but more expensive and bulky than current sensing transformers, so we chose the latter for our prototype module.

The **communication module** consists of a wireless transceiver IC and RF front-end and the necessary firmware which is connected to the power metering IC via an I^2C [13] interface. Finally, all the components of the energy meter are powered by a **power supply** that coverts power from the AC power line into DC values suitable to be fed to both the measuring and communication modules [14, 15]. A switched power supply design was chosen to keep costs low.

4.1.2 Home Wireless Network

The communication among the components in the home environment is done over a wireless network, interface **A** in figure 1. Wireless sensor networking technologies like ZigBee or Z-Wave [16] are the most appropriate for our goals, as they provide the desired mesh networking capabilities necessary for automatic network setup and off-the-shelf modules can be easily obtained. Z-Wave was chosen for our prototype implementation as the certification process and the support for development were more advanced at the time the work started.

The wireless network is formed by several nodes, the electricity meters and the **collection point**, scattered around the home. The network follows a master-slave model: with the exception of the collection point, which behaves as the network master, all the nodes are electricity meters playing the role of slaves. The collection point and the electricity meters communicate directly between each other, as far as possible. If the collection point cannot directly reach another node, it establishes a connection via other nodes operating as routers [16]. All electricity meters have the capability to route frames to nearby nodes, so a meter is not connected only if there is no other within its wireless range.

The collection point is also responsible for managing the inclusion and exclusion of other nodes as well as periodically requesting measurement values from electricity meters. Besides, it collects the answers to those requests and sends them to the **gateway** via interface **B**. The electricity meters are in power down mode for most of the time and only respond when activated by a request from the collection point.

4.1.3 Gateway

The **gateway** is the point that interfaces the home environment and the Internet. The gateway hosts an application that consists of two main parts, one responsible for receiving data from the collection point (via interface B) and another that sends the data to the backoffice via Internet (interface C). Again, this separation keeps the system modular only a specific self-contained part of the application must be changed to accomodate changes in the technology used for interface B or C.

The gateway application should be capable of handling the reception and transmission of multiple data blocks, without data discarding bottlenecks in any of its two parts. It can run on a PC or other type of Internet interface, such as a router. The latter shows a big advantage as a gateway application host, since it is a device that generally is always running, unlike a PC. This way, the home side could constantly send consumption information to the backoffice without the expenditure of unnecessary resources and power. A thorougher discussion on the gateway implementation is presented on Section 5.

In our proof-of-concept prototype, the gateway application is implemented in Java on a PC connected to the Internet (the gateway), with the interface B being USB and interface C a SOAP client that will be discussed in Section 4.2.1.

Figure 1. System Architecture

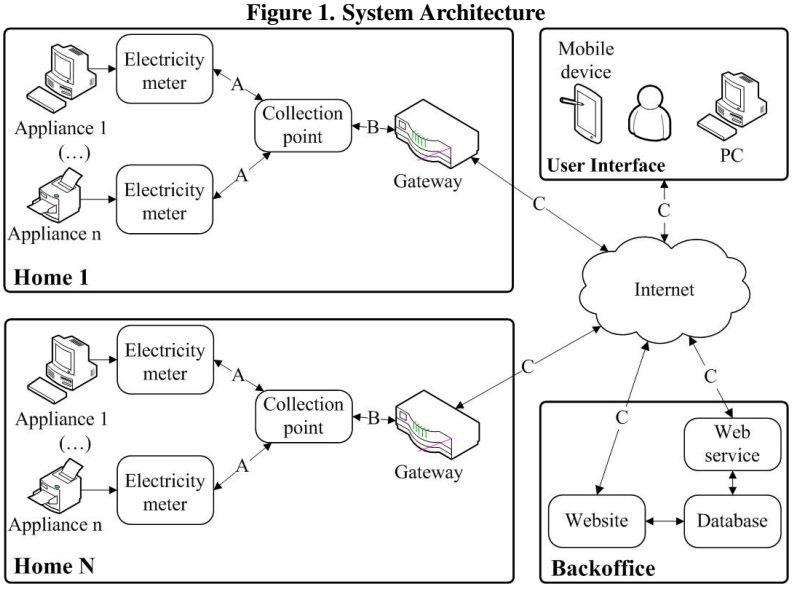


Table 2. Overview of features of off-the-shelf energy metering ICs. Prices obtained from component distributors on July 1st for more than 1000 units.

Chip	Manufacturer	Measured Quantities	Accuracy	Interface	Price [€]
CS5463	Cirrus Logic	Apparent, reactive, real energy	≈0.1%	Serial, Energy to Pulse	3.29
71M6511	Teridian	Apparent, reactive, real energy	≤0.1%	Serial, Energy to Pulse, LCD driver, Optical	2.69
ADE7569	Analog Devices	Apparent, reactive, real energy	≤0.1%	Serial, Energy to Pulse, LCD driver, Optical	3.22
MCP3905/06	Microchip	Real power	≤0.1%	Energy to Pulse	1.64

4.2 Backoffice

The energy consumption data measured in the home is sent to a remote location, the backoffice, in order to be stored, processed and prepared for visualisation. This section first describes the **remote communication** technology used for communication between the backoffice and the home environment, and then the system **database** which stores the measurement data in a structured way. Despite being physically running in the backoffice, the web-based GUI is not covered in this section as it belongs to the user interface (Section 4.3.1).

4.2.1 Remote Communications

The communication between the backoffice and home is based on Web Services [18]. Web Services are of web applications that communicate using the Simple Object Access Protocl (SOAP), an XML-based protocol designed for the exchange of messages in a distributed environment [19]. SOAP is advantageous as it features multi-platform interoperability, i.e., it allows for normal operation between objects of different platforms and/or languages [19], and ease of development due to re-use of existing routines and libraries [20, 21].

A SOAP client application, running on the gateway sends SOAP requests to a remote server that hosts a custom Web Service, specified to execute operations which involving interactions with the main database:

- Insert a new measurement value, from a given electricity meter, in the backoffice database.
- Insert a new electricity meter entity in the database. This operation is related to the automatic registration of an electricity meter, described in Section 4.3.2.

Communication of user data over the Internet raises the need to provide data confidentiality, data integrity and access control. To address those issues, the SSL (Secure Sockets Layer) protocol [22] is used for server-side authentication and data encryption, providing data confidentiality and integrity [22]. For client-side authentication, a username and password combination is used, providing controlled access.

4.2.2 Database

A **database** is used to store the measurement data in a structured way. The database stores information about the end-users of the system, the electricity meters owned by the various end-users and the measurements made by the electricity meters over the time. The measurement values are stored in energy units (kWh), along with its date and time information and the identification of the electricity meter responsible for the measurement.

4.3 User Interactions

The user interacts with the system in two ways: for the installation of the meters and for visualising the data. The latter interaction is provided by the third main system block, the user interface, that groups the application responsible for processing the consumption data and generating the visual information and the renderer software. This is primarily a common browser that runs on any device with Internet access, like a mobile device or a regular PC, but it can also be an RSS feed reader on a similar device or on a photo frame.

4.3.1 Web-based Interface

The graphic visualisations of consumption data are presented through a website, hosted in a HTTP server with access to the back-

office database. In this way, the GUI can be accessed from any device with Internet connection, at home, at work or on the move.

In terms of user interaction, the main goal of the website is to inform the user, displaying the collected electricity consumption information in a useful way that he can easily understand. The consumption information is shown in a quantitative way using energy, equivalent CO_2 emissions and monetary units. Power or energy related values, W (Watt) or Wh, can be directly displayed without previous processing, since this is the specified format for saved data in the system's database. The quantification of CO_2 emissions depends on the electricity production mix, and it is calculated by multiplying the energy consumption in kWh by a factor (expressed in kg/kWh) that indicates the released amount of CO_2 during the production of 1 kWh for a certain nation [23]. The value of these factors is calculated and presented by CARMA (Carbon Monitoring for Action) [23]. The monetary cost units follow the same principle, but in this case the factor expressed in currency/kWh a value dependent on the electricity provider and configurable by the user.

Besides the aspects directly related to the monitoring of electricity consumption, the website allows the end user to manage his electricity meters.

The website is also designed to instruct the end-user, by embedding advices in the website and helping the user relate the tips he receives with the numbers, an option that helps him make correct and informed decisions [1]. One possible way to provide this is to generate advices by cross-examining the data present in the main database with national or regional average consumption databases or even by comparing the consumption pattern of a particular device with its own electrical specifications. Different kind of advices could then be generated such as "*you're below the regional average, good job!*" or "*your fridge is consuming much more electricity than it should... Maybe you should repair or change it!*". Another useful service is a prognostic of the the energy costs at the end of the month if the consumption stays the same and the reduction that could be achieved is some measures were taken [5].

Besides the website, the proof-of-concept implementation provides another way for the end-user to visualise the consumption data: an RSS feed [24]. As an example device, we use a web-enabled photo frame to display graphical information about the home electricity consumption. For privacy reasons, the RSS feed is available on an address that contains a token generated per user, so that only each user can see his consumption history.

4.3.2 Addition of Meters

Another important interaction use case is the process of addition of a new electricity meter in the home, which should be as straightforward as possible to the user. The communication module of the electricity meter was specifically designed to be automatically added to the system upon startup, i.e., besides being included in the home wireless network, its information is also inserted in the system database. Upon connection of a new electricity meter module in the home, the user must simply push a button on the collection point and the consumptions of the new meter start appearing in the visualisation graphics in the GUI.

5 Discussion of the Proposed Architecture

In this paper we present a modular architecture for gathering energy measurement data from a home environment in a backoffice server and providing platform-independent visualisation at low cost. To achieve this, we integrated a home wireless sensor network with existing network infrastructure, namely the Internet.

The modularity of the architecture is the key to a great flexibility in the deployment of such a system, as it enables easy change/ extension of any block of the system without requiring modifications to the other parts, which, in turn, enables the straightforward support of different business models for providing such a service. Here, the way in which the integration of the collection point and the gateway (Section 4) is done plays a major role. For example, if the telecommunications provider, which has nowadays the capability of remotely managing the home router, offers the service, the best option would be to integrate the collection point with the gateway in a single physical device—the router—where both collection point and gateway applications run as firmware (although this implies the exchange of the home router due to the need for new hardware to interface with the sensor network). In case the service is offered by an independent 3rd party, it would be more adequate to integrate the collection point and the gateway in an IP device independent from the router (as Plogg [10] does). This device would communicate with the backoffice server, which could be e. g. Google Power Meter [11] via the home router just like any other device in a home network. These are two exemplifying alternatives to the solution implemented in out prototype, which uses a collection point connected via USB to a PC running the gateway application (Section 4.1.3).

The use of another technology for the home network also would require no change in any other module except the wireless communication modules in the home and the collection point. The same is valid to the addition of other energy measurement modules or the change of the Internet communication. For example, if RESTful communication were to be used instead of Web Services, the only changes would be to the part of the gateway application that handles the communication over the Internet and to the backoffice, while every other block could remain unmodified.

Finally, although the system was clearly designed with private households in mind, the architecture presented is valid as is for the implementation of an energy metering system in any kind of building, like offices of shops.

6 Conclusions and Future Work

This paper presents a low-cost, modular and flexible approach for detailed measurement of energy consumption in buildings, offering user-friendly, platform-independent visualisation of that data from any device with Internet access. A proof-of-concept prototype for an electricity metering system was implemented to gain a better understanding of the issues involved in the development of such a solution, of which many are discussed in the paper.

Next steps consist in the implementation of a resource-oriented service model for the communication between home and backoffice. Initially it shall be done through a gateway for the services, but the goal is to achieve lightweight, RESTful way to directly access the energy consumption data on the sensors. Additionally, the development of service extensions to provide user- and context-aware advice on how to save energy is also an issue to address.

In another direction, future work includes a usability study for the GUI to improve the user-friendliness. Also, other exploratory work will be done related to effective behavioural changes that can be achieved and the best means to achieve it, both in terms of information and visualisation techniques and other features desired by the users.

Finally, how to offer such a service in economically feasible ways remains to be discussed and studied in a systematic way.

7 References

[1] S. G. Helen Ai He, "Motivating Sustainable Energy Consumption in the Home," in *ACM CSCW Workshop. CHI 2008, ACM Press*, 2008.

[2] W. Hongbo and L. Changbin, "Research on incentive mechanism to promote energy efficiency in existing buildings," in *Industrial Engineering and Engineering Management, 2007 IEEE International Conference on*, Dec. 2007, pp. 1704–1708.

[3] E. Elias, E. A. Dekoninck, and S. J. Culley, "Assessing user behaviour for changes in the design of energy using domestic products," in *Electronics and the Environment, 2008. ISEE 2008. IEEE International Symposium on*, May 2008, pp. 1–6.

[4] C. Stawitz, H. McGehee, C. Devlin, F. Tan, and S. Wong, "Increasing awareness of residential energy consumption: Data analysis and presentation for ecomod, a sustainable housing initiative," in *Systems and Information Engineering Design Symposium, 2008. SIEDS 2008. IEEE*, April 2008, pp. 55–59.

[5] M. Chetty, D. Tran, and R. E. Grinter, "Getting to green: understanding resource consumption in the home," in *Proc. of the 10th Int. Conf. on Ubiquitous computing*, ser. UbiComp, vol. 344, 2008, pp. 242–251.

[6] 2 Save Energy Ltd., "OWL Wireless Electricity Monitor," http://www.theowl.com, last access on 1st July 2009.

[7] M. M. . T. Ltd., "Eco-Eye Real-Time Electricity Monitor," http://aaaeco.homestead.com/index.html, last access on 1st July 2009.

[8] I. Tendril Networks, "Tendril - Smart energy for life," http://www.tendrilinc.com/, last access on 1st July 2009.

[9] D. Gislason, *Zigbee Wireless Networking*. Newnes, 2008.

[10] E. O. Ltd., "Plogg: Wireless Energy Management," http://www.plogginternational.com/ploggproducts.html, last access on 1st July 2009.

[11] Google, "Google PowerMeter," http://www.google.org/powermeter/, last access on 1st July 2009.

[12] "Microsoft Hohm," http://www.microsoft-hohm.com/default.aspx, Microsoft Corp, 2009, last access on 19th August 2009.

[13] J. Catsoulis, *Designing Embedded Hardware*, 2nd ed. O'Reilly Media, May 2005.

[14] T. S.D.H., *ZM3102N Z-Wave Module Datasheet*, 6th ed., Zensys, October 2007, datasheet.

[15] *71M6511/71M6511H Single-Phase Energy Meter IC*, Teridian Semiconductor Corp., August 2007, datasheet.

[16] J. Walko, "Home Control," *Computing & Control Engineering Journal*, vol. 17, no. 5, pp. 16–19, October-November 2006.

[17] J. Axelson, *USB Complete: Everything You Need to Develop Custom USB Peripherals*. Lakeview Research, August 2005.

[18] M. Papazoglou, *Web Services: Principles and Technology*, 1st ed. Prentice Hall, September 2007.

[19] J. Roy and A. Ramanujan, "Understanding Web Services," *IT Professional*, vol. 3, no. 6, pp. 69–73, Nov/Dec 2001.

[20] M. Kalin, *Java Web Services: Up and Running*. O'Reilly Media, 2009.

[21] L. Welling, *PHP and MySQL Web Development*. Addison-Wesley Professional, 2008.

[22] W. Stallings, *Cryptography and Network Security Principles and Practices*. Prentice Hall, 2005.

[23] C. for Global Development, "CARMA - Carbon Monitoring for Action," http://carma.org/dig/show/world+country, Center for Global Development, last access on 21st July 2009.

[24] B. Hammersley, *Developing Feeds with RSS and Atom*. O'Reilly Media, Inc., 2005.

Reviews

You are an author of this paper.

4 Reviews

Review 1

Recommendation	Familiarity
Reject (1)	Expert in this area (5)

Contributions (What are the major issues addressed in the paper? Do you consider them important? (Comment on the technical depth, the importance, and the novelty of the problem addressed.))

This paper presents a hybrid star/mesh network architecture for collecting data from power meters in the built environment as well as typical N-tier web application and client-side user interface.

Relevance (Please comment explicitly on the relevance of the paper to BuildSyS topics)

Energy data collection architectures for the built environment are quite relevant to the scope of BuildSys.

Strengths (What are the major reasons to accept the paper? [Be brief])

The paper proposes a reasonable architecture and presents an initial, working design point.

Weaknesses (What are the most important reasons NOT to accept the paper? [Be brief])

The paper lacks novelty. Several other recent papers have appeared that propose a similar architecture and this paper makes little effort to distinguish its contributions for the prior work. It also does not identify challenges that warrant further exploration, so I find it difficult to see how the paper would drive discussion at the workshop.

Detailed Comments (Please provide detailed feedback to help the TPC to assess the paper and to help the authors improve their paper.)

To a first approximation, the architecture proposed in this paper is not particularly surprising (and has been used in earlier papers on power metering, see: Jiang (IPSN'09) and Priyantha (Sensys'09) for two recent examples for the academic literature (there are several industry examples as well, including Sentilla and Arch Rock, are not mentioned, and Tendril and Plogg, which are mentioned).

The paper claims that Tendril and Plogg's solutions are very similar, but designed to be offered by the service provider. This seems to be a distinction without a difference, especially since this paper does not address the many issues that arise with an "open" network (e.g. security, device association, etc.).

The paper also argues that the Tendril and Plogg solutions are too expensive, but then does not offer a more competitive price point in the rest of the paper. Of course, the paper does provide pricing for just the energy metering ICs (between Lbs 1.64 and 3.29), but this hardly constitutes the major cost of a power meter.

The paper argues that the modularity of the architecture is the key to changing/extending any block of the system. Effective modularity has a lot to do allocating functionality and defining interfaces that are compatible across all layers of communication, including link, network, transport, and application. This paper does not address this basic issue, instead offering a token discussion of SOAP because it features multi-platform interoperability.

Others have argued against an application layer gateway, but this paper argues that having one promotes modularity. It's not at all obvious that having an application layer gateway helps in this case. For starters, the application must be installed on the gateway, and this means the gateway itself has to be an open, secure, multivendor environment. This paper does not discuss the majority of issues that arise in this context. See, for example, the OSGi work to get a sense of what's needed.

Review 2

Recommendation	Familiarity
Borderline (3)	Familiar with this area (4)

Contributions (What are the major issues addressed in the paper? Do you consider them important? (Comment on the technical depth, the importance, and the novelty of the problem addressed.))

the paper describes a framework that enables collection of energy consumption data of electric appliances and visualization of the data to end-user. the framework promotes open architecture and interoperability through the use of web services.

Strengths (What are the major reasons to accept the paper? [Be brief])

end-to-end solution for visualizing energy consumption data to users. the system is modular and easy to deploy.

Weaknesses (What are the most important reasons NOT to accept the paper? [Be brief])

not enough distinction from the previous work.

Detailed Comments (Please provide detailed feedback to help the TPC to assess the paper and to help the authors improve their paper.)

the work is similar to other efforts in this area (both commercial and academic).

Review 3

Recommendation	Familiarity
Borderline (3)	Some knowledge of this area (3)

Contributions (What are the major issues addressed in the paper? Do you consider them important? (Comment on the technical depth, the importance, and the novelty of the problem addressed.))

The authors describe a high level system and prototype for collecting energy use data in a home and analyzing and storing the data off site. The overall goal of the system is to help homeowners make intelligent decisions about their energy use.

This paper demonstrates a system which can independently monitor each appliance with a simple plug-in device, which could be a very powerful tool for both homeowners and energy suppliars

Relevance (Please comment explicitly on the relevance of the paper to BuildSyS topics)

As this paper deals with home energy monitoring, it seems to directly fit in to the BuildSys topics

Strengths (What are the major reasons to accept the paper? [Be brief])

The system uses almost exclusively COTS products in order to achieve a prototype. This can help to make the system very quick and relatively cheap to implement on a wide scale.

As this paper looks at the entire system, the technical depth of the interaction of the components is well described and well thought out.

Most importantly, this system gives a complete home monitoring solution for electrical systems (with small mention of being able to expand that to other utilities).

Weaknesses (What are the most important reasons NOT to accept the paper? [Be brief])

Weaknesses (What are the most important reasons NOT to accept the paper? [Be brief])

A prototype is mentioned several times in the paper, but no results are given from this prototype, and no mention as to why such results are not included. Even a simple analysis of the data collected would help this paper tremendously

The topic covered is well studied, with many commercial solutions similar to the proposed system. Many of these products are not mentioned, and not enough emphasis is given to what makes the proposed solution better than what is already available

Many grammar and spelling errors in the paper make some sections difficult to read

Detailed Comments (Please provide detailed feedback to help the TPC to assess the paper and to help the authors improve their paper.)

Though the topic covered is very important, the paper itself fails to make a convincing argument for the authors work. Overall the system does seem to be very promising, and seems to offer a system which could lead to a product people have a great deal of demand for demand for. However, many similar COTS product were never mentioned, most of which can be found with a simple internet search.

If the background section of the paper were expanded significantly to include currently available systems, and some results were given from the prototype mentioned in the paper, this paper would greatly benifit

Review 4

Recommendation	Familiarity
Reject (1)	Expert in this area (5)

Contributions (What are the major issues addressed in the paper? Do you consider them important? (Comment on the technical depth, the importance, and the novelty of the problem addressed.))

As pointed in the document, the proposed architecture is in many ways very much similar to existing and available systems.

Section 3 lists a set of requirements that are fairly well-known. I would just point out that the use of wireless connection in the home is certainly an interesting option but may advantageously combined with PLC type connections.

The paper (although well organized and with clear explanations) states the obvious with a lack of novelty and technical breath.

Relevance (Please comment explicitly on the relevance of the paper to BuildSyS topics)

Very relevant.

Strengths (What are the major reasons to accept the paper? [Be brief])

The paper is well organized and clearly written.

Weaknesses (What are the most important reasons NOT to accept the paper? [Be brief])

Real lack of technical breath and novelty.

Detailed Comments (Please provide detailed feedback to help the TPC to assess the paper and to help the authors improve their paper.)

No more comments.