Faculdade de Engenharia da Universidade do Porto

Multi-Protocol Sensor Node for Internet of Things (IoT) Applications

Pedro Miguel Pinheiro Fonseca Ferreira

FINAL VERSION

Integrated Master in Electrical and Computers Engineering
Telecommunications, Electronics and Computers

Supervisor: Hélio Mendes de Sousa Mendonça
External Supervisor: Pedro Miguel Cruz

24th june 2019

Resumo

Na última década, com o aumento do número de dispositivos conectados à Internet, tem-se generalizado a disponibilidade de soluções IoT para o mercado em massa, prevendo-se um futuro de interconectividade em que grandes quantidades de informação estarão rapidamente disponíveis para influenciar as nossas decisões.

As tecnologias IoT tiram partido dessa conectividade e combinam-na com diferentes tecnologias disponíveis, sejam elas de identificação, opções de sensorização e atuação, processamento de sinal e análise de dados, deste modo contribuindo para a integração e convergência de diversos sistemas. Hoje em dia, há uma grande diversidade de segmentos (transporte, agricultura, *smart cities*, indústria 4.0, *smart homes*, etc.) que estão a ser influenciados e desenvolvidos com base em novas estruturas IoT.

Este documento irá descrever as implementações de uma solução IoT *end-to-end*, focando-se especialmente num *sensor node* multiprotocolo com base na *board* desenvolvida pela Pycom, o FiPy. A presente solução pretende disponibilizar um medidor de energia baseado no chip de medição de energia monofásico da Microchip, o ATM90E26. Uma análise de performance vai ser apresentada, abordando a eficácia tanto dos dois protocolos de comunicação implementados (LoRaWAN e WiFi) como da precisão e robustez das medições adquiridas pela solução proposta. Este sistema vai ser explicado na totalidade, incluindo as partes estruturais que o suportam.

O objetivo é disponibilizar um caso de uso prático para indústria 4.0, podendo originar algum conhecimento na matéria de qualidade na monitorização de energia.

Abstract

Over the last decade, as the number of devices connected to the Internet increases and the availability to mass consumers of IoT solutions is more common, it is foreseeable a future of interconnectivity where large amounts of data are readily available to influence our decisions.

IoT technology takes advantage of that connectivity and combines it with the different available functionalities like identification, various sensing/actuation options, signal processing and data analytics, in this way contributing to the integration and convergence of several systems. Currently, a big diversity of fields (transportation, agriculture, smart cities, manufacturing /Industry 4.0, smart homes, etc.) is being influenced and developed in the basis of new IoT arrangements.

This thesis will describe the implementation of an end-to-end IoT solution, focusing specifically in the multi-protocol sensor node using Pycom's FiPy board. This solution intents on offering an energy meter that is based on the single-phase energy meter sensor chip from Microchip, the ATM90E26. A performance assessment will be presented, addressing the effectiveness of both communication protocols implemented (LoRaWAN and WiFi) and the accuracy and reliability of the measurements acquired by the proposed energy meter. This arrangement will be fully explained, including all the structural parts that support it.

This will provide a practical use case in the field of Industry 4.0, leading to insights for power quality monitoring.

Acknowledgements

Gostaria de começar por agradecer ao meu orientador, Professor Hélio Mendes de Sousa Mendonça, pela disponibilidade manifestada desde o início, pelos concelhos e indicações sobre o melhor rumo a tomar e pelo acompanhamento ao longo de todo o trabalho.

O meu especial agradecimento, também, ao Engenheiro Pedro Cruz por me ter proporcionado a possibilidade de desenvolver este projeto em contexto real de trabalho, por todo o apoio prestado e pelas várias conversas que me permitiram olhar para cada questão sob várias perspetivas.

Gostaria, ainda, de agradecer ao meu colega de equipa, Rafael Miranda, na fase de desenvolvimento do projeto, por toda a colaboração na resolução das dificuldades com que me deparei ao longo deste trabalho e por todo o apoio prestado.

Por último, devo também um agradecimento à minha namorada, à minha família e ao meu grupo de amigos, pelo apoio incondicional, por terem estado sempre presentes, pela sua disponibilidade e pela forma como sempre me acompanharam e motivaram ao longo de todo deste caminho.

A todos, um sincero obrigado.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

List of abbreviations

IoT	Internet of Things
IEEE	Institute of Electrical and Electronics Engineers
RFID	Radio Frequency Identification
LoRaWAN	Low Range Wide Area Network
MQTT	Message Queue Telemetry Transport
NIC	National Intelligence Council
ITU	International Communication Union
FCC	Federal Communications Commission
P&G	Procter & Gamble
CPU	Central Processing Unit
MEMS	Micro-electro-mechanical systems
LED	Light Emitting Diode
GHz	Gigahertz
MHz	Megahertz
m	meters
LPWAN	Low Power Wide Area Network
Mbps	Megabits per second
Gbps	Gigabits per second
ISM	Industry, Scientific and Medical
MAC	Media Access Control
HVAC	Heating, Ventilation and Air Conditioning
HEMS	House Energy Management System
USD	United States Dollars
TCP/IP	Transmission Control Protocol
CSMA-CA	Carrier-Sense Multiple Access with Collision Avoidance
MIT	Massachusetts Institute of Technology
FSK	Frequency-shift Keying
HTTP/HTTPS	Hypertext Transfer Protocol/ Hypertext Transfer Protocol Secure
WSN	Wireless Sensor Network
SF	Spreading Factor

Chapter 1

Introduction

This chapter's goal is to offer a brief introduction to the thesis "Multi-Protocol Sensor Node for Internet of Things (IoT) Applications". On the first section, 1.1, it is presented the contextualization of the subject of this thesis. Then, on section 1.2, the motivation that prompted its realization. On section 1.3, the objectives the project intends to achieve. Lastly, section 1.4 will bring forth the division into different tasks and the scheduling planned for each one.

1.1 - Context

You can not turn a blind eye on the overwhelming increase of the number of devices connected to the Internet over the last decade [13]. According to Gartner predictions [14], around 25 billion connected devices will be in use by 2020. Going hand in hand with that growth is the concept of Internet of Things (IoT), which, despite having a multitude of definitions, was very simply described by IEEE, as "A network of items—each embedded with sensors—which are connected to the Internet."[7].

IoT takes advantage of that connectivity and, combined with the integration and convergence of different available technologies [15], (IP, RFID, sensor networks, home networks, smart metering, etc.) it can influence and further develop an abundance of areas, such as [15]: security, transportation, e-Health, manufacturing (Industry 4.0), utilities, among several others.

In Industry 4.0 greater concerns for automation and energy efficiency are moving a new market where IoT could be of great benefit and where new solutions are being introduced at a very fast pace.

1.2 - Motivation

In this context, it makes sense to fully realize and integrate all the different systems that compose a fully functional IoT solution. That includes sensors, gateways and data processing in shared systems or Clouds. We intend to take advantage on the surging concept of Industry 4.0 to potentiate new applications and introduce a new, fully complete, and highly customizable setup developed specifically for this market.

In this way, the opportunity presents to better analyse and evaluate different key components of such a system and the challenges and/or restrictions entailed in its development.

1.3 - Objectives

The main objective of this project is to develop an entirely IoT enabled multi-protocol (Wi-Fi, LoRaWAN) application using Pycom's IoT development board - FiPy. The application includes a commercially available sensor, the ATM90E26, developed by Microchip. It features energy metering capabilities, in the sense that it can sample current and voltage values in addition to various other metrics relating to power consumption.

The aim of the project is to achieve a final prototype that is capable of successful measurement acquisition and communications, through new firmware implementations. The prototype should seamlessly integrate both the sensor and the FiPy into a single PCB to be encased in a DIN-Rail of small size.

Thus, we can fully grasp the concept of Internet of Things and what all the different parts of the development entail.

1.4 - Task identification and scheduling

After the redaction of the literature review for this thesis project, it was possible to get a better grasp of the technologies involved and to get a superior understanding on how to frame the tasks at hand, and the strategies that will be implemented.

The work plan and the suggested schedules were adjusted considering the total time invested in the development of this project to be the duration of the second semester.

The first task consists of getting acquainted to the development platform (FiPy), developing generic and simple applications/solutions and getting the know-how of the python programming language and the communication protocols involved.

The second task will be the development of an IoT sensor according the required specifications.

Furthermore, there is the integration of the sensor with a gateway and then with a Cloud platform.

Then, the completed solution will be subject to various tests in order to be validated.

The last task consists on writing the final dissertation document and propose possible improvements to the implemented solution.

The following Gantt Chart, Figure 1, will present the aforementioned tasks, as well as their distribution in the time allocated for the development of the project.

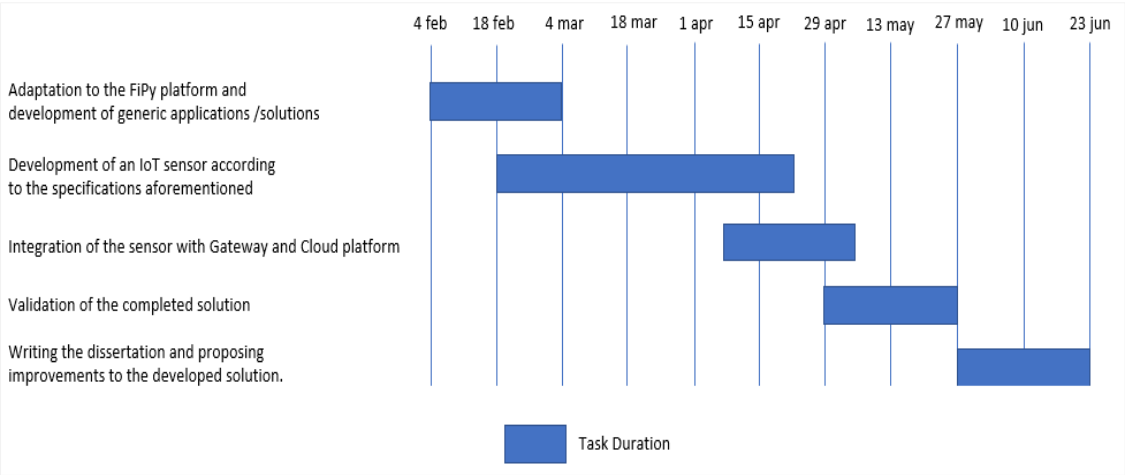

Figure 1 - Gantt Chart with suggested schedule

Chapter 2

Literature Review

2.1 - Internet of Things (IoT)

This chapter presents the literature review requested in the thesis proposal. The subject at hand was extensively researched, using varied platforms and databases, and is intended for the reader to get elucidated and enlightened on the topic in order to better comprehend the work that has been done.

The aspects covered in this Internet of Things overview range from a brief introductory history, in section 2.1.1; the definition of concept, in section 2.1.2; a general idea on the most used IoT architecture, in section 2.1.3; an overview of one of the most important IoT enabling technologies, Wireless Sensor Networks, in section 2.1.4; a succinct look on IoT application areas, in section 2.1.5; and some of the major challenges in IoT implementation, in section 2.1.6.

2.1.1 -History

"*Today computers, and, therefore, the Internet, are almost wholly dependent on human beings for information. (…) If we had computers that knew everything there was to know about things, using data they gathered without any help from us, we would be able to track and count everything and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling and whether they were fresh or past their best.*" – Kevin Ashton, 2009

The term, and now popular buzzword, "Internet of Things" first gained life as the title for a presentation made for Procter & Gamble (P&G) back in 1999, in the context of supply chain management. Kevin Ashton, the then cofounder and executive director of the Auto-ID Centre in MIT, wanted to attract attention to the linkage between a new technology called radio-frequency identification (RFID) and the Internet, as he believed it could allow computers to understand the world around them without human input.

In 2005, the term gains more traction as it is acknowledged by the UN's International Telecommunications Council (ITU) in the first published report on the topic. This report accredits IoT as being a "new dimension" in "the world of information and communication" (6) and mentions RFID as an important enabling technology for IoT.

2008 marked the jump in notoriety and popularity of the expression, both in Europe and in the US. The first European IoT conference was held in Zurich with the purpose of facilitating

the sharing of information and applications by the leading researchers. Meanwhile, the US National Intelligence Council (NIC) has listed IoT as a "Disruptive Civil Technology" with the potential to "contribute invaluably to economic development and military capability"[16] while the FCC (Federal Communications Commission) approves the use of white space spectrum.

In 2010, the Chinese government announced major investments of 5 billion yuan, in the industry of IoT, for their next Five-Year plan [17].

Finally, in early 2014, IoT takes off as Google announces the acquirement of Nest, a smart home company, for 3.2 billion USD.

2.1.2 -Concept

The main idea behind the concept of Internet of Things (IoT) is an expansion on the concept of "Internet", by associating it with the term "things". In doing so, it broadens the spectrum of devices in which the Internet is used, transmitting the notion of a ubiquitous widespread network. As we know it, Internet is "A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols."[18]. However, in this new paradigm of IoT, the referred global network is no longer exclusive to computers. Now, everything is connected. Every "thing", from the smallest wearable to the bigger appliances, even something not associated with electronics at all, such as a piece of clothing will be able to interact with each other. To put it best, "IoT will connect objects around us (…) to provide seamless communication and contextual services provided by them…"[7].

So, the new era of IoT will revolutionize the way we perceive and utilize everyday objects, as the new connected products will have capabilities that transcend the boundaries we are familiar with. These cutting-edge "smart" objects will be equipped with technology that will allow them to be identified, to sense, to process information and to communicate in their network to achieve a common and beneficial goal.

Currently, there are an estimated 4.1 billion Internet users in just over 200 countries, which accounts for over 53% of the world's total population [19]. With the world as connected as it ever was, and considering the major impact the Internet has had in our society, then we can expect that the IoT will be the next step in its evolution, by "taking a huge leap in its ability to gather, analyse, and distribute data that we can turn into information, knowledge, and, ultimately, wisdom"[20].

Companies like Gartner and Statista estimate the number of IoT devices to be between 20-30 billion in 2020, in a market currently sized at 1.7 trillion USD [14, 21, 22]. Thus, in the upcoming years, IoT development will continue to advance as programs like the HORIZON 2020 push for the evolution of such technologies, by funding research projects. In 2016, 6 of those were selected for funding as part of the IoT European Large-Scale Pilots Programme [23].

Nevertheless, there are still some impediments to address in other for this new paradigm to be in full effect. Security, privacy and scalability being a few of them. The creation of a new connected system based on developing enabling technologies is not without its challenges, as some applications have some very specific requirements for IoT integration.

2.1.3 -IoT Architecture

IoT systems do not work as a single unique technology. Instead, they are formed by a compilation of various developing technologies, consisting of a multitude of heterogeneous devices that work collectively as a unit – sensors, actuators, gateways, cloud services, etc.

Its architecture can vary greatly, depending on the requirements of the specific application in which it is used, and it is very important for the effectiveness of the solution. As a result, no one model applies to every one of these systems and different researchers have proposed different architectures, not arriving to a consensus. This section presents a synopsis not only on the different functional/building blocks that comprise IoT, but also on a few of the most well-known and accepted IoT architectures.

i. IoT Building Blocks

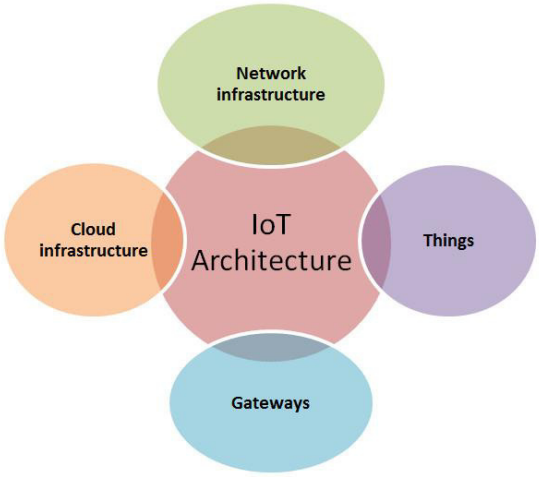

Figure 2 - IoT Building Blocks, extracted from [24]

IoT architecture encompasses four main building blocks – Things, Gateways, Cloud infrastructure and Network infrastructure as seen in Figure 2. Their function is to expedite the various functions of the system, such as sensing, actuation, communication, management, etc. Grasping how each one is prone to failure and which technologies are best suited to mitigate potential issues is crucial to achieve a well-designed, easy to manage, secure system [24]. Table 1 includes further explanations on each one of the main functional blocks.

Table 1 - IoT functional blocks

Functional Blocks	Features
Things	Devices that can exchange, collect and process data either locally or send it to servers or cloud-based services. May contain I/O interfaces, storage, Internet connectivity interface.
Gateways	Provides a safe connection between IoT devices and remote servers for cloud-based services.
Cloud Infrastructure	Provides the analytical and computing power to the system.
Network Infrastructure	Provides the control and security over the data flow. Comprised of routers, aggregators and repeaters.

ii. IoT Architectural Layers

There are several layer-based architectures proposed, some of the most well-known being the Three-Layer and the Five-Layer architectures.

Three-Layer

Considered one of the most basic models, it is also the most accessible to implement. The three layers perceived by this architecture are: the perception layer, the network layer and the application layer.

i. The *perception layer* is considered the physical layer, which through the use of sensors, is able to acquire data from the surrounding environment and also detect other smart devices.
ii. The *network layer* is responsible not only for the connectivity between sensing devices, servers and other network equipment, but also processing sensor data.
iii. The *application layer* provides application specific services to the end user. Also, it defines the various application fields where IoT can be deployed (e.g. smart home, e-health, etc.).

This model, despite its usefulness in describing the main concept of IoT, does not suffice in terms of more detailed aspects of the technology. One of the solutions proposed by researchers is to add more layers that indicate more precisely the numerous stages of an IoT process.

Five-Layer

This particular architecture adds two more layers to the more primitive model. They are the processing and business layers.

i. The *processing layer* can be considered as a middleware that features processing and analytical capabilities. It makes use of databases, cloud computing technologies.
ii. The *business layer* is responsible for managing the entire system, including business and profit models and the privacy of its users.

In this model, the network layer in the Three-Layer architecture is divided into the transport layer and the new processing layer, allowing for a better separation of the functionalities of each layer.

2.1.4 -Wireless Sensor Networks

Internet of Things and Wireless Sensor Networks (WSN) are concepts that can be very closely associated. WSNs take an essential role in an IoT implementations as they provide sensing capabilities that act as the source of data upon which the whole system will act.

iii. Sensor Node

Wireless Sensor Networks consist of a network of very small devices called nodes. The nodes incorporate an embedded CPU with limited storage, a communication unit, a power supply, and smart sensor unit as seen in Figure 3.

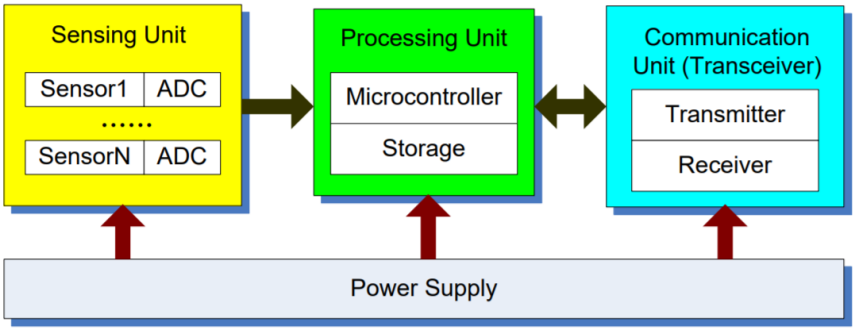

Figure 3 - Sensor Node, extracted from [6]

The Sensing Unit can be comprised of one or multiple sensors that are used to measure relevant environment data such as humidity, temperature, vibration, etc. These are usually associated with an Analog-to-Digital converter in order to send the information to the CPU. New technologies have been developing over the past years, that will favour the way IoT can penetrate different, more diverse markets [11]:

- Micro-electro-mechanical systems (MEMS) (e.g. gyroscopes, acoustic sensors, accelerometers, smoke sensors, magnetometers, chemical sensors, pressure sensors and piezoelectric sensors).
- CMOS-based sensors (e.g. chemical composition, humidity, temperature and capacitive proximity sensors).
- LED sensors (e.g. chemical composition, proximity, and ambient light sensors.

The Processing Unit consists of a Microcontroller coupled with some storage space. Its function is to do some limited data processing and to control the other components of the node.

The Communication unit is important and necessary to keep a centralized system. Its purpose is so to establish a wireless connection to other nodes in the network using a transceiver for both receiver and transmitting functions.

iv. Architecture

The sensor nodes will act in a cooperative way, sensing the surround medium and sharing information between them and the main servers. A WSN architecture typically consists of 3 main components: the sensor nodes, Gateways and the user. The nodes and the Gateways form a sensor field which is interconnected with the user through the Internet, as seen in Figure 4.

The information flow starts in the Sensor Unit of the nodes, where it is first gathered. Subsequent processing is done in the Processing Unit. It is further sent to the Communication Unit where it is sent, via wireless channels to the designated Gateway.

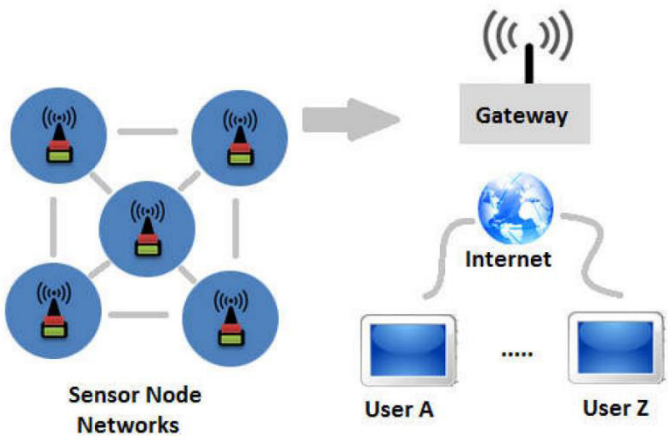

Figure 4 - WSN Architecture, extracted from [11]

The Gateway serves as an interface between the nodes and other devices more easily accessed by the users, such as computers or mobile phones. They can be in one of three different states: active, passive or hybrid. The active Gateway will let its nodes send data freely to the servers. The passive Gateway will first send a request to the nodes, and only then can the information be sent. The hybrid one allows for both.

Connected to the Gateways, via Internet is a platform called Task Manager that will grant users and system admins access (locally and/or remotely) to all the acquired information. The Task Manager main functions are to store, and further process data while also servicing as a data retrieval system.

2.1.5 -IoT Application Areas

Internet of Things has developed to be one of the most appealing and impactful technologies, that has the capability of transforming every facet of our everyday lives. Its versatility and adaptability allow for a vast array of application domains that can span practically every industry as well as other areas such as domotics, healthcare, agriculture, retail, etc.

Despite having such an ample set of application domains, the focus of this section will centre around four of the biggest areas of development that have become more accessible and commonplace over the past few years: agriculture, smart cities, smart homes and industry 4.0.

i. Agriculture

It is in the agricultural sector where WSNs can really shine, by providing much needed data to help advance the techniques used and improve the processes prevalent today. With the integration of IoT applications in the sector, it is implied a transformation of infrastructures and an update to new and improved tools and machinery, with the ultimate goal of empowering farmers and granting them knowledge and mechanisms to further enhance their production at diminished costs [25]. The term “precision agriculture” refers exactly to this ability to, through environment and crop monitoring, administer the correct amounts of resources (water, fertilizer and/or pesticides) and avoid unnecessary wastes usually caused by traditional methods. Proven results have shown in Germany, as herbicide and fuel consumption have been reduced by 10% to 20% [26].

Today, smart farms should be suited for the sensing of multiple relevant variables (soil moisture, light, humidity, temperature, etc.) and offer software solutions capable of analysing the huge amounts of data generated. They should also rely on the connectivity of IoT for both machines, by contributing to the integration of different systems (irrigation, farm equipment, weather analysis, etc.); and the farms themselves, by facilitating the acquisition and sharing of data from multiple sources. This is creating a new paradigm that introduces new business models and a renewed competition ecosystem, as new actors and roles enter the value chain, in a way that demands adaptation in order to remain competitive [27].

New solutions keep entering the market as technologies develop into final marketable products, become more easily available and most importantly – cheaper. An example of such a solution is WiseCrop [28]. Developed in Portugal, the company presents an operating system that allows the remote monitoring of information regarding the crops, equipment, machinery and irrigation systems, in addition to solutions that control different actuators (irrigation, climatization, motors, etc.).

ii. Smart City

As societies expand and develop, urbanization levels will inevitably increase and, as a consequence, already present problems become even more evident. Everything relating to management and monitoring at larger scales will benefit immensely from IoT. So, the aim of these projects is to mitigate the damage caused by over population and direct efforts towards trying to find new original solutions that have a positive impact on our environment and our social and economic conditions.

The term “smart cities” is remarkably broad and can encompass a very diverse range of settings and conditions for IoT projects to focus on. With that in mind, and resorting to a graph, present in Figure 5, taken from an IoT analytics report about IoT segments in 2018, it is easier to classify the multiple projects into 7 major areas.

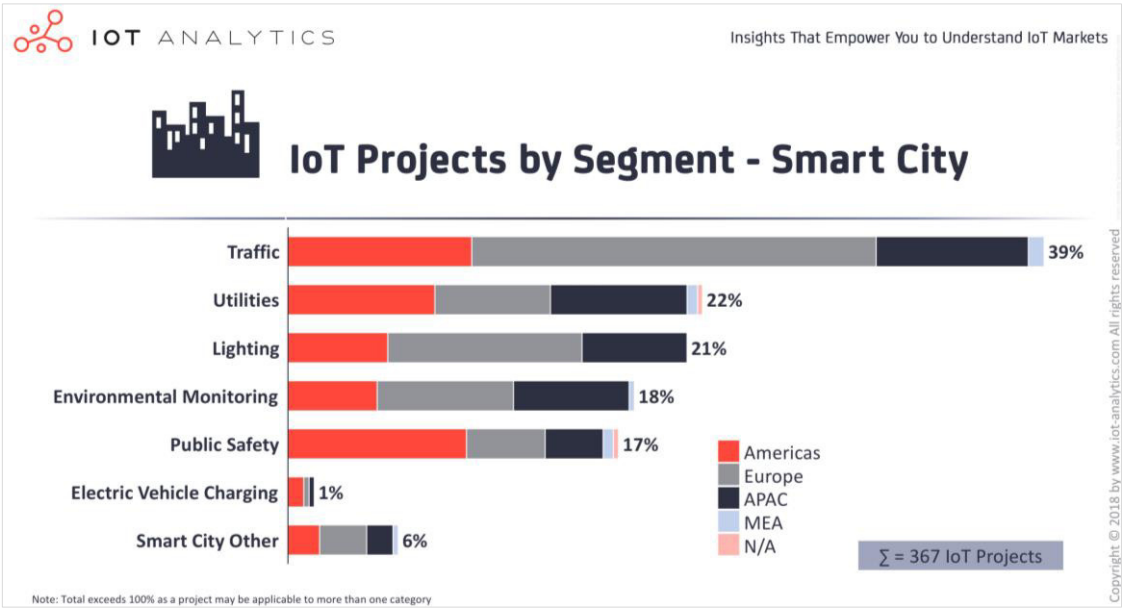

Figure 5 - IoT Projects by Segments, extracted from [5]

In the last couple of years, projects in smart cities have increased in such a way that is has now become the segment with the largest number of projects. In the graph, it is possible to observe a greater interest in projects relating to Traffic, as well as Utilities and Lightning, followed closely by Environmental Monitoring.

The trend is to continue developing smarter communities in both cities and rural areas and take advantage of innovative technologies that are more focused on the human aspects, to increase quality of life for the population, by providing safer and healthier environments. Some changes will occur in the way we interact with each other and in the processes of decision making we partake in every day. In brief, more and more data is being harvested about our preferences that could be used to guide our decisions. In addition, this increased connectedness will help boost collaboration and engagement in social affairs.

iii. Smart Home

In recent years, the popularity of smart home devices has reached a point where its accountable for a 4.5 billion USD share of a 351 billion USD industry [29]. IoT technologies, in the context of smart homes, introduce automation and control to facilitate and push for better, more comfortable living conditions, in accordance to each person's lifestyles.

These conditions include the still prevalent necessity to improve energy efficiency which has led most researchers to focus on monitoring and managing systems or HEMS (House Energy Management System).

Proposed projects on that field include Smart HVAC systems [30] that adapt the homeowners' comfort level while still reducing energy use; Self-Learning Home Management Systems (SHMS), that use machine learning capabilities to optimize the energy consumption by prioritizing the most used devices in a given hour [31]. Such complex systems take advantage of a multitude of smart devices (plugs, appliances, etc.) as the source of data for thermal profiling and energy consumption rates.

iv. Industry 4.0

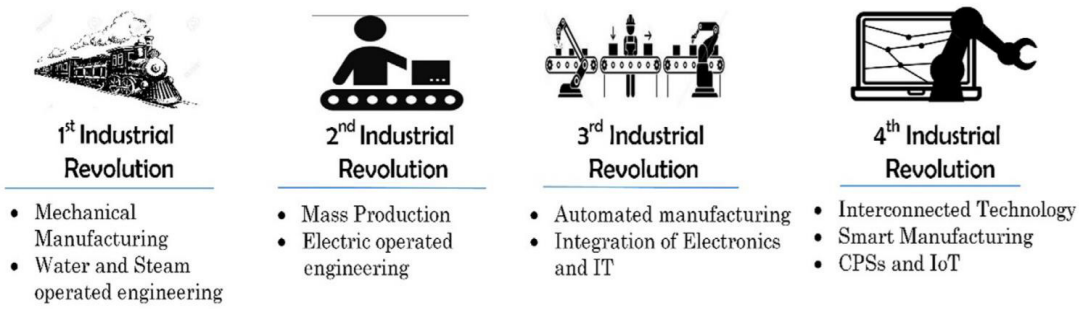

Figure 6 - Evolution of Industrial Revolution, extracted from [9]

Industry 4.0 refers to the most recent revolution in the industry segment, in which a complete transformation takes place. It involves connectivity and integration with the digital intelligent realm of IoT, where robotics and automatization are the main aspects. It is necessary for these systems to be constantly connected, with human assistance and able to make decisions in a decentralized manner. The major components of Industry 4.0 include cyber-physical systems, cloud computing, big data analytics, additive manufacturing, etc.

Coupled with the surge of this new paradigm comes a greater awareness and concern for energy efficiency, both in terms of a consumer behaviour shift that favours more environmentally friendly practices and as it could also prove to be an enormous advantage for businesses, since energy costs are on the rise and more energy efficient manufacturing processes [32] would be beneficial for profits in the long term.

2.1.6 -Challenges and Impediments

By assembling various technologies, IoT can be a very complicated concept that entails many issues in its attempt at seamlessly integrating them. This section focuses on some of the implementation challenges IoT is currently facing while briefly describing each one.

i. Energy Consumption

IoT energy problem is a crucial one. IoT devices are in the billions, spread across multiple networks around the world, and since most of them are battery powered or use energy harvest methods, it is of the utmost importance that only the necessary data is transmitted. It is unfeasible to change batteries on all of these devices, so if such implementations are to be sustainable on the long run then energy savings is a primal concern.

ii. Devices/Links Heterogeneity

IoT implementations employ a wide variety of devices and connectors that will work with different protocols, data formats, etc. that, most likely, will not be compatible with one another. Such heterogeneous applications imply challenges that are unique to IoT and to the additional technologies involved. It is important that a “one-size-fits-all” kind of model is able to consolidate and merge such an array of devices into a uniformized selected few, in an attempt to standardize the technology and make it more accessible to work with.

iii. Security

Security in IoT is one of the main challenges that is still being worked on by researchers and developers. Both privacy and trust are to be taken into account when designing a new IoT solution, due to the vulnerability of connecting the systems to established network infrastructures. Four major security related challenges can be seen in Figure 7.

Figure 7 - IoT security challenges, extracted from [12]

iv. Massive Scaling

Scalability requires interoperability between different systems. The massive amounts of nodes deployed, and the huge amounts of data generated will be a challenge not only in terms of cloud and network capacity, but also in the routing algorithms that will be used. The introduction of IPv6 as come as a way to mitigate some of the existing addressing issues, as IPv4 is quickly running out of available addresses, but being a very recent technology, should be used cautiously, letting development teams learn the details and adapt to them.

2.2 -Communication Protocols

Essential to a correct implementation of IoT solutions is the communication protocols that ensure correct connectivity in the system. The protocols define data formats, data encoding, routing and addressing. Some also include additional capabilities of sequencing and flow control. Below, Table 2 presents a broad comparison between different well-established and commonly used protocols.

Table 2 - Comparison between existing communication protocols [10]

Table 2 Comparison of the existing communication technologies.

Parameters	WiFi	WiMAX	LR-WPAN	Mobile communication	Bluetooth	LoRa
Standard	IEEE 802.11 a/c/b/d/g/n	IEEE 802.16	IEEE 802.15.4 (ZigBee)	2G-GSM, CDMA 3G-UMTS, CDMA2000 4G-LTE	IEEE 802.15.1	LoRaWAN R1.0
Frequency band	5–60 GHz	2–66 GHz	868/915 MHz, 2.4 GHz	865 MHz, 2.4 GHz	2.4 GHz	868/900 MHz
Data rate	1 Mb/s–6.75 Gb/s	1 Mb/s–1 Gb/s (Fixed) 50–100 Mb/s (mobile)	40–250 Kb/s	2G: 50–100 kb/s 3G: 200 kb/s 4G: 0.1–1 Gb/s	1–24 Mb/s	0.3–50 Kb/s
Transmission range	20–100 m	< 50Km	10–20 m	Entire cellular area	8–10 m	< 30 Km
Energy consumption	High	Medium	Low	Medium	Bluetooth: Medium BLE: Very Low	Very Low
Cost	High	High	Low	Medium	Low	High

This section will focus not only on IoT centered LPWANs (LoRaWAN, SigFox, RPMA, Weightless) offering a detailed description on how each one of them works, but also on the set of IEEE 802.11 protocols (WiFi) featured in this project, with special attention to two particular ones: WiFi HaLow and the brand new WiFi 6. In addition to these, MQTT will also be mentioned as a higher-level protocol that takes part in the communications of the developed application, due to its favourable characteristics.

2.2.1 -Low-Power Wide-Area Networks (LPWAN)

LPWAN describes a new communication standard that offers long-range communications between low-power, low data rate devices. As a result, it proves to be of great value to IoT applications as it aims to fulfil the prerequisites that the usual legacy technologies are not able to. However, these protocols cater only to a portion of the IoT market: applications that are

delay tolerant, whose data rates and power consumption can remain low for correct operation and whose costs tend to be on the lower end.

On the market today are numerous different LPWAN technologies, each applying different methods and techniques to service a diversified lot of IoT arrangements that may require long range communications in a wide coverage area.

For instance, it this sub-chapter, LoRa [33], SigFox [34], RPMA [35] and Weightless [36] will be discussed with the purpose of bringing out the differences between them and where each one can be better applied to satisfy the needs of the user.

i. LoRa and LoRaWAN

LoRa employs a unique Spread Spectrum (SS) modulation that is akin to Chip Spread Spectrum (CSS) and couples it with Forward Error Correction (FEC), in order to avoid interference between channels with different data rates. Adding to that, LoRa technology assigns:

- Different orthogonal Spreading Factors (SF) that make possible for several packets with different SF's to be sent simultaneously on the same channel, improving efficiency;
- Different Coding Rates (CR) which enable the function of FEC and, depending on the value, can increase or decrease the robustness of LoRa transmissions.

With this implementation, LoRa makes up for low data rates by allowing for robust and highly sensitive networks that along with a high level of customization for the user.

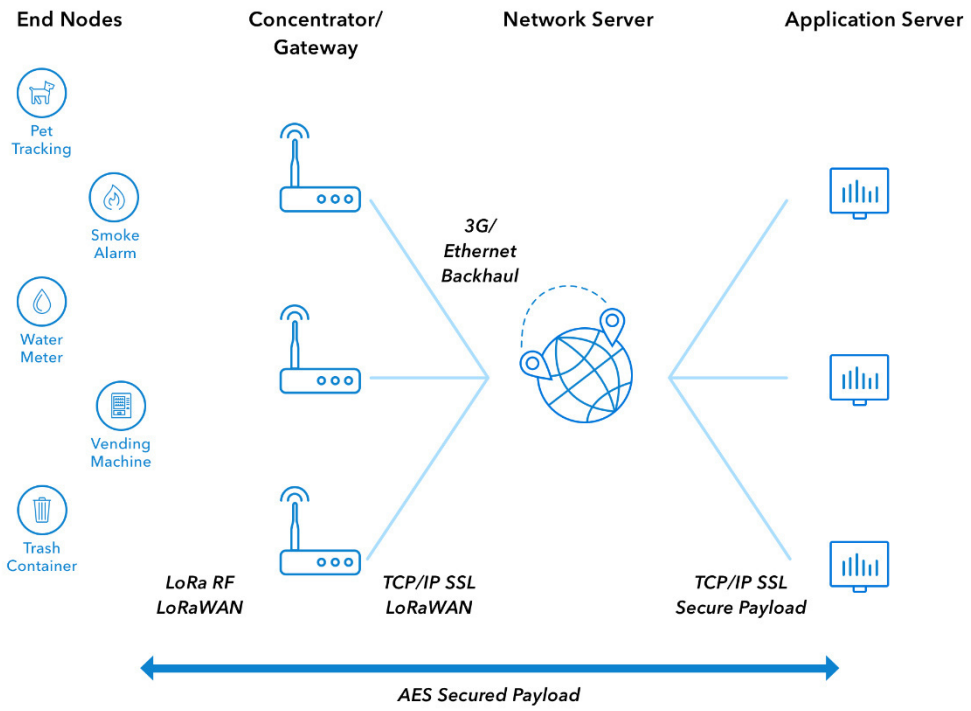

Figure 8 - LoRaWAN network topology, extracted from [1]

LoRaWAN is a cloud-based media access protocol (MAC), that builds upon the lower physical (PHY) layer LoRa protocol, proprietary to Semtech Corporation.

LoRaWAN networks, as seen in Figure 8, follow a topology in which gateways transmit messages between end-devices and a central Network Server, that then forwards them to the respective Application Server.

End-devices use single-hop LoRa or FSK to communicate to one or many gateways, while the gateways work with standard IP connections to the Network Server. The communications are usually bi-directional, though the main source of traffic is from the end-devices to the Network Server.

LoRaWAN devices are categorized in three types: Class A, Class B and Class C (refer to Figure 9). All end-devices must implement functionalities off Class A, which corresponds to the basic level of LoRaWAN, while Class B and C represent additional features.

Class A devices incorporate bi-directional communications to gateways. The transmission of uplink messages from the end-devices is permitted at any time. It is followed by two short timed receive windows at specified times. If there is no response from the server in those timeframes, only after another uplink transmission will there be another chance to do so. The server is only allowed to communicate in one of those windows, never on both.

Class B end-devices expand on Class A devices by including extra receive slots. These windows for downlink communication will open at scheduled times, by receiving a time synchronized beacon from the gateway.

Class C end-devices have the maximal receive slots. The receive windows in these devices are kept continuously open until the following uplink transmission. Therefore, these devices will have the lowest latency communications despite using more energy to operate than both preceding classes.

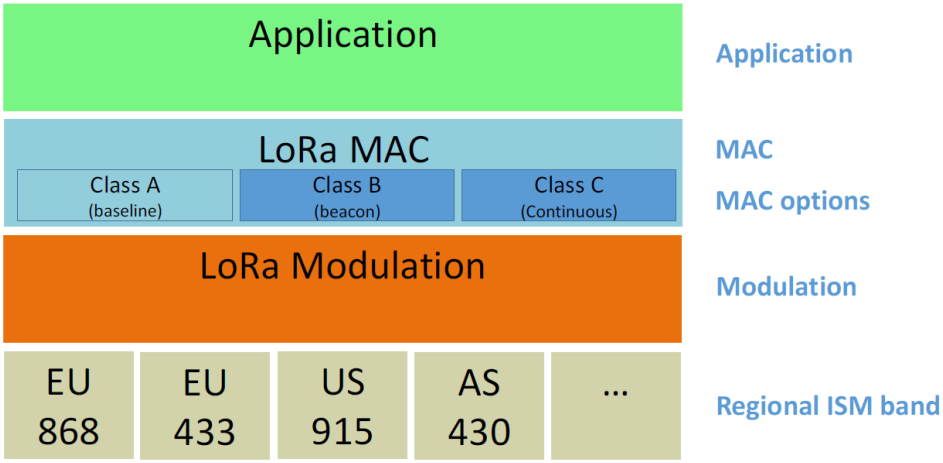

Figure 9 - LoRaWAN Classes, extracted from [3]

To operate, LoRaWAN uses radio frequencies present in unlicensed radio spectrum – the ISM band. These frequencies are reserved for industry, scientific and medical use, therefore LoRaWAN avoids large transmission fees. However, throughout the world, some countries impose restrictions to the usable band of frequencies that disturb the attempts at uniformizing communications. Consequently, there are band specifications for some particular regions as seen in Table 3.

Table 3 - Regional frequency band specifications

Region	Frequency Band (MHz)
Europe	863-870
United States	902-928
Australia	915-928
China	779-787 and 470-510

Between the limits of each frequency band, LoRaWAN defines multiple sub-bands. For Europe, these are defined in section 7.2.3 of the ETSI standard EN 300 220 [37]. The standard also defines maximum transmit power of +14 dBm, allowed spreading factors, bandwidths, along with other important parameters/metrics and with specific duty-cycle conditions for each sub-band, as follows:

- g (863.0 – 868.0 MHz): 1%
- g1 (868.0 – 868.6 MHz): 1%
- g2 (868.7 – 869.2 MHz): 0.1%
- g3 (869.4 – 869.65 MHz): 10%
- g4 (869.7 – 870.0 MHz): 1%

As for data rates, they depend on two variables: the bandwidth that is used and the aforementioned SF, that can change between 7-12 for Europe and 7-10 for North America. Channel bandwidth used in LoRaWAN can vary from 125 kHz, 250 kHz and 500 kHz, and that is dependent on the region and the frequency plan. Thus, data rates range from 0.3 kbps to 50 kbps.

ii. SigFox

SigFox is a technology that is based on Differential Binary Phase-Shift Keying (D-BPSK) modulation and takes advantage of its easier implementation, high efficiency and very robust and noise-resistant characteristics.

For transmissions between nodes, it uses an Ultra-Narrow Band (UNB) that specifies a wide-reaching but very low power consuming signal, whereby messages have a fixed bandwidth of 100 Hz and are transmitted with a data rate of 100 bps (Europe) or 600 bps (United States) through the unlicensed ISM frequencies, in the region of 868 MHz (Europe) and between 902-928 MHz (rest of the world). The band is only 192 kHz wide, so in the case of Europe, it only spans from 868 MHz to 868.2 MHz. Transmissions in SigFox are bi-directional, though downlink messages from the Gateways are limited to 4 messages per day and uplink messages are limited at 140.

The lightweight protocol employed is specially designed and is limited to a size of 12 bytes per payload. Such a payload would be capped at 26 bytes when including the protocol overhead. Compared to conventional protocols, it requires much less data to send and less energy to do so. As a result, it becomes ideal for battery-powered devices and allows for a higher network capacity.

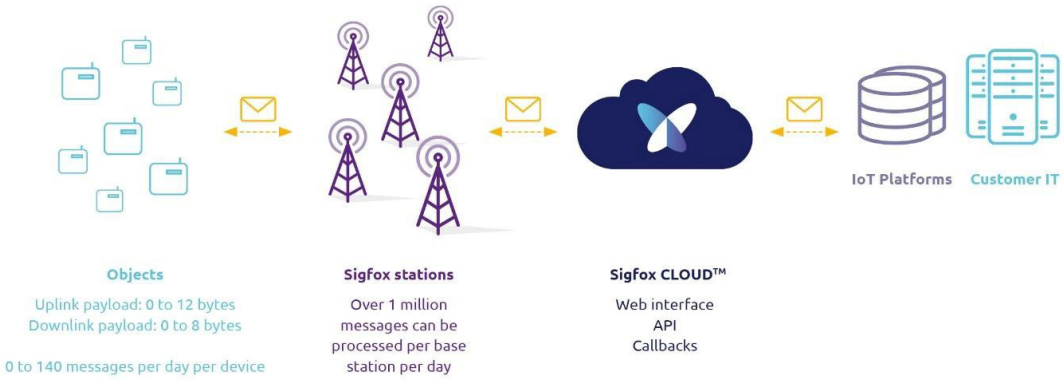

Figure 10 - Overview of SigFox Architecture, extracted from [8]

Figure 10 shows a general SigFox architecture. It follows a familiar model albeit with some differences. It is composed by multiple end-devices and SigFox's own gateways (Base Stations) and Cloud Services to communicate with the users' final platform or application. The SigFox Cloud is crucial in the implementation and correct operation of the systems, as it includes features such as: Base Station management, monitoring and message processing; message and modem metadata storage; APIs and web interfaces to allow communication between different actors (Customer IT, end-users) and to grant access, through web browser, to message storage and allow downlink messaging.

iii. RPMA

Developed by Ingenu, this protocol developed with IoT in mind is named after the technology it is based on – Random Phase Multiple Access (RPMA). It features changes in various layers of the stack: from the physical layer (PHY), to the MAC and network layers.

It the PHY layer, the protocol operates using Direct Sequence Spread Spectrum (DSSS). This technique is defined by a parameter named "processing gain" that measures the amount of "spreading" used. As the gain increases so does the sensitivity of the receivers and coverage of a network, but in detriment of data rates. No different than the previous discussed protocols, this one also makes use of unlicensed ISM bands, only this time it in the 2.4 GHz spectrum, using a 1 MHz bandwidth and D-BPSK for modulation.

RPMA uses longer frames than most protocols and, to add, the actual data they hold is surprisingly small so that the receiver sensitivity is achievable. As a result, data rates range from 60 to 960 bps.

iv. Weightless

Weightless is yet another standard for LPWANs, that features usual characteristics such as, low throughput and higher latencies that are favourable for IoT applications. Similarly, it operates in the unlicensed spectrum of sub 1 GHz frequencies, for instance: 138MHz, 433MHz, 470 MHz, 780 MHz, 868 MHz, 915 MHz, 923 MHz, depending on the geographical position.

Communications in Weightless are acknowledged, bidirectional and very uplink oriented, also being optimized for IoT, messages include short payloads of around 48 or fewer bytes. Typical data rates will range from 0.2 kbps to 100 kbps. Ranges are dependent on many

variables including data rate, location and signal patch, link capacity and antenna size and quality. With data rates pending to the lower end, Weightless becomes a comparable alternative to other LPWAN technologies and reaches ranges of about 2 km in urban environments.

At a physical level, Weightless uses Time-Division Multiple Access (TDMA), allotting time slots to each device to multiplex the communications, along with Frequency-Division Multiple Access (FDMA) in 12.5 kHz very narrow bands. For modulation, GMSK and offset-QPSK to increase efficiency and interference resistance.

Weightless network architectures are composed by end-devices, that are low in complexity cost and duty cycle; Base Stations that connect multiple end-devices in a star topology; Base Station Network, that associate every Base Station in a network to control resource allocation, and various aspects relating to the network (authentication, scheduling, roaming).

2.2.2 - IEEE. 802 Protocols – WiFi

IEEE 802.11, commonly known as Wi-Fi, is a set of established wireless communication protocols that have become the most used in today's communications around the world. They specify both MAC and physical layer protocols that enable WLAN implementations.

This series of protocols involve half-duplex communication while applying CSMA-CA (carrier-sense multiple access with collision avoidance) as a contention procedure to avoid collisions in the network, by restricting transmissions to the periods where nodes have sensed the channel as 'idle'.

As for the operation frequency bands, they are dependent on which specific protocol is being used and in which country. In Europe, protocols 802.11b, 802.11g and 802.11n use the 2.4 GHz ISM band, while protocol 802.11a operates on the 5 GHz band. The highest frequency band used in the set is for protocol 802.ad – 60 GHz. It is important to take into consideration that the higher the carrier frequency, the lower the effective range for communication as the waves are more easily absorbed by the surrounding obstacles.

Data rates in Wi-Fi communications range from 1 Mbps (802.11b) up to 6 Gbps (802.11ad) and keep getting higher as new developments arrive to mass markets. The communication ranges vary between 1-10 m for the very high frequency variants, to 250 m for lower frequency variants. All these values suffer fluctuations due to location (geographic and indoors/outdoors) and other implementation particularities (router firmware, placement, etc.).

A recent standard from 2017 emerged that was designed to support IoT use cases and, similarly to LPWANs, would operate on sub 1 GHz frequencies – IEEE 802.11ah. Diverging from the other standards, this one focused less on high data rates and instead was optimized for longer distances (up to 1km) integrating a larger number of lower data rate devices. Consequently, some changes are required to the physical and MAC layers. Based on Orthogonal Frequency-Division Multiplexing (OFDM), this protocol supports bandwidths of 1 MHz, 2 MHz, 4 MHz, 8 MHz and 16 MHz yet, in Europe, only 1MHz and 2 MHz bandwidths are supported, in the frequency range of 863-868.6 MHz. 802.11ah also supports multiple modulation coding schemes (MCS) that will alter the data rates. The referred values are present in Table 4, using only 1 spatial stream.

Table 4 - MCS and data rates for 1 MHZ, 2 MHz and 16 MHz bandwidths and 1 spatial stream [38]

	Modulation	Code Rate	1 MHz (Mbps)	2 MHz (Mbps)	16 MHz (Mbps)
MCS 0	BPSK	1/2	0.3	0.65	6.5
MCS 1	QPSK	1/2	0.6	1.3	13
MCS 2	QPSK	3/4	0.9	1.95	19.5
MCS 3	16QAM	1/2	1.2	2.6	26
MCS 4	16QAM	3/4	1.8	3.9	39
MCS 5	64QAM	2/3	2.4	5.2	52
MCS 6	64QAM	3/4	2.7	5.85	58.5
MCS 7	64QAM	5/6	3	6.5	65
MCS 8	256QAM	3/4	3.6	7.8	78
MCS 9	256QAM	5/6	4	N/A for 1 spatial stream	86.7

New and improved Wi-Fi standards keep being introduced as new advancements in the technology are reached, allowing for better data rates and better efficiency. In 2019, it is expected the arrival of Wi-Fi 6, by Wi-Fi Alliance, that will ensure interoperability between devices based on the new 802.11ax standard [39]. The following Table 5 will conclude this overview by elucidate on the various standards of the 802.11 family and displaying the absolute maximum data rates they allow for.

Table 5 - IEEE 802.11 standards

Standard	Frequency (GHz)	Data Rates (maximal)
802.11a	5	54 Mbps
802.11b	2.4	11 Mbps
802.11g	2.4	54 Mbps
802.11n	2.4/5	600 Mbps
802.11ac	5	3.4 Gbps
802.11ad	60	6 Gbps
802.11ax	2.4 and 5	11 Gbps
802.11ah	sub 1 (ISM bands)	347 Mbps*

* 16MHz channel bandwidth, 4 spatial streams, 256QAM, 5/6 coding rate

2.2.3 -Message Queue Telemetry Transport (MQTT)

MQTT, or Message Queue Telemetry Transport is a very simple messaging protocol that is used on top of TCP/IP protocol. It uses a publish/subscribe architecture of accessible implementation, whereby clients subscribe to topics thus making them subject to the messages published to the respective topic. Such a model supports as far as thousands of remote clients in a single server. Necessarily, MQTT becomes an optimal fit for constrained networks with low bandwidth, or high latency consisting of devices with restricted processing power and memory.

Because MQTT minimizes bandwidth and resource requirements while maintaining network reliability it becomes notably appropriate for IoT implementations, mainly in WSNs.

The protocol is based on an assortment of basic notions that tries to ensure accurate message transmission, while at the same time trying to keep the message size as short as possible:

- This protocol admits three **quality of service** (QoS) levels, each level representing more effort to ensure reliable message delivery, at the cost of a higher bandwidth usage. MQTT also admits;
- MQTT **retains messages**, despite having sent them to every available subscriber. This allows new subscribers to the topic to receive retained messages;
- MQTT establishes **clean sessions** and **durable connections**. That means that when a new client connects, an optional clean session flag is created. If the flag is set to true, upon disconnection, all subscriptions are removed. If it is set to false, the connection is considered durable. As such, subscriptions remain after disconnection.
- Clients in MQTT have **wills.** They are used to inform the server that the particular client has a message it wants published to one or more topics should an unexpected disconnection occur.
- This protocol is often compared with HTTP/HTTPS, the most widely used protocol that is also based on TCP/IP. The following Table 6 shows a comparison between both protocols in an attempt to further elucidate on the advantages MQTT for IoT implementations.

Table 6 - Comparison between MQTT and HTTP [7]

	MQTT	HTTP
Design orientation	Data centric	Document centric
Pattern	Publish/subscribe	Request/response
Complexity	Simple	More complex
Message size	Small, with a compact binary header just two bytes in size	Larger, partly because status detail is text-based
Service levels	Three quality of service settings	All messages get the same level of service
Extra libraries	Libraries for C (30 KB) and Java (100 KB)	Depends on the application (JSON, XML), but typically not small
Data distribution	Supports 1 to zero, 1 to 1, and 1 to n	1 to 1 only

Chapter 3

Architecture and Specifications of the System

The following chapter intends on presenting, in a clear but comprehensive way, the architecture used in this system and the specifications of this particular setup.

The aim of this section is to provide the reader with all the necessary information about the structural organization in place, starting with a broader overview, and progressively homing in on the details of each component and the interactions between them, all the while directing the spotlight at the main focus of this project – the Sensor Node. By understanding the building blocks in this IoT arrangement, then more specified explanations further down the report, on the implementations developed, will be made simpler and easier to conceive.

This chapter will be divided in the following way. First, in Section 3.1, an architecture synopsis will be given, with visual support to better visualize the concept as a whole. Section 3.2 will present the concluding remarks regarding the implementation.

3.1 - Architecture Overview

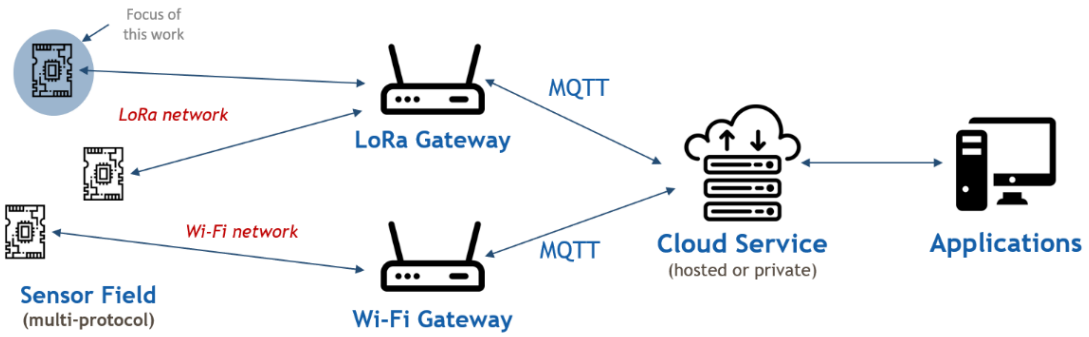

Figure 11 - Complete architecture of an end-to-end IoT solution with multi-protocol sensors.

The development of this project made use of an IoT-centered framework already in development by Controlar. It is mainly built upon open-source hardware and software solutions, that were adapted to fit the requirements established by the company, with the goal of realizing a low-cost, multi-protocol IoT system.

Starting with an overall view, Figure 11 depicts a schematic that portrays the composition of this IoT arrangement. With its aid, we can more clearly describe the operation of the system.

First off, the Sensor Field, comprised by Sensor Nodes, will contribute with the sensing capabilities. They are the ones who gather all the data to be processed further down the pipeline. Next, through either LoRaWAN or WiFi networks, the measurements acquired will be sent by the Nodes to the dedicated Gateway. In this setup, the Gateway is considered to be active, so the packets containing the data can be forwarded, without restriction, to a centralized Server/Cloud Service. Finally, to visualize obtained data, several different Applications can be integrated with the system.

In the following subsection 3.1.1, the specific components utilized will be listed and further detailed, making sure the reasons for the choices made are identified. Then, subsection 3.1.2 will delve into the various aspects concerning the microchip that was essential for the Sensor Node itself and what makes it ideal for this purpose.

3.1.1 -Main Components

There are two main reasons that accredit such a description of the components and documentation of their behavior. The first one, is to facilitate the differentiation of this system and pick it apart from others. The second reason is to get a better perception of the conditions that could better suit the specifications of this solution so that it is use can be fully optimized.

Worth mentioning that, as detailed as these descriptions attempt to be, the goal is only to elucidate on the bigger picture this project is inserted into and this was not the focus of this thesis work. As such, questions regarding its development will not be featured in this document, unless they directly relate to the implementations presented.

i. Server/Cloud Service

As the central Server/Cloud, the backbone of the whole implementation, it is employed an open-source version of LoRa Server [6] with specific adaptations for this multi-protocol solution and considering MQTT over TCP [7] communication with the Gateway.

The LoRa Server project is a platform that offers a set of applications to support the different parts that make up an IoT network. One of the specifications it fulfils, for this project, is to be able to send, through the packet-forwarder, the information that arrives at the gateway, from the sensors to a local server.

Figure 12 illustrates the overall architecture of the server and evidences the three-part division that this application undergoes: there is the LoRa-gateway-bridge, the LoRa Server and the LoRa-app-server.

First, the LoRa-gateway-bridge receives messages sent by the gateway, by way of UDP protocol, and then converts to another protocol, JSON over MQTT, capable of being interpreted and understood by the server.

LoRa Server is the central part of this platform. It implements a network server, that is responsible for the manipulation and storage of received data and for downlink communication with end-devices.

The LoRa-app-server provides a set of web interfaces to manage devices, applications and users, as well as an API for external integrations, such as Grafana which will be referred to, later. The app server is also responsible for handling the necessary encryption and decryption.

This server, besides being made up by these three tools, uses two database applications: Redis and PostgreSQL. The first one being used to store non-persistent session-related data and the second for persistent data, e.g., data from FiPy's sensor node.

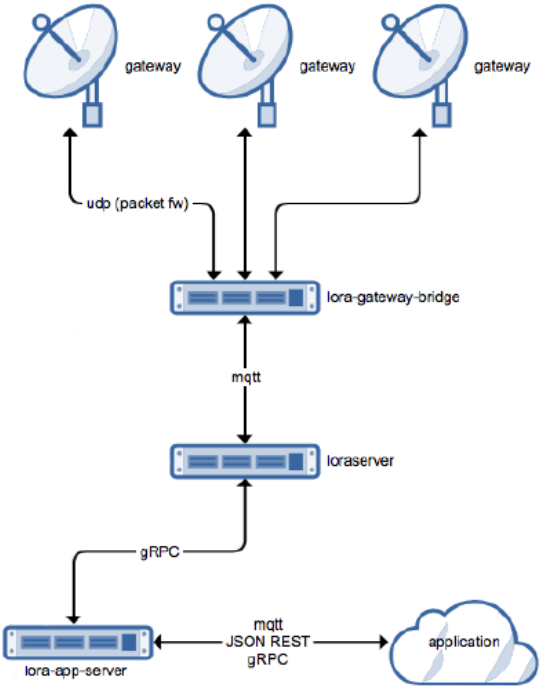

Figure 12 - LoRa Server architecture

ii. Gateway

The multi-protocol gateway adopted for this project is composed by a single entity based on the Raspberry Pi 3 Model B+ with an additional LoRaWAN concentrator board. It uses proprietary software to allow multi-protocol LoRaWAN/Wi-Fi functioning.

It is worth noting that the implemented software in the Gateway will forward all the LoRaWAN packets onto the central server using a compliant MQTT implementation, that will dictate the common baseline that will be followed in both connectivity options (LoRaWAN and Wi-Fi) from the gateway upwards.

iii. Application UI

To better visualize the acquired data, we opted for a free version of InfluxDB along with a free version of Grafana.

InfluxDB is a time series database (TSDB), meaning it is a software that is made specifically to store and organize data by following indexed timestamps. Time series data is ultimately, measurement results that are monitored and stored over time. The purpose of InfluxDB is to detect fluctuations over time, as it analyses large amounts of already sorted information. It is useful to note that time series data allows for summarization and lifecycle management, and it is often required when working with this type of data. We opted for InfluxDB instead of the already integrated PostgreSQL, because it too is of straightforward integration with both LoRa Server and Grafana, only InfluxDB is more suitable and optimized for IoT time sensitive data.

Grafana was used to query and visualize, through dashboards, the acquired data by the sensors. It is a very complete tool that allows for different graph types, the definition of alert

signals for different metrics and for diverse filtering and comparisons of data. For InfluxDB integration, Grafana provides exhaustive query options and other mechanisms to help expedite and simplify the querying and visualization of data.

3.1.2 -The Sensor Node – Microcontroller (MCU)

The focal point of this work is centered around the FiPy, a multi-protocol microchip, and its capabilities as a processing and communication unit, visible in Figure 13 and Figure 14. Coupled with different sensors, it is possible to form very complete modules that are very competent at performing the desired tasks for the implementations at hand. Despite being an expensive piece of equipment (around 54€ [4]), the ease of development, the great online support, the low learning curve and the access to both the desired networks make it the preferred board for prototype boards.

Here, it will be explicit the specifics of this MCU and it will be described in more detail its capacity in terms of the connectivity modules for LoRaWAN and Wi-Fi.

Figure 13 - FiPy Module, extracted from [4]

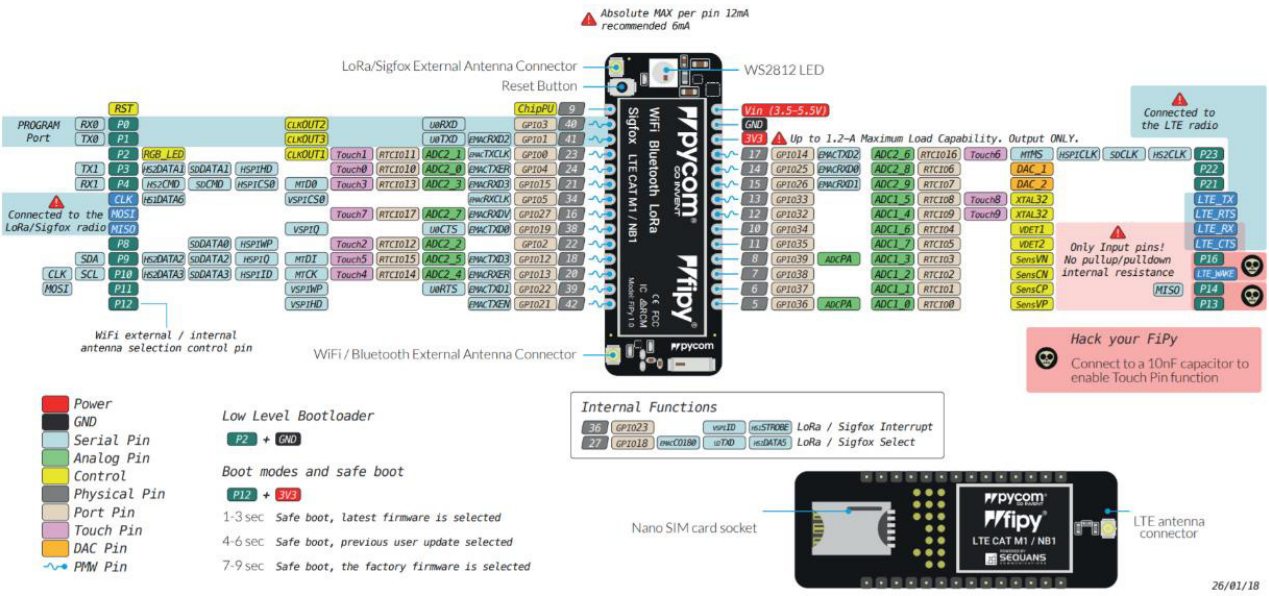

Figure 14 - FiPy Pinout, extracted from [2]

i. Specifications

FiPy features an ESP32 Dual Core SoC as its internal processor, with 520KB + 4MB of RAM and 8MB of external flash (accessible through FTP server). It admits input voltages of 3.5V – 5V, as it includes an internal voltage regulator that reduces it to 3.3V to serve as a voltage output supply for external loads of up to 1.2 A.

For external peripherals, it includes all of the most used protocols: Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I^2C) and Universal Asynchronous Receiver Transmitter (UART) which are remappable to any GPIOs.

The chip's three SPIs allow for 1-line full duplex or 1/2/4-line half-duplex communications, while also supporting four transfer modes (depending on clock polarity and clock phase) and up to 80 MHz of transmission speed. There are two I^2C interfaces (can be either slave or master), with two designated transfer modes (Standard mode – 100 kbps; Fast mode – 400 kbps) with dual addressing modes (7 and 10 bit). Finally, also with three interfaces, UART provides both RS232 and RS485 standards, with communication speeds up to 5Mbps, while also providing hardware management and flow control.

When connecting with analog devices, FiPy offers 12-bit ADCs in multiple I/O's, which introduces a very high level of resolution for signal conversion, very useful in sensitive applications.

In terms of power consumption, the values can be seen in Table 7, varying slightly as some parameters of radio transmissions that differ among the modes, exert their influence. Such parameters include available spectrum band or limited bandwidth, payload sizes, higher capacity (multiple access), security and addressability.

Table 7 - Power Consumption by Feature

Mode	Average Current at 5V Input Voltage (mA)
Idle (no radio)	62,7
LoRa Transmit	156
SigFox Transmit	192
WiFi AP	126
WiFi Client	137
Bluetooth	121

The board is programmed in MicroPython, a microcontroller optimized version of Python 3.5 that allows for faster development. Likewise, for better performance, FiPy has Python multi-thread capabilities. It also takes advantage of Pymakr, a plugin developed for the source code editor – Atom – which enables a console that eases the process of debugging and facilitates internal memory management.

ii. LoRa Module

FiPy's uses a certified Semtech SX1272 as a LoRa module that implements the LoRaWAN stack, v. 1.0.2, and supports operation as in either a Class A or a Class C device, in addition to the possibility of acting as Nano-gateway with capacity for 100 nodes and a range of up to 22km (under ideal conditions). This module supports a spreading factor from 6-12, bandwidth values between 125 – 500 kHz, bitrates ranging from 0.24 - 37.5 kbps and sensitivity extending from -117 to -137 dBm. This LoRaWAN transceiver operates in the g1 range of frequencies (868.0 –

868.6 MHz) mentioned in previous chapters, meaning it is forced to comply to 1% of duty-cycle restrictions.

The protocol was selected as one of the connectivity options for this solution mainly due to its long-range radio coverage and possibility of investigation under industrial scenarios where it is less explored but showing a huge potential.

iii. Wi-Fi Module

For the Wi-Fi connectivity the full capacity of FiPy's module is used and the node's firmware is implemented in such a way that messages can be published and subscribed with an MQTT-compliant communication. Wi-Fi coverage is provided by the multi-protocol gateway, when it is operated in the access point (AP) mode and allowing direct management via a web-based UI.

It is important to make note of the fact that Wi-Fi communication is limited to the 2.45 GHz ISM band, which encompasses only the IEEE 802.11 b/g/n versions.

iv. Other Modules

FiPy also includes other communication modules that were not in use for this project, that will be briefly characterized in this section. As our focus shifted to LoRaWAN and WiFi, there were three modules left unexplored. Those were: Bluetooth, SigFox and LTE-M1/NB-IoT communication modules.

To begin with, FiPy's Bluetooth module is compliant with version 4.2 Basic Rate/Enhanced Data Rate (BR/EDR) for shorter communications with constant data rates and Bluetooth Low Energy (BLE) for longer range communications with short bursts of data. It supports class-1, class-2 and class-3 transmitters without an external power amplifier with receiver sensitivity of up to -97dBm. Next, the SigFox module, which operates in the frequency range of 868.13MHz – 869.52MHz (Europe), allows for data rates from 100 – 600 bps. The receiver sensitivity extends to -126 dBm.

Finally, LTE-M1/NB-IoT features one of the latest releases by 3GPP, for the next generation of cellular communication standards – release 13, also known as LTE Advanced Pro. It supports 12 frequency bands from 699MHz to 2690MHz, for worldwide coverage. Likewise, narrowband LTE ue-Categories M1/NB1 support is included, with data rates ranging from 55 kbps to 300 kbps. Only available certified carrier is Verizon US.

3.2 - Concluding Remarks

So, the architecture is comprised by three distinct elements, from the node up. There has to be seamless integration between the different parts to assure correct operation by the whole system. In the node itself, as the central part of the workload, special attention was given to the MCU, which is a very powerful tool for developing and prototyping IoT applications and whose capabilities required further explaining and detailing. Finally, to really grasp the potential of this solution, a coverage result was performed, and the outcome showed promising results in the effectiveness of this solution in industrial environments and propelled the evolution of this solution with the already selected tools.

In the following chapter, to better understand the implementation that will be supported by the structure we now discussed, every component entailed in its development will be thoroughly explained.

Chapter 4

Implementations

The bulk of this project was regarding the implementation of an energy metering solution that takes in several measurements relating to energy consumption of industrial machinery, which required a considerable large amount of work.

The energy meter demanded more thought and more effort in designing the circuitry needed and, in that sense, there was also more research and problem solving involved. As a result, a more polished finished device is expected with positive perspectives of becoming a fully-fledged final product capable of being aggregated with future Controlar machines.

This chapter will contribute with in-depth characterization of this application. In section 4.1 everything with respect to the Energy Monitor and its performance will be explicit.

4.1 - Energy Meter

Developed with the intent of enabling control and monitoring over the power consumption of industrial machines, this single-phase energy meter will feature a sensor developed by Microchip, the ATM90E26 [40, 41]. This sensor is capable of taking in current and voltage measurements for sampling circuits and obtain several. Gathered data will then be communicated, in an architecture like the one previously mentioned, through to the gateway and on to the server; where integrated applications will provide visualization options for end users to access information that will serve as a prolific tool in understanding resource allocation for manufacturing processes.

This section will present information needed to understand the development process and the reasons behind the choices and decisions made throughout the course of its implementation, in order to better assess the final product and its accomplishments. Regarding the calibration of the circuit, details will be given in subsection 4.1.1. In terms of the circuitry used in current and voltage sampling, further information will be available in subsection 4.1.2 and 4.1.3. Next, subsection 4.1.4 will disclose the schematics and in section 4.1.5 the firmware will be described comprehensively, with particular care to explain the lightweight payload approach in section 4.1.6. Finally, in 4.1.7 some remarks regarding final prototype.

4.1.1 -Circuit Calibration

To ensure correct operation of the circuit and guarantee accuracy and reliability of the measured values, the ATM90E26 must be calibrated at one specific current, designated basic current or Ib.

This IC includes two calibration types: metering calibration and measurement calibration. Metering calibration compensates for the offset introduced by interference that in turn, is caused by surrounding circuits and power supply variations, usually when sampling low currents with very susceptible shunt resistors. Measurement calibration refers to offset compensation for fluctuations around zero that prevent the acquisition of precise values of voltage rms, current rms, mean active power and mean reactive power caused by outside components. Other measurements including frequency, phase angle and power factor have no need to be calibrated, as it is assured by design.

To begin with, all registers in this IC are of 16 bits, and the ones related to metering calibration are between 20h-2Ch. The Calibration Start Command Register (CalStart, 20h) marks the beginning of a new calibration when set to '5678h'. Soon after calibration, CalStart should be set to '8765h' and the chip will check the correctness of 21h-2Bh registers and whether to start normal metering or interrupt it, by raising an error on the register System Status (SysStatus, 01h).

Next, comes the calculation of the PL_Constant, which refers to the threshold for determining whether the calculated energy is larger enough to be accumulated in the designated registers. This value is calculated as defined in the following equation:

$$\mathrm{PL_{Constant}} = \mathrm{int}\left(838860800 \times \frac{\mathrm{G_L} \times \mathrm{V_L} \times \mathrm{V_U}}{\mathrm{MC} \times \mathrm{U_{N`}} \times \mathrm{I_b}}\right)$$

MC: pulse constant of the energy meter, unit is imp/kWh or imp/kvarh;
U_n: reference voltage, unit is V;
I_b: basic current, unit is A;
G_L: L line current circuit gain;
V_L: sampling voltage of the L line circuit at I_b, unit is mV;
V_U: sampling voltage of the voltage circuit at U_n, unit is mV;
838860800: constant.

In this case, the values used are present in Table 8. The obtained value for PL_Constant was 40297 which translates to 9D69 in hexadecimal.

Table 8 - Values used for PL_Constant calculation

Variable	Value used
MC	3200 imp/kWh
U_n	230 V
I_b	0,26 A
G_L	1
V_L	L line shunt = 0.005Ω, V_L=1.3 mV
V_U	Voltage divider coefficient = 880, V_u=272 mV

Furthermore, it is important to configure the Metering Mode (MMode) register which has multiple options for setting up different metering arrangements that need to be adapted for this specific one. Table 9 contains the available changeable settings, the values they can assume, and the values decided for this case, which correspond to the specifications required.

Table 9 - MMode register description

Options	Description	Possible Values
Lgain	L line current gain	1,4,8,16,24
Ngain	N line current gain	1,2,4
LNSel	Metering on L or N line	N line, L line
DisHPF	Configure High Pass Filter after ADC	• Enable both • Enable HPF1, disable HPF0 • Disable HPF1, enable HPF0 • Disable both
Amod	CF1 output for Active Power	• Forward or reverse energy pulse output (default) • Absolute energy pulse output
Rmod	CF2 output for Reactive Power	• Forward (inductive) or reverse (capacitive) energy pulse output (default) • Absolute energy pulse output
Zxcon	Configure zero-crossing mode	• Positive zero-crossing • Negative zero-crossing • All zero-crossing: both positive and negative zero-crossing (default) • No zero-crossing output
Pthresh	Configure the L and N line power difference threshold in anti-tampering mode	12.5%, 6.25%, 3.125% (default), 1.5625%, 1%, 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%, 11%, 12%

To suit our system, both Lgain and Ngain values should be set to 1; on LNSel, the selected metering line was N; on DisHPF, both high pass filters were enables; all the following options for configuring MMode register were left on their default values as they were not used and would not interfere with the correct functioning of the application.

For measurement calibration, both L line and N line current rms gains had to be calibrated, together with voltage rms gain. This should be done at reference voltage and basic current, which in this case are 230V and 0,26A respectively.

Voltage rms gain calibration is done using

$$\mathrm{Ugain} = \mathrm{int}\left(\frac{\mathrm{Ugain_{old}} \times \mathrm{U_n}}{\mathrm{Vol_{mea}}}\right)$$

where Vol_{mea} refers to the value read from the Urms register (after conversion to decimal base and division by 100, in V), reference voltage U_n is the actual voltage in the grid (in V) and $Ugain_{old}$ refers to the previous value of voltage rms gain, read from Ugain register (power-on value is 26400). After a readout value of 209,56V from Urms register, the actual voltage being 232,86V and considering a $Ugain_{old}$ of 54372, Ugain took the new calculated value of 60417, or in hexadecimal, $EC01_h$.

N line current rms gain is calculated using

$$\mathrm{Igain} = \mathrm{int}\left(\frac{\mathrm{Igain_{old}} \times \mathrm{I_b}}{\mathrm{Cur_{mea\_N}}}\right)$$

where Cur_{mea_N} is the readout value of the IrmsTwo register (after conversion to decimal base and division by 1000, in A), base current I_b is the actual current (in A) and $Igain_{old}$ refers to past values of current rms gain, obtained from IgainN register (power-on value is 30000). Subsequent measures of 0,17A obtained from IrmsTwo register, actual current levels of 0,26A and an old IgainN of 28233, the calculation resulted in a value of 43179 for the new IgainN, which converts to $A8AB_h$ in hexadecimal.

L line current rms gain is yet to be calculated, but to do so, the same equation will be used, only the power on value for the old IgainL is 31251. After a measurement is taken, the process will be the same as for N line current gain calibration.

4.1.2 - Voltage Sampling

To take samples of the AC voltage waveform, a very simple method was employed. In the N line, the 230 V power supply is connected to a series of four 220 kΩ resistors, so that the voltage drops to an acceptable level, and it is also filtered before being connected to the sensor IC port for positive voltage input (VP). Meanwhile, the negative input port (VN) is filtered and connected to the circuit ground. The circuit can be visible in Figure 15.

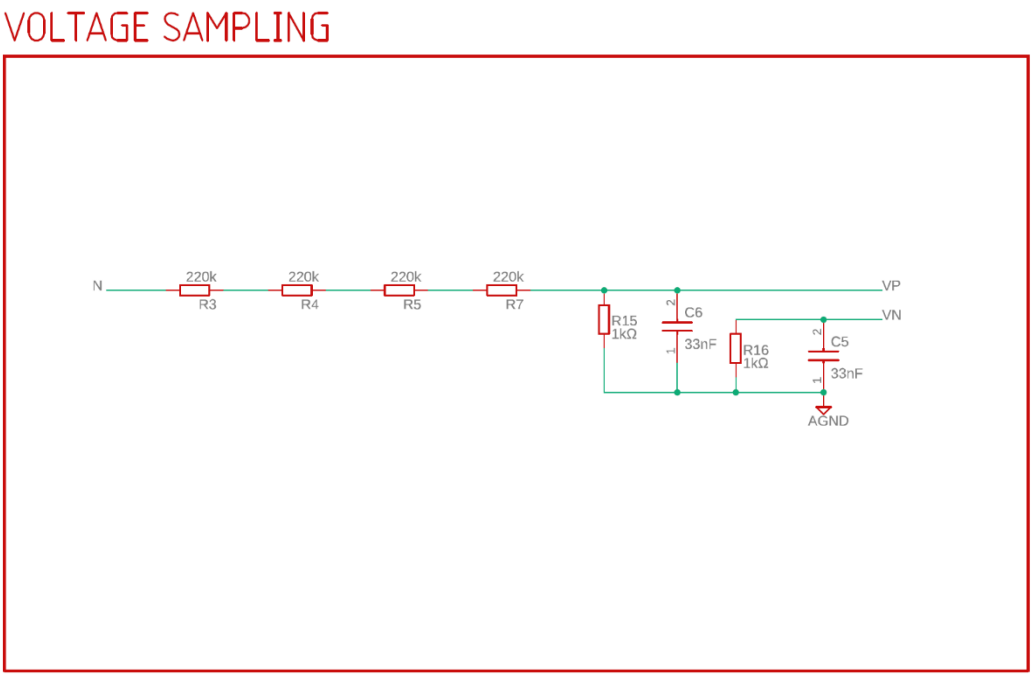

Figure 15 - Voltage Sampling Circuit

The voltage channel allows for values ranging from 120μVrms~600mVrms, when the N line gain is 1. If the gain is increased, the resulting valid range is going to be reduced, e.g., in this application, and considering N line gain can only be programmed to 1, 2 and 4, if gain is increased to 2 then the valid range is going to be halved (60μVrms~300mVrms). So, with gain

set to 1, to verify if maximum range was not achieved, so to not damage the IC, and assuming the varistor between N and L lines would filter out peaks in the power supply voltage, a simple voltage divider check was applied, see equation below. The resulting VP value is approximately 261 mVrms.

$$\mathrm{VP} = \frac{1\ \mathrm{k\Omega}}{1\ \mathrm{k\Omega} + (220\ \mathrm{k\Omega} + 220\ \mathrm{k\Omega} + 220\ \mathrm{k\Omega} + 220\ \mathrm{k\Omega})} \times 230\ \mathrm{VAC}$$

In an attempt to validate the circuit, before advancing with physical tests, a simulation was made using the software TINA-TI [42], a SPICE simulator developed by Texas Instruments. In this simulation, the voltage sampling circuit was reproduced, and, through the application of an oscilloscope, Figure 16 was obtained.

The oscilloscope's display is illustrating the wave being sampled in VP, as VN is connected to ground. The red and blue cursors are placed approximately in the bottom and top peaks of the wave, respectively. Bottom peak (YA) is at 370,81mV and top peak (YB) shows 366,74mV. The corresponding rms values should be close to what was calculated above. By dividing the peak values by $\sqrt{2}$ we confirm that is the case, as the resulting values are approximately 262,2mVrms and 259,3mVrm respectively. This means that the actual voltage in the IC port is of $262{,}2 + 259{,}3 = 521{,}5$mVrms, which, despite being quite close to the limit, still leaves a respectable margin for errors in the grid. For that reason, tests were also made for 240V (a very common value), which concluded, not surprisingly, with small increases in the overall voltage levels to values around 540mVrms. Still in range, but with lesser margin for voltage peaks. Only voltages near 270V would reach and exceed the upper limit of 600mVrms for this particular arrangement of resistors.

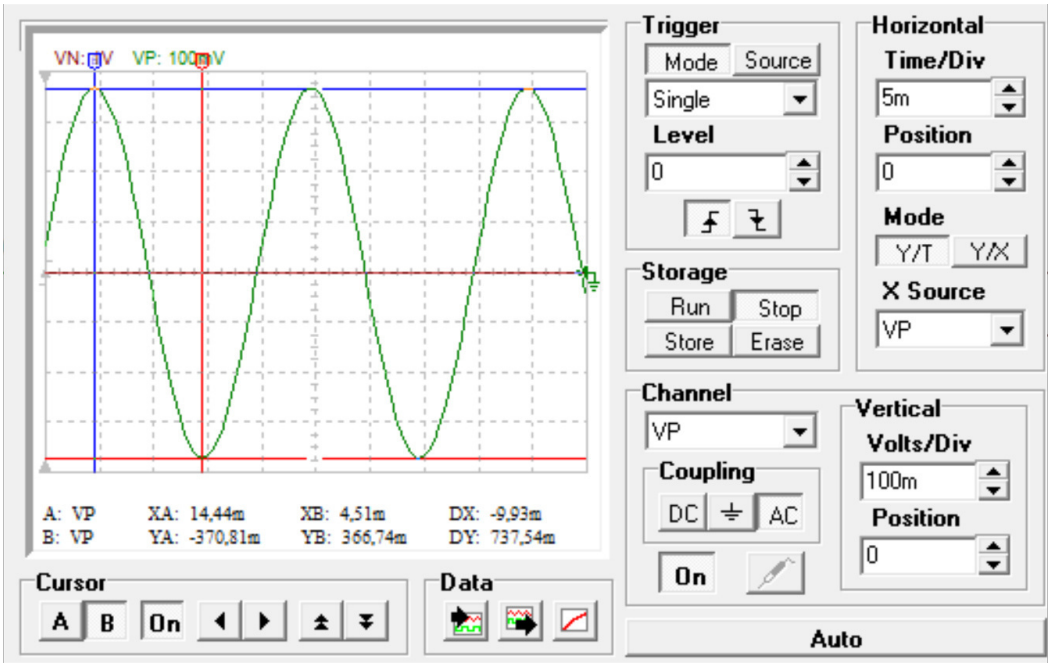

Figure 16 - Visualization of sampled wave in VP in an oscilloscope

To summarize, the absolute resistance value of 880 kΩ was adequately chosen to sample voltage, as it reduced the voltage to an acceptable level to fit the desirable range.

4.1.3 - Current Sampling

i. N Line Current Sampling through Current Transformer (CT)

For this application, current sampling was made in two different methods, to take advantage of the capabilities of the M90E26 sensor. The first one, applied to the N line, was through a commercial current transformer (CT), model SCT-013-030 from YHDC, that admits current values up to 30A and outputs up to 1V. In this case, current values do not draw near 30A. Instead, and considering basic current of 0,26A, the output of the CT that is connected to the IC would range around 8mV. With such small values, it is possible to increase the gain so that we get more precision on the measurement results. However, this number was obtained with a small 60W light bulb as the load for the circuit, which does not compare to the power consumption of industrial machinery this meter was designed to work with. Likewise, if N line gain was increased, changes would have to be made to the voltage sampling circuit.

In Figure 17, it is presented the schematics of the current sampling circuit for this case, in addition to a representation of the actual external current transformer that will connect to the circuit via a 3.5mm jack. It is visible that the CT connects to external filter circuitry, that filters out frequencies above $1/(2\pi \times 1\mathrm{k} \times 33\mathrm{n}) = 4{,}82\ \mathrm{kHz}$ (1/2πCR) and further goes into the sensor IC differential ports - I2P and I2N.

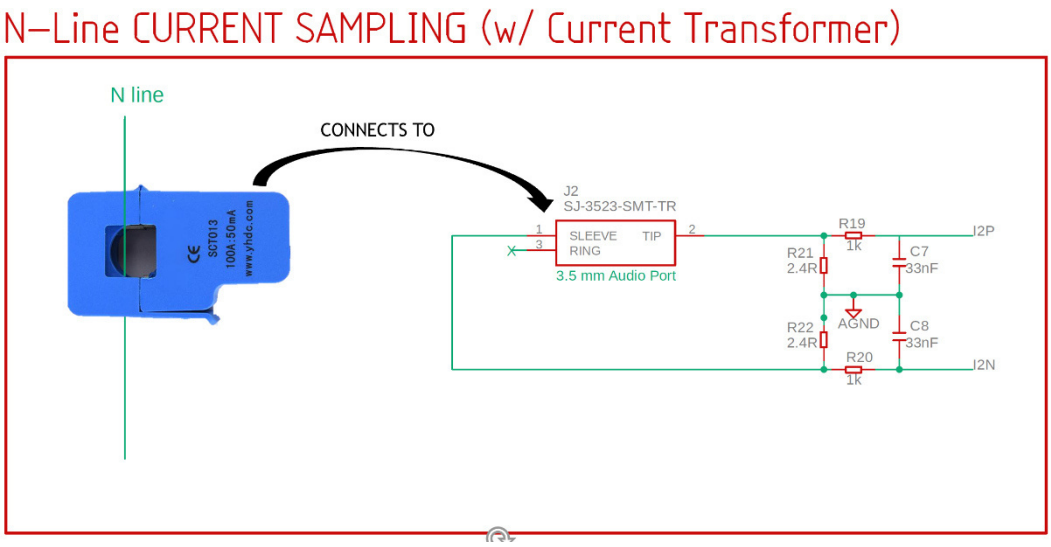

Figure 17 - N-line current sampling circuit with CT

Since the upper limit for voltage input is 600mVrms, we assume that 30A, that results in 1V of output from the CT, will be exceeding said limit. As a result, it is necessary to calculate which value of current would yield 600mVrms on the output of the transformer. To do that, we use the following equation, that takes into consideration the electric parameters found in the current transformer's datasheet

$$I_{meas} = V_{out} \cdot \left(\frac{T_r}{R_b}\right)$$

where I_{meas} is the current that is being measured by the CT, V_{out} is the 600mVrms upper limit input of the IC port, R_b is the burden resistor inside the transformer that forces a voltage output

and T_r is the number of turns in the secondary winding. In the CT's datasheet R_b is 62 Ω and T_r is 1800. So, replacing those values in the formula will return a value of 17,41A.

To assess the validity of this sampling circuit, a simulation was created, again resorting to the TI simulator, TINA. This time, a simulation of the current transformer was done, displayed in Figure 18, with the objective of validating the calculations made and to later use to validate the whole sampling circuit.

The current generator (Measured Current) produces a piecewise linear function, that first increases to value of 30A for 100ms, then plateaus at that value for the same amount of time and then decreases back to 0A. The yellow box (CS1) mathematically represents the CT, by applying the same formula aforementioned.

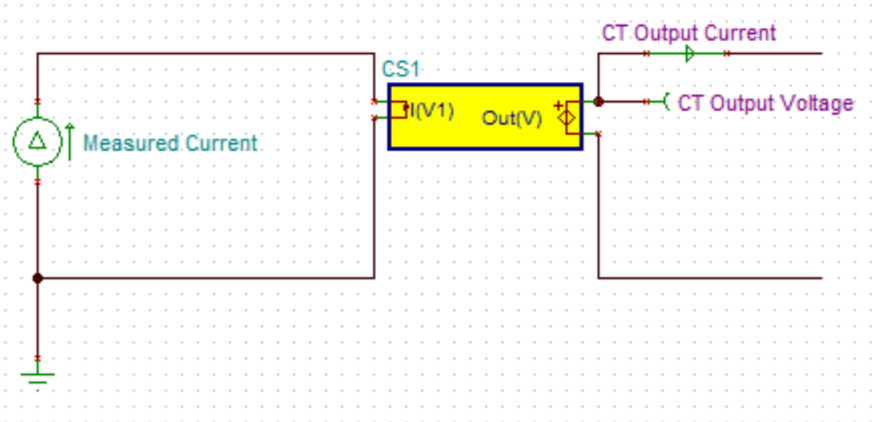

Figure 18 - Simulated circuit of the Current Transformer

The outputs of both current and voltage are present in the graph in Figure 19, resulting from a transient simulation was performed for 300ms. In the "CT Output Voltage" graph, cursor A (in red) was placed precisely at 600mVrms. Then, in the "Measured Current" graph, cursor B (in blue) was placed on top of cursor A to mark the same point and check the reading of current. The obtained value was 17,49A which is very close to calculated one and it is probably due to limitations in the number of samples in the graph that it is not the exact same.

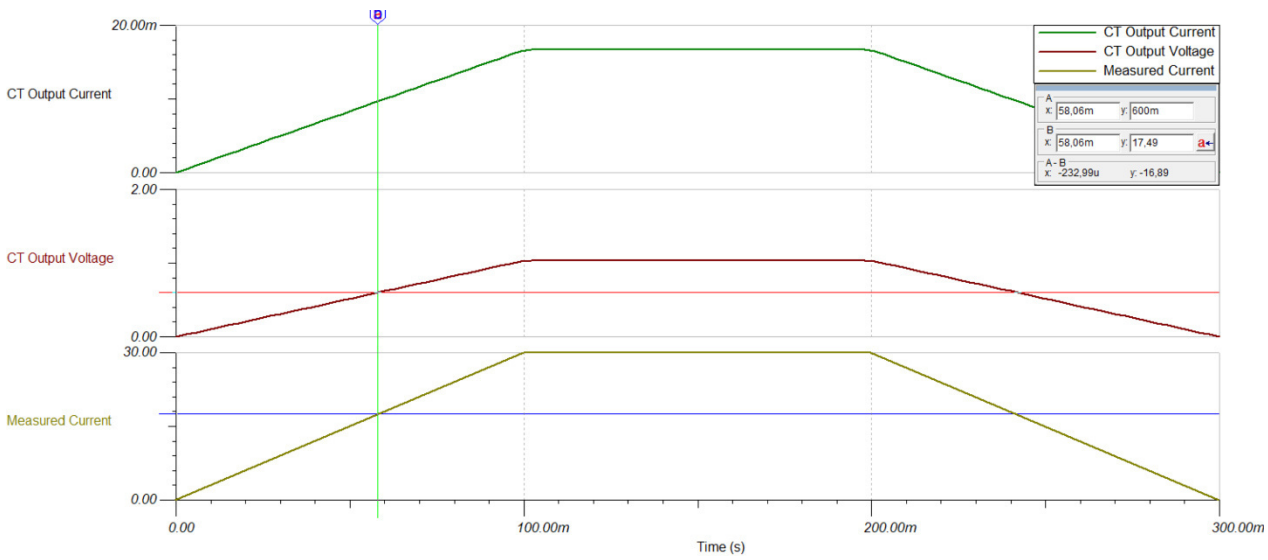

Figure 19 - Simulation results of CT behaviour with piecewise linear function

Then, as was done for voltage, the whole circuit was recreated in the software. Now, the measured current would be limited to calculated and verified maximum of around 17,4A. The behaviour of the current generator (Measured Current) was changed from a piecewise linear function to a Sine wave that is more according to the physical circuit we are testing. At this point, another transient simulation of 300ms was conducted, and the result are shown in Figure 20. Both I2N and I2P are the same wave, only completely out of phase. The cursor A was placed to verify the top peak of the I2N wave and cursor B was used to check the bottom peak of the I2P wave. Both results were nearly identical to 300mV, the necessary to reach 600mVrms.

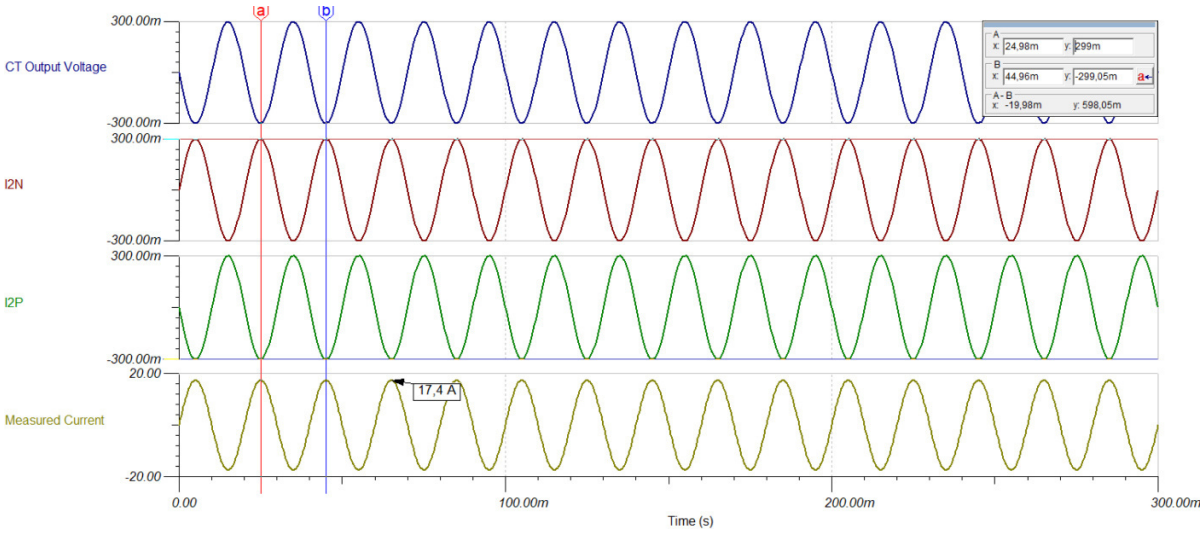

Figure 20 - IC ports (I1N and I1P) inputs while measuring maximum current possible of ~17,4A

ii. L Line Current Sampling through Shunt Resistor

The other method employed in this project was through a shunt resistor that would be placed in the L line, as seen in Figure 21. The fundamental part of this circuit was to choose the correct shunt to place, that had to support large amounts of heat caused by the passage of current through them, without permanently damaging the shunt. Additionally, a low-pass filter would be applied to filter out signals with frequencies above 1/2π*100*330n = 4,82 kHz.

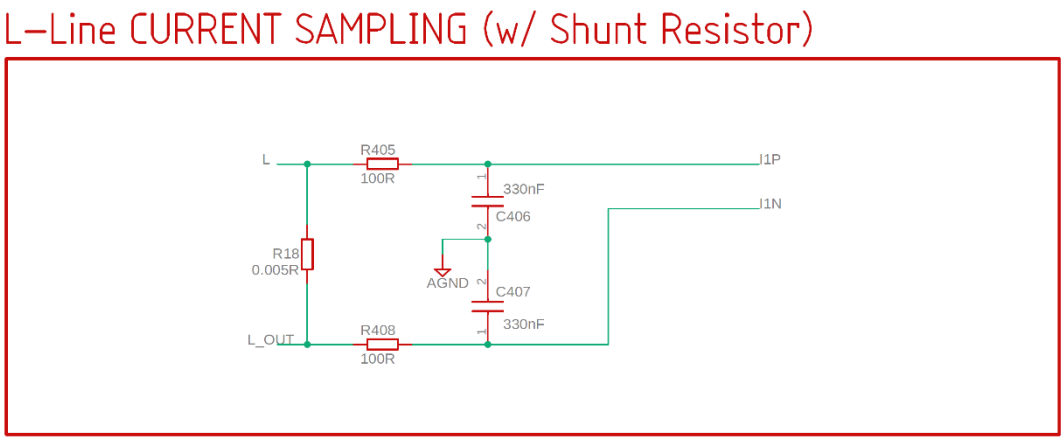

Figure 21 - L-line current sampling circuit with shunt resistor

To calculate the shunt resistor, two premises had to be applied. The first one is that the input range of 600mVrms, of the IC ports, had to be complied with. The other one had to do with the power dissipation of the shunt resistor that could not be greater than what was specified in the datasheet. Furthermore, the load that is attached to the circuit is very important in order to calculate the current going through the shunt, if the current was too small the resolution of the ADC's in the microchip's ports would not have enough resolution to perceive a measurable signal. It is worth remembering that a 12-bit ADC allows for 4096 levels, which considering the 600mVrms, would result in a resolution of 146μV. A 60W bulb was used, with the knowledge that it would produce very little current, but still enough to be measured. Employing $W = V \times I$, and replacing the power (W) with 60 and voltage (V) with 230, the value for current (I) was determined to be 0.26A.

Now, assuming a shunt resistor value of 5mΩ, it is possible to work backwards and verify if the selected value is compliant with the premises established above. If so, then it is an acceptable value of resistance. In this case, using $W = I^2 \cdot R$, we get a value for power of 338μW which is a much lesser amount than the 3W specified in the datasheet for permanent power without a heat sink. For the voltage, using Ohm's law, we get 1,3mV, which is also a readable value inside the parameters established.

Next, it makes sense to check for a larger and more common value of power in the load. This time, assuming 500W of power and, again, using $W = V \times I$, current values reach approximately 2,17A. Calculating the power in the shunt, using $W = I^2 \cdot R$, yields a value of around 23,5mW. Finally, the voltage going into the sensor is close to 10.8mV. So, with an already acceptable amount of power in the load, there is still a long way to the limits of both the shunt power rating and the voltage input range limit of the ATM90E26.

To determine those, considering the limit value for power rating of the shunt resistor of 3W, resultant values for current would be 24,49A. So that would be our limiting factor for current, which translates to a power in the load of around 5.6kW and a voltage of 0,12V in the IC pins. With regards to the voltage input range limiting factor, the outcome of Ohm's law for current in the shunt will be $I = 0.6V/0.005\Omega = 120A$, which is a massive value for current.

Nonetheless, a simulation was carried out to validate these assumptions and the circuits designed. After replicating the circuit and changing between the two loads, a transient simulation was performed, and the results can be seen in Figure 22.

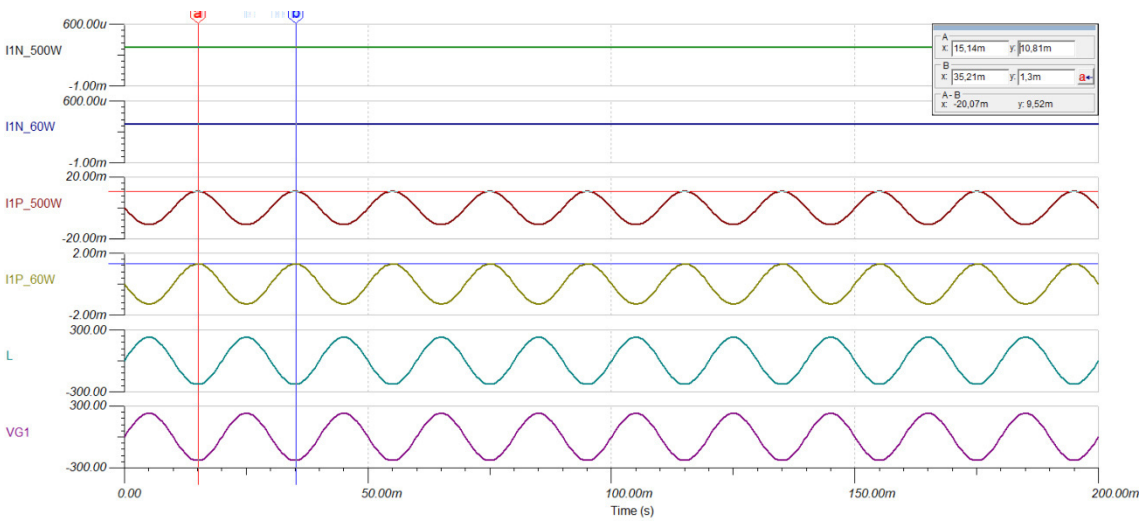

Figure 22 - Simulation results for transient analysis for 60W and 500W loads

As expected, the values from the simulation correspond to what was calculated previously, meaning that the calculations were well executed, and the circuit should work as intended when put in practice. It is very important to mention, that, for the shunt current sampling circuit to operate correctly, it is necessary for the L line to be connected to the circuit ground or for the same ground to be left floating. If the reference of the circuit is connected to the N line (usually connected to Earth) when sampling is being conducted than extremely high voltages (230V) will be present in the ports of the sensor causing catastrophic and irreparable damage to the circuit. This behaviour is visible in Figure 23.

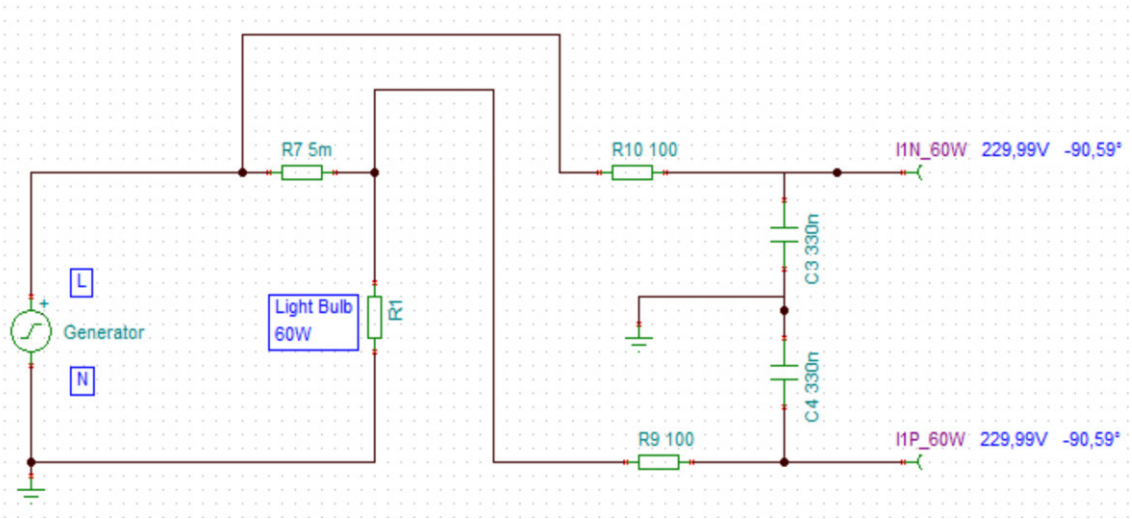

Figure 23 - Output of the shunt current sampling circuit when referenced to N line

iii. Concluding Remarks

As a concluding remark, both sampling circuits are isolated from one another and feature different calibrations for each individual line, as previously mentioned. As a result, interferences the shunt resistor may suffer from surrounding circuitry may not correlate to effects in the current transformer's accuracy, which means acquired values of current through both circuits may vary slightly from one another. Calibration adjustments should aim to diminish these undesired variations.

4.1.4 -PCB Design and Schematics

After testing and validating all the sampling circuits that would be employed, one of the goals set for this project was to design a PCB layout. Ideally, it would fit in three modules of DIN-Rail meter's standard dimensions, around 54mm x 90,5mm, and the aim was to encompass the sampling circuits, together with the supply circuits for the ATM90E26 sensor and the FiPy and the remaining additional circuits into one single board. This section will explain the reasoning behind some choices as well as the function of the circuits themselves, primarily by analysing the schematics not already discussed in the previous section. Also, an overview of the first version of a PCB containing the whole circuit will be given, as well as a clarification of the design.

i. Power Supply Circuits

The first thing we set out to do was to build a supply circuit. Since the meter was planned to be connected to an electrical grid at all times, then, with correct adaptation, it would be self-sufficient and would not require additional energy supplies (generators, batteries, etc.). There were two things to take into consideration before designing the circuit. The first and most important is that two totally separate and isolated grounds would necessarily need to exist. One, referred to as AGND would, essentially, be connected to the entirety of the circuit (sensor, sampling circuits) and the other, referred to as DGND, would be almost exclusive for the FiPy (also control LEDs). AGND should then be connected to the L line, while DGND should be left floating. The aim was to avoid interferences from the supply grid and to protect the MCU from power surges that might occur.

The second consideration was to add protective measures against power spikes and noise. For that, three methods were employed: a ferrite bead, a varistor and a polyfuse. The ferrite bead would block electromagnetic interference (EMI), produced by the high frequency noise of the radio frequency (RF) communications in use, by acting as a low-pass filter. The varistor employed is a Metal Oxide Varistor (MOV) that was placed in between the L and N lines. Its purpose is to protect the circuit against power surges in the supply grid by dissipating the excess energy as heat, as a result voltage is bonded to safe levels. Finally, as an added security for the transformer, a polyfuse was placed in N line input of the AC/DC transformer to safeguard against current surges and overheating.

The supply circuits concerned are illustrated in the schematics shown in Figure 24 and Figure 25 and 26. The idea behind this set up is to use the output of the AC/DC converter to supply the FiPy microcontroller and, through a DC/DC isolated converter, regulate the same 5V output to supply 3.3V to the ATM90E26 sensor chip.

The AC/DC converter in use is a RECOM Power RAC05-05SK. The reasons that led to this choice were primarily based on the required prerequisites for the component. The leading factor was for the output to be between 3.5V and 5.5V, as those were the tolerated values for

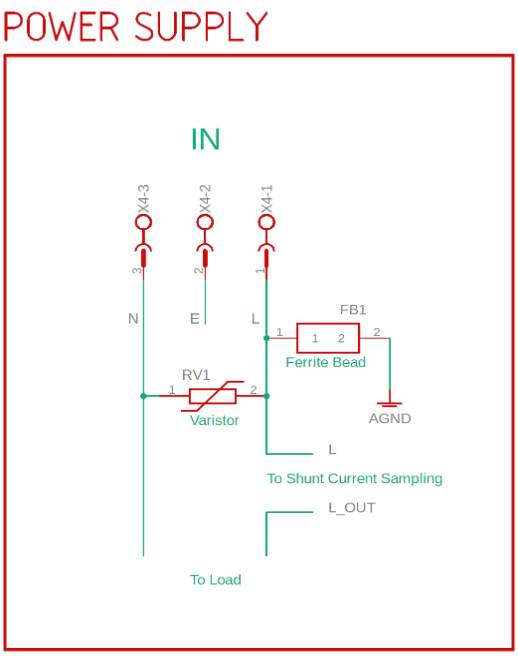

Figure 24 - Power Supply

input of the FiPy. Moreover, the converter's supplied current had to be enough to power on the devices. According to the datasheet, FiPy's current consumptions near values of 200mA, so, to be cautious, we selected a value of 1A for output current which would be more than enough to power the device, the DC/DC converter and the rest of the circuits downstream. Then, we payed close attention to the size of the device, as we were quite restricted in our initial plans to install the whole circuit in a single DIN-Rail module. We found out that the differences between different converters were insignificant, as ones were more rectangular and others were squarer shaped. Finally, we ambitioned for a low-cost device, so the price was also a concern.

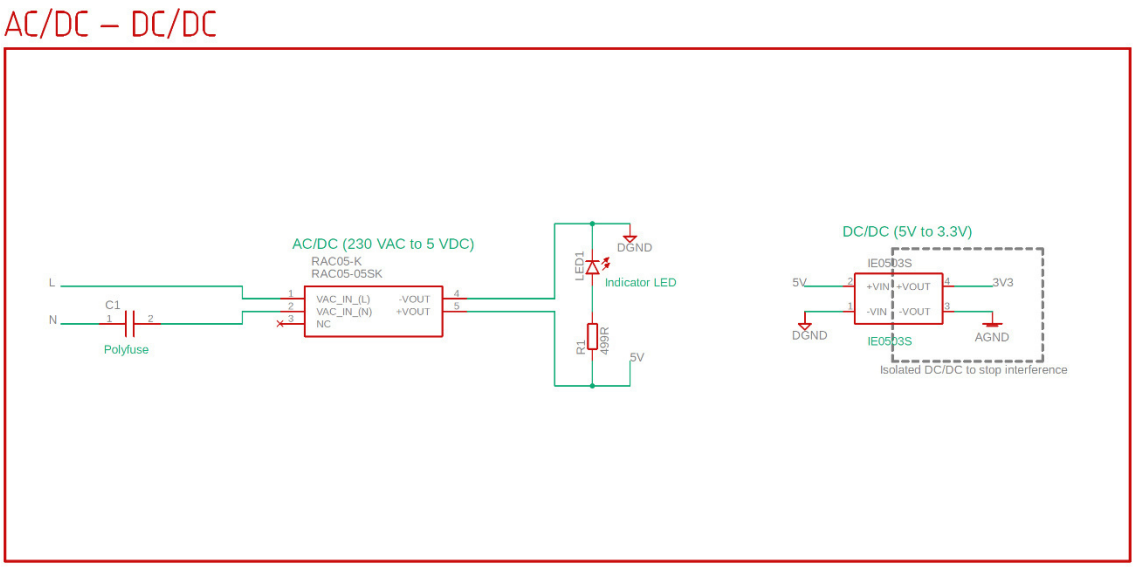

Figure 25 - AC/DC and DC/DC converters

For the DC/DC converter, in was fundamental for it to be isolated, since it was going to be connected to the power supply of the FiPy but could not be connected to the same ground. For voltage, this time we had to make sure the input was 5V and the output was 3.3V, the required for powering the ATM90E26. As for current, this converter outputs 300mA and, as stated in the datasheet for correct operation, values near 3.4mA are sufficient. So, we opted for the IE0503S a very small component, which isolates up to 1kV, and fulfils all the remaining requirements.

ii. FiPy and ATM90E26 Circuits

With the supply circuits in place for both main circuits, comes the time to describe how the ATM90E26 and the FiPy are connected to each other and what other circuitry is adopted to properly adjust the sensor to meet the desired operation mode. The schematics for the circuits in question, are illustrated in Figure 26.

The FiPy is to be connected to female headers, so that it is easily removed for firmware updates, which require a specific board with Micro-USB support. In the schematic, the FiPy symbol is only placed to clear up on the connections that will be made and the correct orientation of the component. The numbers of the pins that will be referred for FiPy's connections are on the I/O lines (from 1 to 28). Ultimately, only the female headers placed in each side will have actual connections represented. For power, Pin 28 is connected to the 5V output of the AC/DC and, through Pin 27, the board GND will be connected to DGND. Other

connections include the SPI connections for a 4-line half-duplex interface with the sensor, through Pins 8, 10, 11, and 14 that represent, respectively, CS, SCLK, SDI and SDO; and an RGB LED that connects to Pins 23, 24 and 25 that each represent one of the colours.

The connections visible in the bottom of the symbol, which, by default, are not present, refer to UFL connectors that are physically present in the top of the chip for the WiFi and the LoRaWAN external antennas.

Regarding the ATM90E26 sensor, there are several things to take note of. First, with concern to the MMD0 and MMD1 pins, which will configure which metering mode will be selected. In this case, for an L and N lines single-phase three-wire system, MMD1 will be high and MMD0 will be grounded. Subsequent connections must be made for both Analog Power Supply (AVDD), Digital Power Supply (DVDD) and their respective grounds (AGND_1 and DGND). These will power the analog and digital parts of the sensor. AVDD and DVDD are connected to 3.3V and should be decoupled. DVDD with a 10µF electrolytic capacitor and a 100nF capacitor, and AVDD with an 100nF capacitor. The RESET on Pin 4 is a power-on feature, and the pin should be connected to 3.3V so that the metering parameters are not reset every time the circuit switches off. The USEL pin, connected to ground, is a selection pin for the interface (SPI/UART) that is going to be used, which is SPI for this application. The VREF is a pin that outputs very precise and stable reference voltage, which, according to the datasheet, should be connected to two capacitors of 1nF and 1µF. On the right side, there is all the SPI ports that were referenced above, two of which double as UART pins, if so required. Then there are two pins that will connect to the crystal oscillator (OSCO and OSCI), which will provide clock signals of 8.192 MHz to synchronize internal operations.

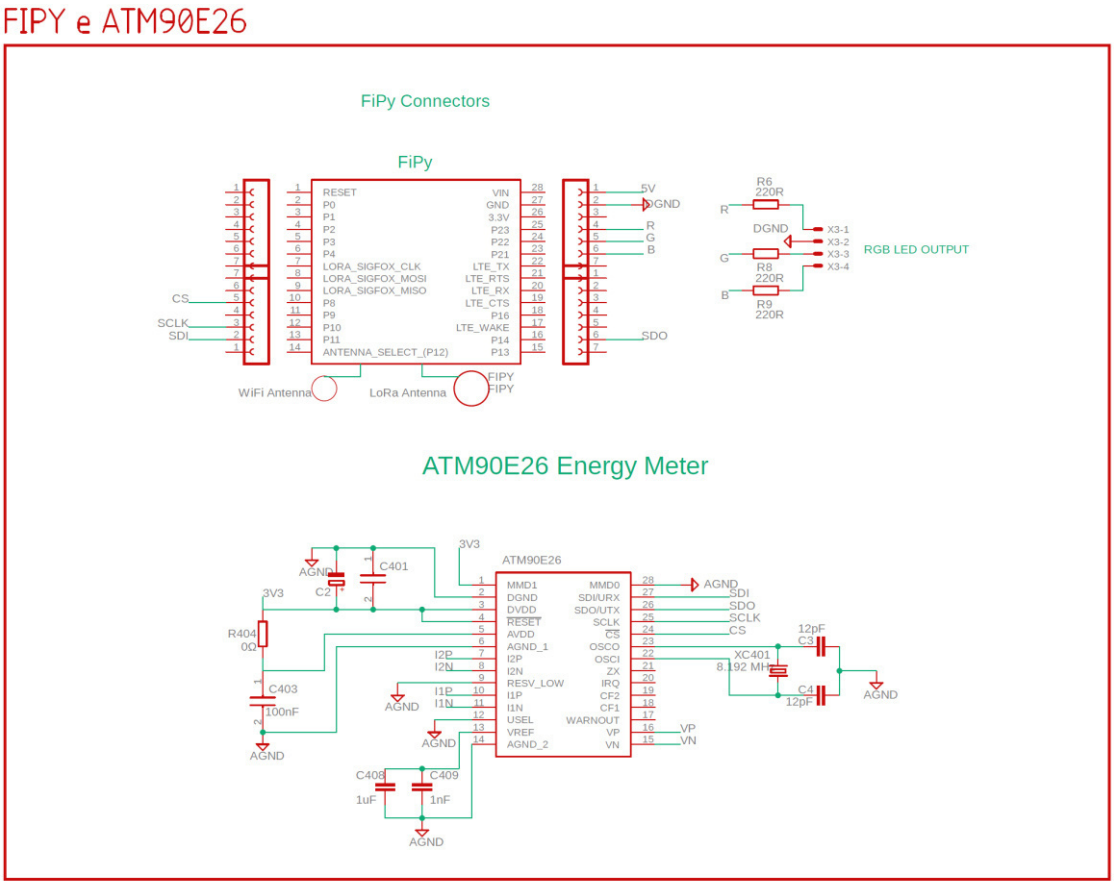

Figure 26 - FiPy and ATM90E26 circuits

The following pins are not in use, but still a brief description will be given to elucidate on their purpose. The ZX pin is a programmable voltage zero-crossing output, which can be configured for positive, negative and all zero-crossing. The IRQ pin is an interrupt request output, which outputs high in several occasions: metering/measurement parameter errors, active/reactive energy direction changes, voltage sag and metering line change when in anti-tampering mode. Both CF1 and CF2 are energy output pins, CF1 outputs active energy pulses and CF2 outputs reactive energy pulses. These pins can either be used for calibration, or when connected to the FiPy be used for energy accumulation. The remaining pins are WARNOUT, to alert to parameter errors, and RESV_LOW, which is a reserved pin that should be connected to ground.

iii. PCB Layout

The PCB layout that ended up being conceived had a couple of differences from the original idea. As demonstrated in Figure 27, a larger design, of double the size, took precedence over a more compact one, at least in first generations of the design. That is because some components may change in the future, or even some may be added, depending on the application. As an example, optocouplers or other form of isolation may be implemented to avoid interference in the SPI interface, or even support for Micro-USB connection to allow for in-board firmware updates could be added. Also, another thought to back-up a more sizeable implementation, involves the opportunity the keep the current transformer hidden inside the device, which was not possible otherwise.

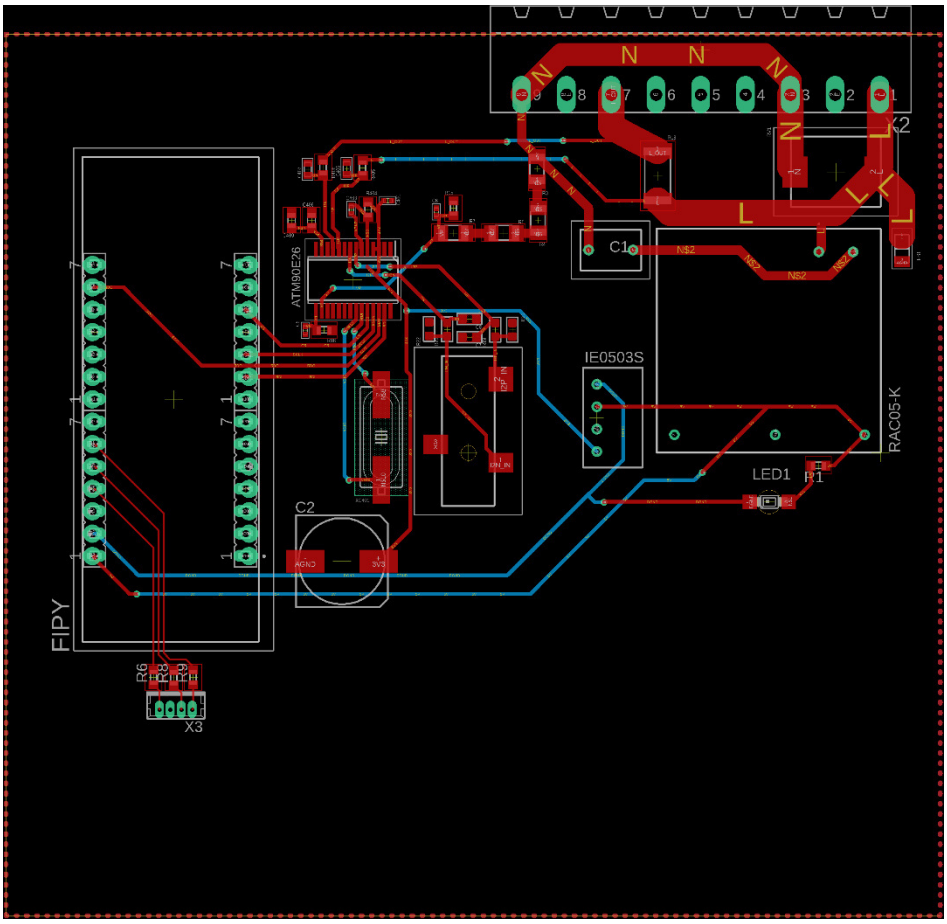

Figure 27 - PCB Layout

The layout itself was divided into three distinct sections: the left side was reserved for the MCU (FiPy); in the middle, all the sampling circuits and sensing components, including the ATM90E26 sensor itself; and in the right side, the higher AC voltage connections and the converters that supply the rest of the circuit. There was a conscious effort to preserve this division and to keep more sensitive equipment the farthest away from the potentially damaging high voltage.

Some caution was taken when selecting the width of the tracks for the high voltage connections. To allow for loads of close to 1kW, calculations suggest a value for current of around 4.3A. Through an online calculator [43], and considering a PCB copper layer of 0.07mm, the value for the track width returned was 2.92 mm. We complied with this value in our design and opted for a width of 3mm for the tracks which connected the AC supply to the load. If so required, this value could potentially be reduced by either using both top and bottom copper layers (with vias connecting them) to distribute the current, or allowing for greater temperature increases in the tracks, or even increasing the copper layer's thickness.

Other cares involve the size of specific components, which will require capacity for larger amounts of current, i.e., a higher power rating. Those components were evidently in the sampling circuits which are in contact with the supply grid, meaning the shunt resistor and the series of resistors for voltage sampling, as shown in Figure 28. The shunt resistor in particular has a package size of 2512 (6.3 x 3.1 mm), since all the current going to the load has to pass through it. The resistors for voltage sampling, on the other hand, do not need such large packages, because the current going through them is $230\mathrm{V} \times 880\mathrm{k}\Omega = 0.26\mathrm{mA}$, as such, their package size is only 1206 (3.2 x 1.6 mm). However, current surges may occur sporadically, and we need to account for that.

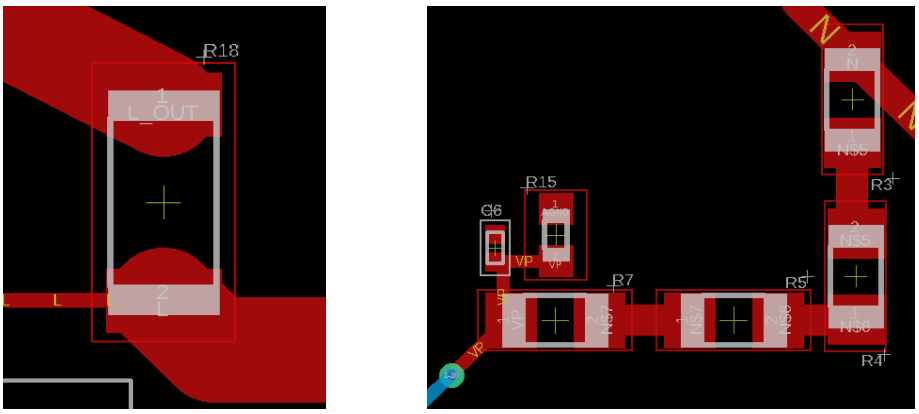

Figure 28 - Layout of shunt resistor and series of resistors from current and voltage sampling circuits employed.

Overall, the design is straightforward, and the layout is of accessible understanding and does not feel congested. There is a good deal of space to place new components, if improvements are required. If not, and if there are no rigorous size restrictions then, the module is at an acceptable size as is. There is also an abundance of safety measures that will protect the circuit and its load from harmful situations.

4.1.5 -Firmware

The developed firmware was created with the intention of implementing a set of libraries and functions that could fulfil the requisites of this project, in terms of both data processing and possible actuation methods; and communication that can either be between the MCU and ATM90E26 Sensor or between the MCU and the gateway.

For the formulation of the source code, the selected tool was Atom, a very complete and versatile source code editor that, as previously mentioned, is favoured by having access to the Pymakr plug-in that facilitates interfacing with the microcontroller. Before discussing the implementation, it is important to brief on the file structure of a MicroPython program and how it is organized in this case.

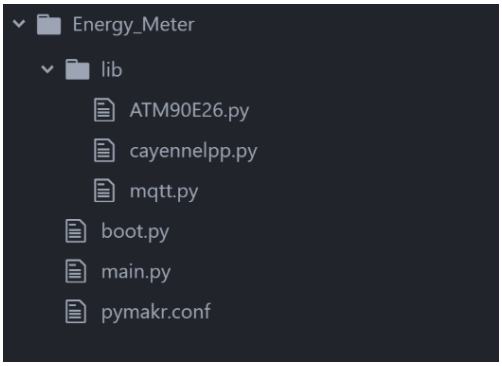

Figure 29 - Library management in FiPy internal memory

Demonstrated on Figure 29 is the structure that the energy meter's firmware follows, as a MicroPython project. The first script that runs when the device is switched on is the `boot.py`. This file usually runs the set-up methods for communication protocols, and enables FTP connection, avoiding unnecessary code in the `main.py`. This script immediately follows `boot.py` and, as the name indicates, it encompasses the main code that will run on the device. Finally, inside the lib folder should be included any extra libraries used in the firmware. It is recommended that only .py files are included, since, by default, MicroPython will not detect libraries inside sub-directories.

To develop the firmware, it was necessary the use of three additional libraries, apart from what is already included in the `main.py`. This division not only made the entire project much clearer and readable for external users, but also simplified immensely the development process.

The first library, named `ATM90E26.py` had the main purpose of controlling the ATM90E26 sensor and managing SPI communications. The script includes a function that establishes communication for reading from and writing to specific registers of the sensor; one for initiating the sensor itself, by setting the values for both metering and measurement calibration and some conditions to validate those values; and then several functions that take the readout value from different registers that hold relevant information for metering purposes (current rms, voltage rms, power, power factor, frequency, phase angle and active and reactive energy) and make the necessary adjustments (converting to decimal base and adjusting the decimal point location) before returning.

The second library contains multiple functions that facilitate immensely the process of constructing a CayenneLPP payload. They avoid the cumbersome process of having to continuously add the “data channel” and the “data type” to every additional sensor reading included. Furthermore, the library offers a sending function for LoRa, that resets the payload for next transmissions.

Finally, the remaining library is mainly used to coordinate MQTT activity. It provides a way to create an MQTT client and to set up its different attributes. Besides, it contributes with a multitude of functions that handle every aspect of MQTT protocol communication. Those include message publishing and topic subscribing methods; connecting and disconnecting methods; a pinging method; one to wait for the arrival of a message from the server and methods for setting the callback function and the last will before disconnecting.

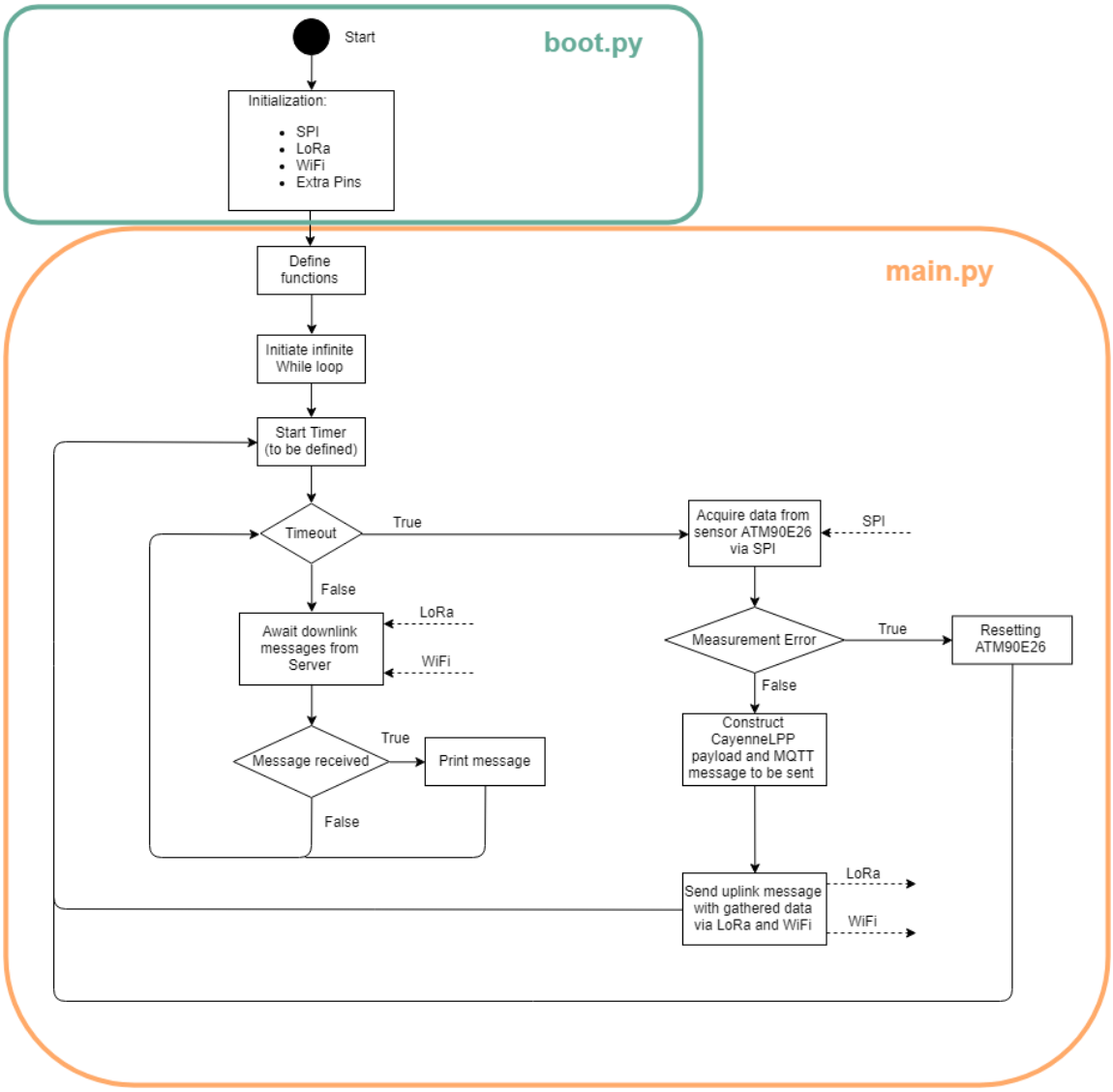

Figure 30 - Flowchart of Firmware operation

Figure 30 was included to better understand the sequence of events the program goes through and the communications entailed through the course of the script’s execution. It will guide the succeeding explanation and will be referenced throughout.

As previously stated, the boot.py is the first script to run, and this is no exception. We can observe, in the green box of the figure, that the `boot.py` is used to initialize different communication modules and to declare extra Pin configuration.

One of the modules is the LoRaWAN, whose stack is initialized and before the joining procedure with the server, it is authenticated via Activation by Personalization (ABP). This method assumes the encryption keys are hardcoded in the device, and the connection is automatically established without the need for a "handshake" between the end-device and the server. Then, the LoRa socket needs to be created. Its initialization includes parameters such as, data rate definition (between 0 – 5, which reflects to 0.3 - 5 kbps), and confirmed or unconfirmed messages (through acknowledgments).

The WiFi module is initialized in 'Station' mode to be able to connect to the desired network. The network SSID and password are also hardcoded and need to be changed manually if any of the parameters change. This is also where the MQTT client is created, the callback function is set and the subscription to a specific predefined topic is made. For now, the only purpose of the callback function is to print out the downlink messages to the console.

SPI initialization is a single line of code which sets different parameters to adequately configure the interface. These include whether the node is a slave or a master, the Baudrate utilized, the polarity and phase of the clock, the number of data bits and the pins for each signal.

Finally, the pins that are set, relate to the RGB LED which will inform of various events that are occurring during the operation of the firmware, by displaying different colours.

Then we move on the next script, which is the main program. It starts by defining a few functions that will be required later on. The first is responsible for acquiring the data from the SPI interface, be resorting to methods declared in the `ATM90E26.py` library. The following will construct the CayenneLPP payload to send via the LoRa socket and to include in the MQTT message to be published. This is done by employing functions from the `cayennelpp.py` library. The last is a sending function that sends data via LoRa and publishes the message, via WiFi, on the desired MQTT topic, while also checking for errors in the transmissions.

Eventually, the program enters the infinite cycle of awaiting message arrival, followed by data acquisition and concluding in message transmission, which in turn will renew the cycle. The timer, limiting the window for downlink message arrival, is set for 30s. If in that timeframe a message arrives, then it will be printed to the console. If not, then the timer will runout and the data gathering process begins.

This process consists of getting the readout values of different measurement registers in the sensor, through SPI interface. If something goes wrong in the process, the sensor in reset and the cycle starts again without having sent any new messages.

So, in short, `boot.py` will initiate communication modules and set up correct configurations for each one. Then, the main script will have the infinite loop which will start by waiting 30s (reconfigurable) for a downlink message to arrive. At timeout, the measurement readings will be acquired and sent uplink to the gateway.

4.1.6 -Lightweight Payload Approach

In order to comply with low duty-cycles imposed by LoRaWAN standard, an open-source approach for a low power payload byte-based was selected – Cayenne Low Power Payload (LPP). This approach resides in giving each sensor a uniquely identifiable data channel byte (ID),

followed by a data type byte that points the type of measurement to come in the following bytes, and finally the data bytes with specific limits linked to each one, as per IPSO Alliance Smart Objects Guidelines. For this specific application, the data types used are displayed in Table 10.

Table 10 - Data types and dimensions under use

	IPSO Object Code	LPP Data Size [bytes]
Voltage RMS	3316	2
Current RMS	3317	2
Grid Frequency	3318	2
Active Power	3328	2
Power Factor	3329	2

During a frame transmission, both time of emission and time on-air are recorded, using pre-existing functions of the firmware, with the intent of calculating the time in which the same sub-band is inaccessible, or it would potentially create packets to be lost forever due to blockages with other sensor communications. So, in order to calculate the time off one must employ

$$\mathrm{T}_{\mathrm{off\_sub\_band}} = \frac{\mathrm{T}_{\mathrm{On\_Air}}}{Duty\_Cycle_{sub\_band}} - \mathrm{T}_{\mathrm{On\_Air}}.$$

So, per metric measured, it is added to the payload a total of 4 byte, as specified above. Assuming a complete payload of 20 bytes with measured data plus the commonly fixed 13 bytes with LoRaWAN metadata, one can estimate the time on air for transmitting such a message and thus, obtaining the maximum repetition period.

Table 11 shows calculations for different LoRaWAN configurations and WiFi communication as a baseline for comparison. The calculations are performed resorting to an online tool specific for LoRaWAN air time calculations [44]. From the obtained metrics, it is possible to conclude that such values are very dependent on the payload size and spreading factor adopted, but with a careful selection, practical and quasi real-time repetition message periods are achievable and perfectly in line with more demanding and of lower coverage range WiFi based solutions.

Table 11 - Comparison of message time repetition for the different connectivity protocols

	Time on Air	Repetition Period
WiFi	N/A	every 1 or 2 seconds (if needed)
LoRaWAN @ SF12, 125 kHz, 33byte payload	1811ms	approx. each 3 minutes
LoRaWAN @ SF7, 125 kHz, 33byte payload	72ms	approx. each 7 seconds
LoRaWAN @ SF12, 125 kHz, 13byte payload	1155ms	approx. each 2 minutes
LoRaWAN @ SF7, 125 kHz, 13byte payload	46ms	approx. each 5 seconds

4.1.7 -Final Prototype

The work done in throughout this chapter regarding the different stages of the implementation, has not yet led to a final working prototype. Some delays in the arrival of components led to some setbacks that ended hampering the development.

However, through other means, we assembled several blocks which were connected to each other via wiring and led to the results shown in the following chapter. The FiPy was used while mounted in its Expansion Board (used for firmware updates, also supplies power and Micro-USB connection), a board containing the ATM90E26 was also ordered [45] to facilitate the process of acquiring results as fast as possible. Finally, the sampling circuits were assembled in a breadboard and, at last, the whole circuit was implemented.

Chapter 5

Tests and Validation

Now that the implementations have been described in detail and it is clear on what are the capabilities of this IoT solution, it is time to subject it to some tests. The main purpose is to verify if what is implemented, so far, has met the objectives for an accurate meter with precise measurement acquisition that was capable of communicating with a gateway and, consequently, with the LoRa server for data visualization.

Section 5.1 will present the results of a coverage analysis performed on both communication networks, as well as the conclusions drawn from them. In section 5.2, a comparison between the developed solution and an already available one will be established. An assessment will be conducted based on the graphs containing the resulting measurement values from each one.

5.1 - Coverage Results

In the context of an IoT project, a simple coverage analysis of one illustrative industrial building was executed.

It is quite important to understand initial network radio coverage for both connectivity options, so that optimization or a rearrangement of any in the implemented IoT solution piece could be employed, before settling for a final design.

Such an evaluation of the network coverage is vital for the correct operation of the system and allows not only to validate the device being used but also to give a better perception of the differences between LoRaWAN and Wi-Fi coverages on mixed indoor/outdoor industrial environment.

This basic coverage study was based on obtained radio signal strength indicator (RSSI) metric for both LoRaWAN and Wi-Fi networks, being created by the multi-protocol gateway. It was performed with a special firmware running on the implemented FiPy's node, which records those values into the memory for offline processing. The data composed of position and respective RSSI values, is being sent also directly to the central server for further corroboration.

Looking at Figure 15, it is possible to check the different coverage conditions for LoRaWAN and Wi-Fi in various positions of the industrial building. Making a rapid assessment with a simple division in green, yellow and red colors, it is crystal clear the larger coverage range provided by the LoRaWAN network, as expected. To exemplify, there are some of the measured points where the communication via Wi-Fi is not possible at all, since the COIOTE gateway AP is not

being detected at all. Such situation was already expected due to external concrete walls of the industrial building. It is worth mentioning that LoRaWAN network was run on data rate zero (DR0), i.e. configuring the spreading factor to SF12 and a bandwidth of 125 kHz, which translates into an indicative bit rate of just 250 bit/s (lowest data rate encoding available [10]).

Finally, it is showed that there is a greater reliability in LoRaWAN because it uses a lower ISM band that is far less subjected to interferences from other networks, becoming more common within factory environments like crowded Wi-Fi industrial areas.

Figure 31 - Coverage analysis performed at several measurement points for a common industrial ground floor, composed of open spaces, electrical/electronic machinery and walls separating the different divisions.

5.2 - Energy Meter

So far, the state of development of this solution is at a point where every aspect of the circuits implemented has been simulated and some have been physically tested as well. The voltage sampling and current sampling (w/ CT) have been tested and are operating correctly, which will be shown next. However, the shunt resistor revealed to be more challenging and a failure to correctly set it up proved to be disastrous and ended up damaging both FiPy and the ATM90E26 sensor beyond repair. The error has been identified and, in this report, it is identified the proper procedure for correct operation for all the sampling circuits. However, there will be no results for measurements using that circuit. The PCB is close to being validated, but will only be sent to manufacturing, once the circuits have been physically tested and verified.

5.2.1 -Results and Comparison with Existing Product

So, a 60W bulb was used to test out voltage sampling and current sampling with current transformer. For, approximately 30 minutes, the circuit measured voltage rms, current rms, grid frequency and power and the results are expressed in the graphs below which have been taken from Grafana. Each 30s, a reading would be taken from the sensor's registers to be sent via LoRaWAN and WiFi to the gateway. The values obtained were compared to those of a

commercial meter from Hiking, model DDS238, shown in Figure 32. The comparison was not done in the same time frames, only a broad value comparison was done between the two pieces of equipment, to have an estimation if the acquired values were in close vicinity.

Figure 32 - Hiking DDS238 single-phase meter

In Figure 33 and Figure 34, it is visible the similarities in both graphs. The voltage values are very close in proximity and present very close resemblances in the obtained variations.

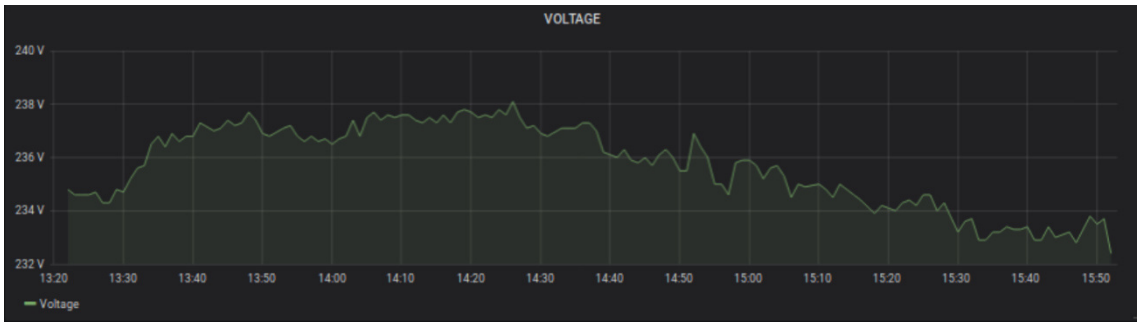

Figure 33 - DDS238 acquired values for Voltage

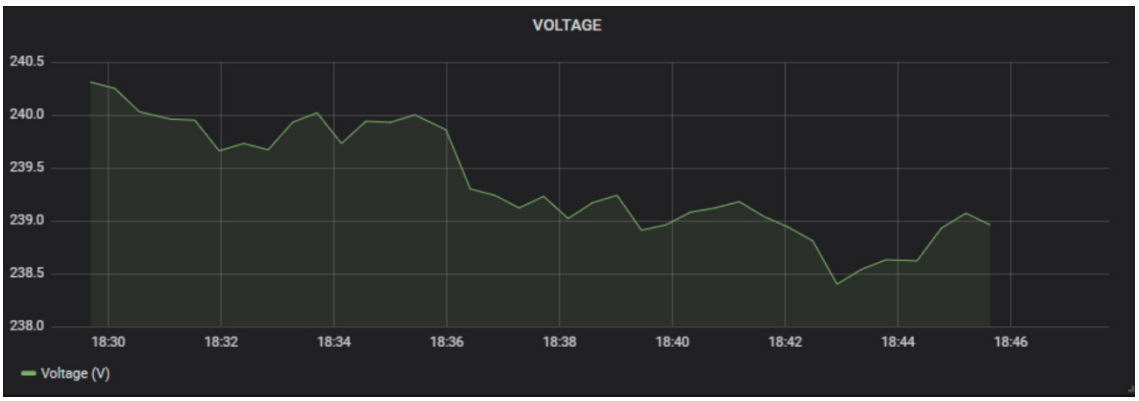

Figure 34 - Energy Meter acquired values for Voltage

In the following couple of graphs, in Figure 35 and Figure 36, current values are both very stable in variations and very precise in their measurements.

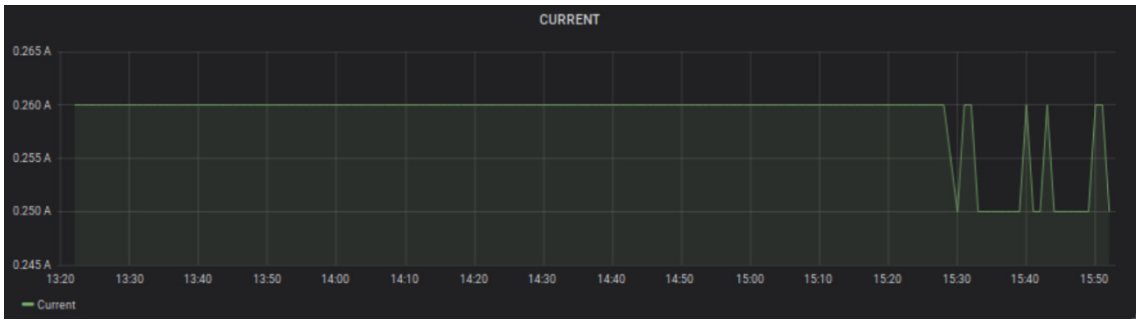

Figure 35 - DDS238 acquired values for Current

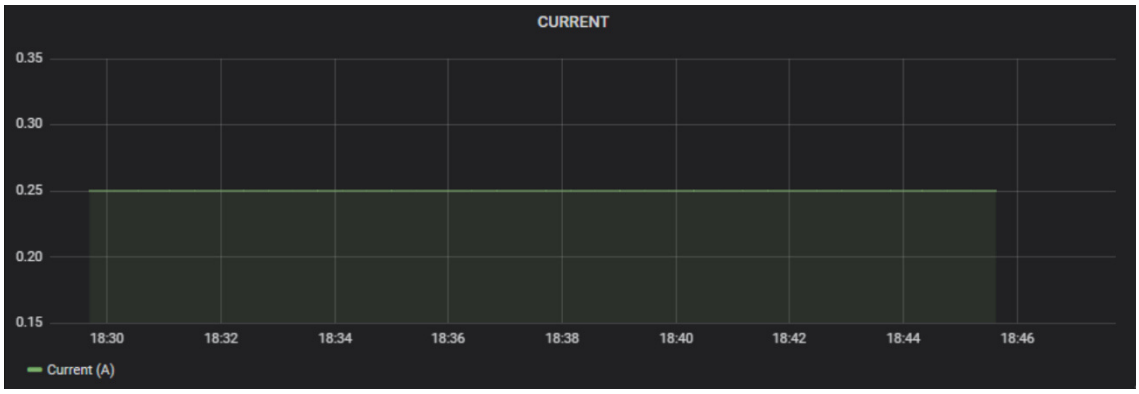

Figure 36 - Energy Meter acquired values for Current

The trend maintains for power, as values, seen in both Figure 37 and Figure 38, are very close to those of the Hiking model. However, throughout these measurements, it's noticeable a lack of resolution in the developed solution. This is a limitation of the projected device due to both CayenneLPP and the sensor's data formats and range. In CayenneLPP the maximum resolution is 0.01 which is not enough for current measurements that have very small fluctuations. For power, the range is from -32.768~+32.767 kW, meaning the resolution is 1W, which for lower values of power limits the accuracy of the measurements. Both these limitations might not be as evident or detrimental when measuring higher values.

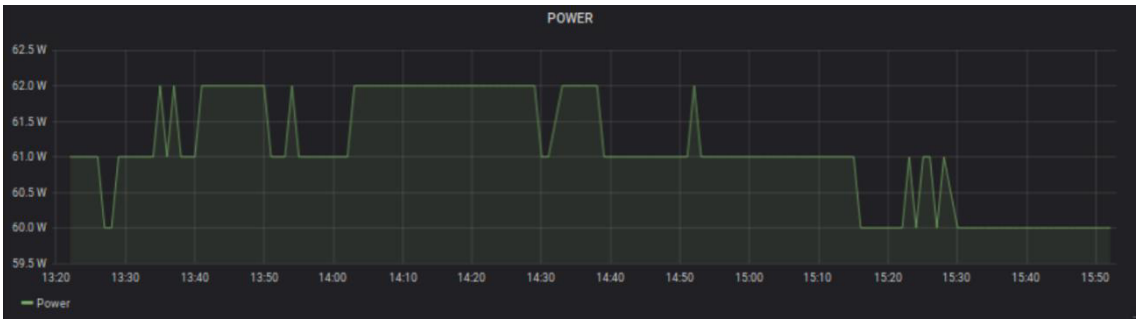

Figure 37 - DDS238 acquired values for Power

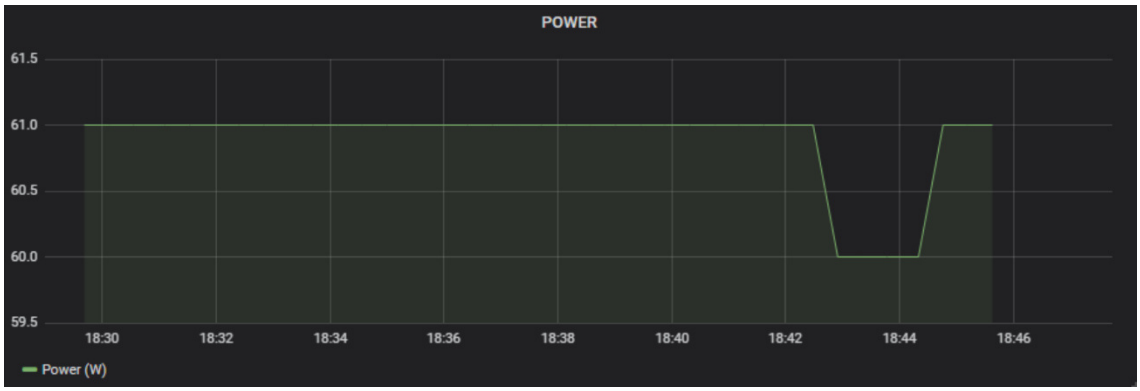

Figure 38 - Energy Meter acquired values for Power

Chapter 6

Conclusion

6.1 - Final Thoughts

What this project set out to do was implement an end-to-end and multi-protocol IoT solution for Industry 4.0 applications that focused on an energy metering use case. In that front, it was a successful project that even allowed for a conference article to be produced which will be featured in the appendix A.2.

It benefited from the convenience of such a complete MCU (FiPy) that added practicality and ease of use to what could have, otherwise, been very cumbersome tasks of module integration and adaptation for the communication protocols. However, the implementation entailed some challenges in understanding the architecture behind the sampling circuits and the care they required to be adequately designed. Also, the need to tinker with multiple registers to correctly calibrate the circuit, proved to be very challenging and added to the effort of developing an accurate device capable of providing reliable measurements.

The main focus on multi-protocol, which allowed for a more adaptable end product and tapped into the potential for LoRaWAN in industrial applications, could open doors to more opportunities in the business, by employing the same base technology to other applications, in other conditions and with slightly different purposes. The connectivity tests, both in terms of coverage and regarding the integration of both protocols with a single gateway and server, proved to be successful and showed potential for a very appealing final product.

6.2 - Future Work

The prototype developed is already sparking interest in potential integrations in with available industrial machines, in the future. Although, it is still in its early stages, the main concepts of energy metering and multi-protocol are to be kept for further advancements in the development of the solution. Right now, possibilities for improvement are many and some ideas have certainly been discussed.

One of them is related to the price of the end product. As we intended on keep the costs low, because of scalability issues and competitiveness in the market, the resolution on substituting the FiPy for cheaper alternatives was almost certain. That would mean that only one communication would be available, because, if not, a cheaper solution would most likely not be possible. That is due to certifications for LoRa transceivers which would have to be

mandatory for this project but increased the cost of most transceivers. That coupled with the rest of the components for WiFi, and the cost would be almost the same as the FiPy. More in-depth research would need to be done on the topic.

The other was more associated to the energy metering part. The proposed solution fixed on single-phase three-wire, when in reality, by doing so, we are neglecting a big portion of potential consumers, as commercial and industrial buildings with higher power consumption use three-phase based power grids. So, the next logical step should be to include the two systems, which in and of itself should be an equal amount of work as the project at hand.

The sensor used for this solution would not be applicable as it only measures in single-phase systems. In that case, extensive research would need to be made to find an appropriate sensor to perform the task and then repeat the process this project underwent.

As a commercial product, we strive to make it as adaptable to any conditions as possible, while trying to keep the price on the lower end without compromising the integrity and reliability of the solution. This should make the final product more appealing to consumers. By taking part in the Industry 4.0 revolution, our efforts resided in the pursuit for innovative solutions that could empower companies with better and more energy-conscious decision-making tools.

Appendix

A.1 - Controller/Monitor of an Industrial Machine

During the course of this project another application was conceived, with demonstrative purposes, that was planned from the get-go to be featured in an existing machine to be seen at a technology exhibition in Exponor – the 360 Tech Industry – that focused on Industry 4.0, robotics and automation. It took advantage of the work already in development for the Energy Meter and adapted it for this case.

The device's purpose was to enable remote monitoring of temperature and relative humidity, along with control options for two outputs (led strip and the machine's fan). It also supplies power for an output (RF Switch) which does not support any control or monitoring. In that sense, the application proved important and valuable, as it served as an example for downlink message processing, and consequent actuation commands these devices offer, to complement management solutions.

A.1.1 -Board Design and Schematics

Part of the work involved in this application was the development of a board that incorporated all the components into a single casing, of small proportions, to be included inside the machine. It consisted of a power supply circuit for the FiPy and the devices in the load; a relay board which allowed for simple on/off switching actuation of the outputs; and the circuits for connecting to the FiPy and to the temperature/humidity sensor.

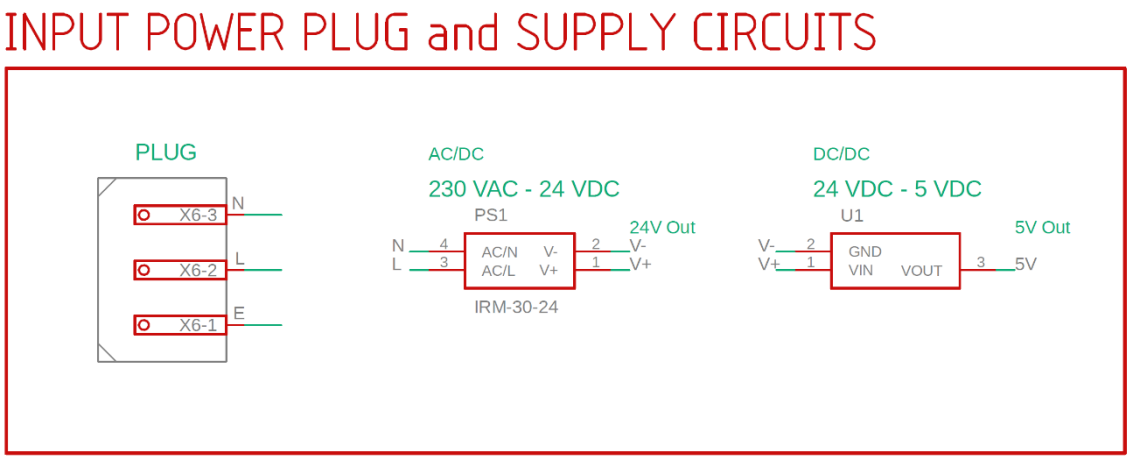

Figure 39 - Power Supply circuit schematic

The power supply circuit, seen in Figure 39, much like the energy meter's, is based upon converting the AC voltage from the supply grid to the required DC values. In this case, an AC/DC converter is used to supply 24VDC to the led strip and RF Switch. Another converter, this time DC/DC, takes advantage of the 24VDC output and down converts it to 5V to supply the FiPy and the temperature/humidity sensor.

The power supply for the fan (230VAC) and for the led strip (24VDC) first go through separate relays to enable control by way of FiPy inputs dictated by user commands via a proprietary GUI developed in Controlar. The relays are in Normally Closed (NC) mode, which means that, when powered off, they are always conducting power to the output(s). When voltage is applied the relays disconnect and the output(s) and switched off.

The circuits in question are shown in the schematic in Figure 40. The grey section represents the relay board used, which is represented in Figure 41.

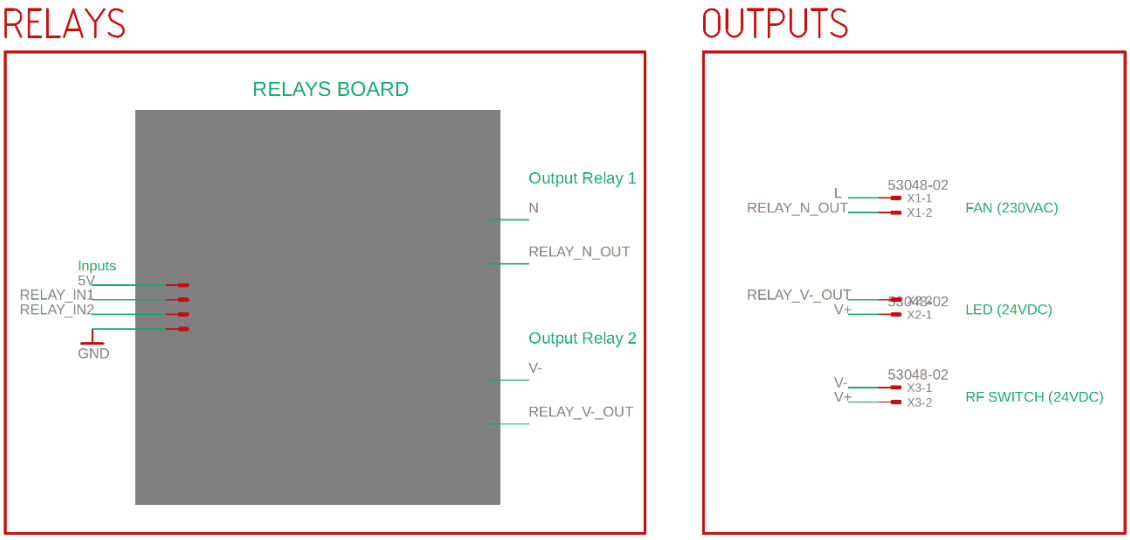

Figure 40 - Relay board and output circuit schematic

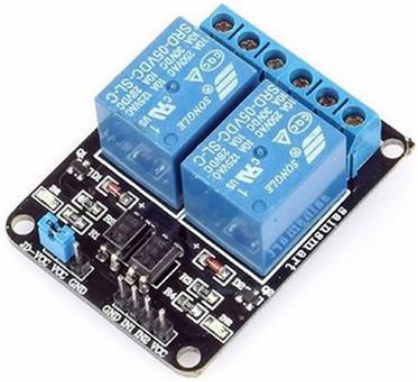

Figure 41 - Relay board

The temperature/humidity sensor utilized is a DHT11 [46], which is powered at 3 – 5.5V and communicates to the FiPy via a single-wire two-way serial bus. It has four pins but only three are connected. Further will be discussed about data types and processing in the firmware section.

Lastly, there are connections that need to be established to the MCU so that data processing, communication and actuation procedures can occur. Again, the FiPy will be connected via female headers, so that it is more accessible. The few needed connections are pretty straightforward. 5V will be connected from the AD/DC output, the same for GND which will connect to V-. The remaining connections are to the DHT11 data line, and to the inputs of the relay board for individual relay control. The schematics of the circuits for FiPy and DHT11 are shown in Figure 42.

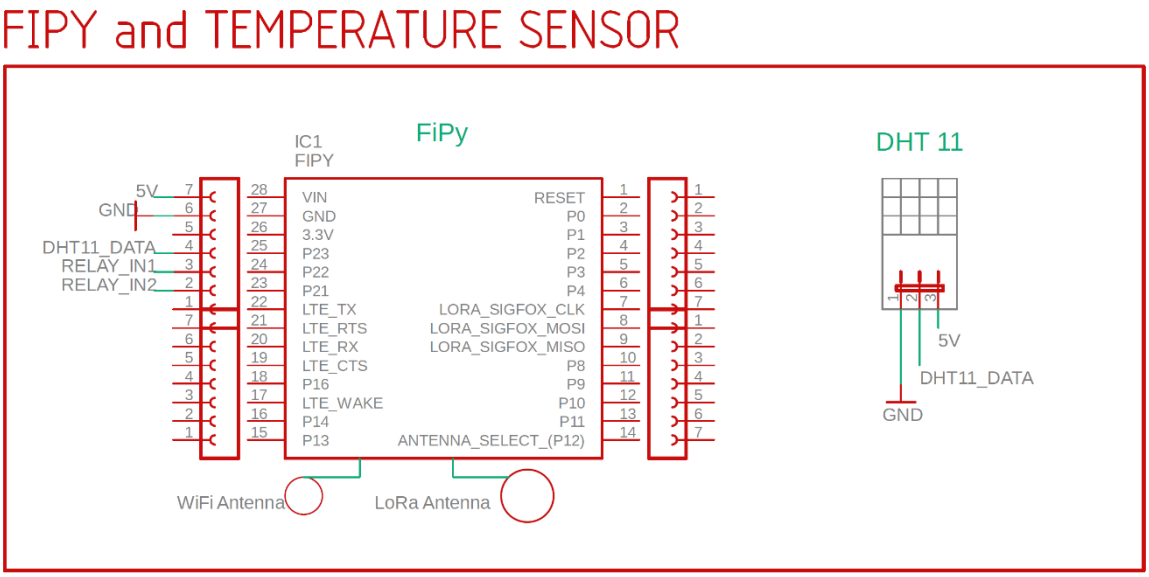

Figure 42 - FiPy and DHT11 sensor connections

A.1.2 -Firmware

The firmware used for this application shares much of the same code used for the energy metering one. The overall firmware structure is the same, meaning there is both a `boot.py` and a `main.py` that run sequentially, and whose purposes are the same. The first sets up communication protocol configurations and the second contains the main infinite loop cycle that will run during the lifetime of the device. Additionally, there are three libraries that include multiple functions that are required for correct operation of the device. Two of them are shared with the previously mentioned application, `mqtt.py` and `cayennelpp.py`, since both of them will use the same communications. The other one is dht11.py and will be used to configure the sensor and communicate with it.

A DHT11 transmission in its integrity is composed by 40bit. The first 16 will be dedicated to the humidity values, 8bit for decimal part and the remaining 8bit to the integral part. Additional 16bit will be used for temperature values, adopting the same allocation for decimal and integral parts. The remaining 8bit will be for a checksum that will check the validity of the transmission.

In the `dht11.py` library there is a primary read method that will need to resort to auxiliary functions to collect the inputs, parse the data and calculate and validate the checksum before returning valid information about environment temperature and relative humidity.

To enable actuation, the downlink messages from the server will need to be interpreted. Every function relating to actuation will be featured in main.py script. First, MQTT messages

that are received via WiFi will need to be parsed in order to extract only the portion containing the data. As it is in JSON format, it will contain a lot of metadata that will not be required for the actual interpretation of the user inserted input command. Next, when messages arrive via MQTT, the data section is encoded in base64 and a decode message will need to be set in place to perform the task. Meanwhile, through LoRaWAN, messages are already decoded

A.1.3 -Final Prototype

Following the work developed for this implementation, it was time to assemble the final prototype and perform tests for the accuracy of the sensor readings and the validity of the actuation methods in the firmware.

In Figure 43, it is displayed the arrangement of the different components in the casing that were included in the machine to be featured in the exhibition. The AC/DC (black box) takes up most of the available space. In the bottom, near the three outputs, resides the FiPy, with both SMA to uFL adapters for the two antennas that will be placed in the top edge of the box (for LoRaWAN and WiFi). Lastly, the relay board will be placed on top of the remaining DC/DC converter and the temperature/humidity sensor, which turned out less than ideal.

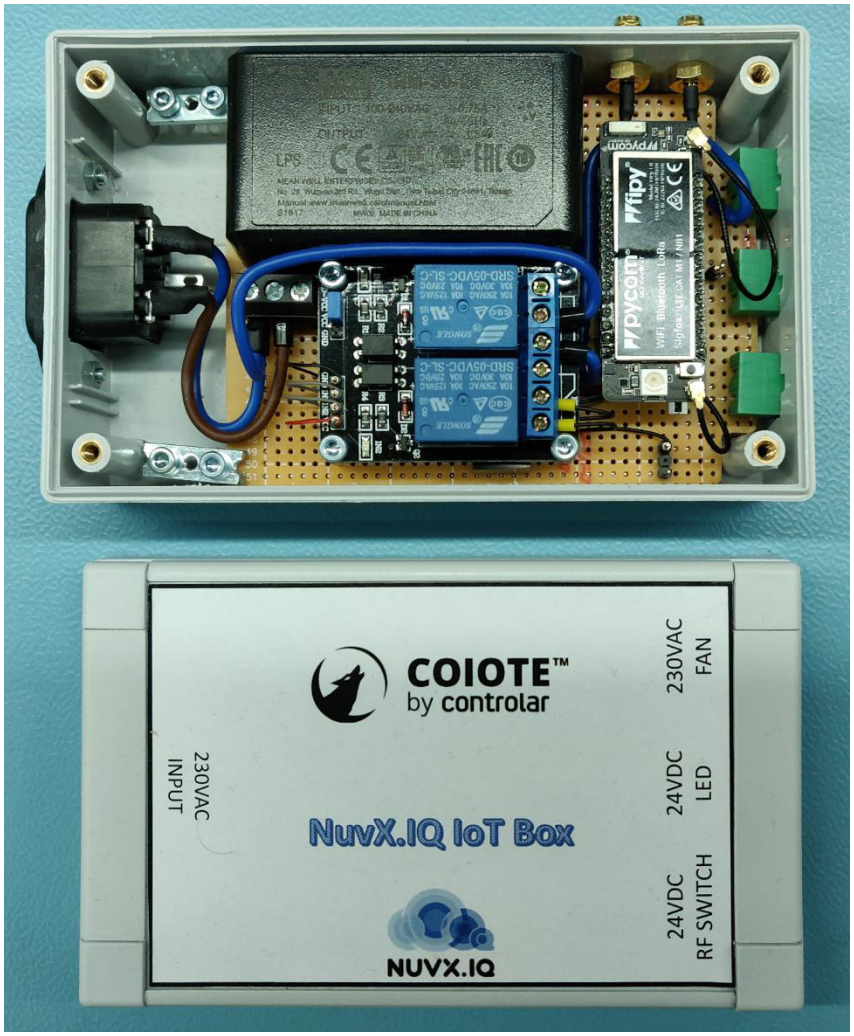

Figure 43 - Final Prototype

As it is demonstrated in Figure 44 and Figure 45, the sensor will get excessively hot and will interfere with the ability to correctly measure both temperature and humidity. The temperature rise will imply a decrease in relative humidity.

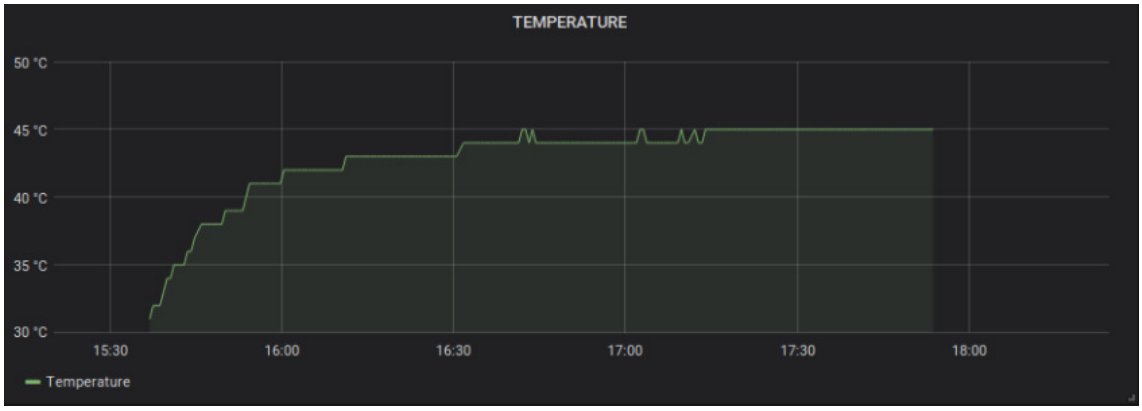

Figure 44 - Acquired values for Temperature

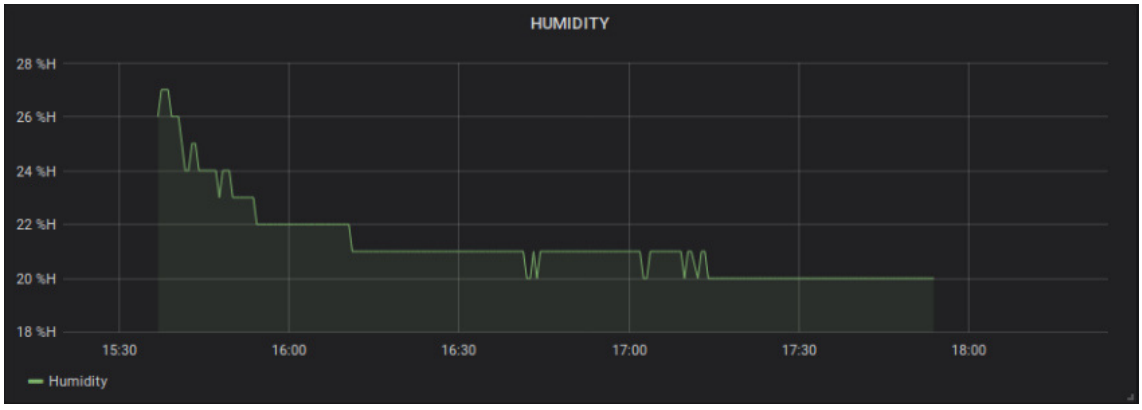

Figure 45 - Acquired values for Relative Humidity

A.2 - Annexed Paper

Pedro Ferreira, Rafael N. Miranda, Pedro Miguel Cruz and Hélio S. Mendonça
ICEAA-IEEE APWC 2019, Granada, Spain, 9 – 13 September 2019

Multi-Protocol LoRaWAN/Wi-Fi Sensor Node Performance Assessment for Industry 4.0 Energy Monitoring

Pedro Ferreira
Dept. of Electrical and Computer Engineering, University of Porto
Porto, Portugal
up201308037@fe.up.pt

Rafael N. Miranda
Controlar S.A.
Alfena, Portugal
rafaelneves.miranda@controlar.pt

Pedro Miguel Cruz
Controlar S.A.
Alfena, Portugal
pedro.cruz@controlar.pt

Hélio S. Mendonça
Dept. of Electrical and Computer Engineering, University of Porto
Porto, Portugal
hsm@fe.up.pt

***Abstract*—This paper describes the implementation of an end-to-end Internet of Things (IoT) solution, focusing specifically in the multi-protocol sensor node with LoRaWAN and Wi-Fi connectivity options (Pycom's FiPy). A performance assessment will be presented, addressing a comparison between the different protocols (LoRaWAN vs. Wi-Fi) in terms radio coverage, timing issues, among others.**

Further, it will be investigated the integration onto the sensor node of sensor/actuator circuit blocks for energy metering, supported on Microchip's ATM90E26 single-phase meter. This will provide a practical use case in the field of Industry 4.0, leading to preliminary insights for power quality monitoring.

Keywords—Industry 4.0, Internet of Things, LoRaWAN, Wi-Fi, energy monitoring.

I. Introduction

THE overwhelming increase in connected devices in recent years, [1, 2], has allowed for the growth of Internet of Things (IoT) technology by taking advantage of this connectivity and merging it with already existing functionalities such as identification, various sensing and actuation options, signal processing and data analytics [3], thus creating a multipurpose widespread network that is changing today's digital transformation paradigm. Currently, a big diversity of fields (transportation, agriculture, smart cities, manufacturing / Industry 4.0, smart homes, etc.) is being influenced and developed in the basis of new IoT arrangements, with the focus of this paper being centred on industrial solutions and new potential applications.

Industry 4.0 refers to the most recent revolution in the industry segment, in which a complete transformation takes place. It involves connectivity and integration with the digital intelligent realm of IoT, where robotics and automatization are the core aspects. It is necessary for such systems to be constantly connected, with human assistance and able to make decisions in a decentralized manner. The major components of Industry 4.0 include cyber-physical systems, edge and Cloud computing, data analytics, additive manufacturing, and so on [4].

Coupled with the surge of Industry 4.0 comes a greater awareness and concern for energy efficiency, both in terms of a consumer behaviour shift that favours more environmentally friendly practices and as it could also prove to be an enormous advantage over the competition, because energy costs are on the rise and it is impossible to turn a blind eye to more energy efficient manufacturing processes [5].

This paper approaches IoT and digital transformation as one of the means toward this greener goal, due to its sensing capabilities and the capacity to offer huge amounts of data around energy monitoring in quasi real-time operation. With this is mind, the solution presented here depicts a low-cost approach for an energy metering solution, employed at machine level, which can track and measure detailed energy consumption parameters in real-time.

The gathered data will be further processed after being sent through dedicated gateways to a central local server that will provide integrations of specific applications with user interfaces. Being a multi-protocol solution, the proposed implementation gains on both protocols' advantages, performing better in different use cases. When available, Wi-Fi connectivity will allow for greater data payloads and more frequent uplink transmissions, and to give real-time insights onto the network-grade performance by using available mobile devices in a ubiquitous action connecting directly to applications on the central server. Complementarily, LoRaWAN will make use of its low-energy consumption and associated long range coverage to cover at once big industrial plants, promoting an integrated optimization of entire factory.

The paper is organized as follows. Section II outlines the architecture implemented for this project (COIOTE) and the different elements that compose it. Next, Section III presents the gathered data relating to Wi-Fi and LoRaWAN radio coverage for a common industrial building. In Section IV, an in-depth characterization of the energy metering approach under use is provided. Lastly, Section V will show real measurement results for an exemplificative load (60 W lamp), which are validated by a commercially available energy meter, and finally some conclusions are drawn.

II. Architecture of Implemented Solution

This section presents a brief overview of the architecture proposed for the end-to-end IoT solution, as well as a brief rundown of the different blocks, covering the multi-protocol node, gateway, central server and user interface application. Fig. 1 presents a blueprint of the topology used.

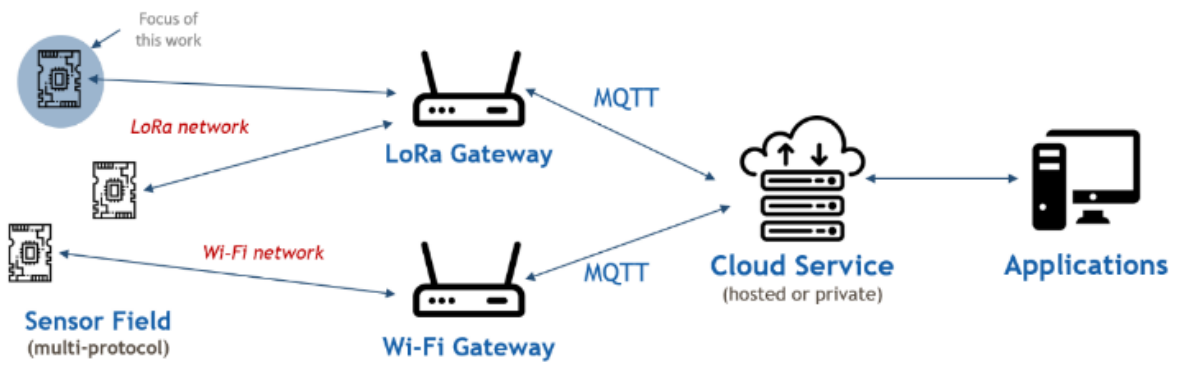

Fig. 1. Complete architecture of an end-to-end IoT solution with multi-protocol sensors.

The implemented IoT architecture is mostly based on open-source hardware and software solutions, as follows:

- **Application UI:** using free version of InfluxDB and Grafana to visualize obtained data with timestamp embedded;
- **Central Server/Cloud:** employing an open-source version of LoRa Server [6] with specific adaptations for this multi-protocol solution and considering MQTT over TCP [7] communication with the Gateway;
- **Multi-Protocol Gateway:** a single entity based on Raspberry Pi 3 Model B+ with an additional LoRaWAN concentrator board and proprietary software to allow multi-protocol LoRaWAN/Wi-Fi functioning;
- **Sensor Node:** based on Pycom's FiPy [8] device with multiple protocol capability but focusing on LoRaWAN/Wi-Fi communication modules.

As the main focal point of this work is on the multi-protocol sensor node, it will be described in more detail its capabilities in terms of the connectivity modules for LoRaWAN and Wi-Fi.

A. *LoRaWAN Module*

LoRaWAN protocol was selected as one of the connectivity options for this solution mainly due to its long-range radio coverage and possibility of investigation under industrial scenarios where it is less explored but showing a huge potential.

LoRaWAN operating frequency band, in Europe, is in the 863-870 MHz range. Operation in this ISM band is regulated and that means a couple of operating rules must be obeyed, with one of the most important being the duty-cycle allowed for uplink and/or downlink transmissions. In Europe the duty-cycle must follow the stipulations in section 7.2.3 of the ETSI standard EN 300 220 [9]. Such restrictions apply not only to the sensor nodes but also to gateways along the way, as they both transmit on the same frequency channels.

This European standard also defines maximum transmit power of +14 dBm, allowed spreading factors (SF7-SF12), bandwidths (125 or 250 kHz), along with other important parameters/metrics, as well as available sub-bands for operation, with specific duty-cycle conditions in each one, as follows:

- g (863.0 – 868.0 MHz): 1%
- g1 (868.0 – 868.6 MHz): 1%
- g2 (868.7 – 869.2 MHz): 0.1%
- g3 (869.4 – 869.65 MHz): 10%
- g4 (869.7 – 870.0 MHz): 1%

FiPy's LoRaWAN transceiver operates in the g1 range of frequencies, meaning it is forced to comply to 1% of duty-cycle restrictions. Later, in Section IV, one will address specific strategies to benefit accomplishment of such low duty-cycle conditions via the construction of lightweight payloads with an open-source approach.

It is worth noting that in the Gateway it is implemented software to forward all the LoRaWAN packets onto the central server using a compliant MQTT implementation.

B. *Wi-Fi Module*

Shortly, for the Wi-Fi connectivity the full capacity of FiPy's module is used and the node's firmware is implemented in such a way that messages can be published and subscribed with an MQTT-compliant communication. Wi-Fi coverage is provided by the multi-protocol gateway, when it is operated in the access point (AP) mode and allowing direct management via a web-based UI.

It is important to make note of the fact that Wi-Fi communication is limited to the 2.45 GHz ISM band, which encompasses that only IEEE 802.11 b/g/n versions are valid.

As a final comment, such an implementation dictates that a common baseline is followed in both connectivity options (LoRaWAN and Wi-Fi) from the gateway upwards, being just MQTT messages up and down.

III. Coverage Results and Discussion

In the context of COIOTE project, a simple coverage analysis of one illustrative industrial building was executed.

It is quite important to understand initial network radio coverage for both connectivity options, so that optimization could be employed or rearrangement of any piece in the implemented IoT solution.

Fig. 2. Coverage analysis performed at several measurement points for a common industrial ground floor, composed of open spaces, electrical/electronic machinery and walls separating the different divisions.

Such an evaluation of the network coverage is vital for the correct operation of the system and allows not only to validate the device being used but also to give a better perception of the differences between LoRaWAN and Wi-Fi coverages on mixed indoor/outdoor industrial environment.

This basic coverage study was based on obtained radio signal strength indicator (RSSI) metric for both LoRaWAN and Wi-Fi networks, being created by the CCOIOTE's multi-protocol gateway. It was performed with a special firmware running on the implemented FiPy's node, which records those values into the memory for offline processing. The data composed of position and respective RSSI values, is being sent also directly to the central server for further corroboration.

Looking at Fig. 2, it is possible to check the different coverage conditions for LoRaWAN and Wi-Fi in various positions of the industrial building. Making a rapid assessment with a simple division in green, yellow and red colours, it is crystal clear the larger coverage range provided by the LoRaWAN network, as expected. To exemplify, there are some of the measured points where the communication via Wi-Fi is not possible at all, since the COIOTE gateway AP is not being detected at all. Such situation was already expected due to external concrete walls of the industrial building.

It is worth mentioning that LoRaWAN network was run on data rate zero (DR0), i.e. configuring the spreading factor to SF12 and a bandwidth of 125 kHz, which translates into an indicative bit rate of just 250 bit/s (lowest data rate encoding available [10]).

Finally, it is showed that there is a greater reliability in LoRaWAN because it uses a lower ISM band that is far less subjected to interferences from other networks, becoming more common within factory environments like crowded Wi-Fi industrial areas.

IV. Energy Metering Application

In order to push the validation to the node's operation, an energy metering use case was considered and is detailed in this section, which explains up to some extent how the measurement process takes place and overall integration of sensing/measuring part with the communication and processing units assured by the FiPy's module.

A. *Voltage and Current Sampling*

The energy metering sensor selected for this application was a Microchip's ATM90E26 single-phase meter. This integrated chip (IC) provides several measurement variables, as: voltage rms, current rms, mean active and reactive powers, grid frequency, power factor, L line and N line phase angles, mean apparent power, among others, even though not all of those were considered in the current implementation. Additionally, actual system status conditions are also offered in the available information at IC's memory.

As per ATM90E26 datasheet's [11], the frequency accuracy is 0.01 Hz, and the other measurement accuracy resides around 0.5 %. This sensor also supports two current sampling circuits – for L and N lines – with separate gain configurations.

For this specific application, it was used the sampling of N Line current by using a commercial current transformer (CT) that connects with some external filter circuitry via a 3.5mm jack and goes into the sensor IC differential port. To take a sample of the AC voltage waveform, a very simple method was employed by connecting the 230 VAC power supply to a series of two 430 kΩ resistors, so that the voltage drops to an acceptable level, and it is also filtered before being connected to the sensor IC port. The schematic of employed circuits is shown in Fig. 4.

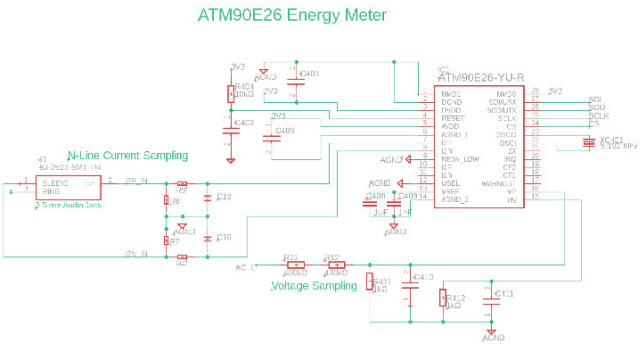

Fig. 3. Schematic of designed energy meter based on ATM90E26 single-phase meter, including voltage and current sampling circuits.

The ATM90E26 energy meter sensor provides selectable UART and SPI interfaces to the control unit. For sake of clarification, in the current design the SPI option was the selected method for this communication with the FiPy unit.

B. *Lightweight Payload Approach*

In order to comply with low duty-cycles imposed by LoRaWAN standard, an open-source approach for a low power payload byte-based was selected – Cayenne Low Power Payload (LPP). This approach resides in giving to each sensor a uniquely identifiable data channel byte (ID), followed by a data type byte that points the type of measurement to come in the following bytes, and each one has a specific data size linked to it, as per IPSO Alliance Smart Objects Guidelines. For this specific application, the data types used are displayed in Table I.

During a frame transmission, both time of emission and time on-air are recorded, with the intent of calculating the time in which the same sub-band is inaccessible, or it would potentially create packets to be lost forever due to blockages with other sensor communications. So, in order to calculate the time off one must employ (1).

$$T_{off\_sub\_band} = (T_{On\_Air} / duty\_cycle_{sub\_band}) - T_{On\_Air} \qquad (1)$$

Assuming a complete payload of 20 bytes with measured data plus the commonly fixed 13 bytes with LoRaWAN metadata, one can estimate the time on air for transmitting such a message and thus, obtaining then the maximum repetition period.

TABLE I. DATA TYPES AND DIMENSIONS UNDER USE

	IPSO Object Code	LPP Data Size [Bytes]
Voltage RMS	3316	2
Current RMS	3317	2
Grid Frequency	3318	2
Active Power	3328	2
Power Factor	3329	2

Table II shows calculations for different LoRaWAN data rates and Wi-Fi communication as a baseline for comparison. From the obtained metrics, it is possible to conclude that such values are very dependent on the payload size and data rate adopted, but with a careful selection, practical and quasi real-time repetition message periods are achievable and perfectly in line with a most demanding and of lower coverage range Wi-Fi based solutions.

TABLE II. COMPARISON OF MESSAGE TIME REPETITION FOR THE DIFFERENT CONNECTIVITY PROTOCOLS

	Time on Air	Repetition Period
Wi-Fi	N/A	every 1 or 2 seconds *(if needed)*
LoRaWAN @ SF12, 125 kHz	1811 ms	aprox. each **3 minutes**
LoRaWAN @ SF7, 125 kHz	72 ms	aprox. each **7 seconds**

Just for illustration and validation of the proposed solution, Fig. 4 shows a portion of data received at the central server application (Grafana) with a repetition period of 30 seconds.

Fig. 4. Application user interface with received data each 30 s with LoRaWAN configured to SF7-125 kHz data rate mode.

V. CONCLUSION

The proposed end-to-end and multi-protocol IoT solution was initially validated for industry 4.0 applications, and more specifically focused on an energy metering use case.

The implementation of a multi-protocol LoRaWAN/Wi-Fi sensor node based on FiPy's module was described, together with details for the realization of a practical energy metering system based on commercially available ATM90E26 single-phase meter. LoRaWAN was demonstrated as a feasible communication solution for quasi real-time operation of sensor/actuators within industrial scenarios, when carefully designed for low bit rates operation.

REFERENCES

[1] Statista, "Number of network connected devices per person around the world from 2003 to 2020", Available: https://www.statista.com/statistics/678739/forecast-on-connected-devices-per-person/. [Acessed June 3, 2019]

[2] Gartner, "Leading the IoT: Gartner Insights on How to Lead in a Connected World", Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf. [Acessed June 3, 2019]

[3] V. Galetić, I. Bojić, M. Kušek, G. Ježić, S. Dešić and D. Huljenić, "Basic principles of Machine-to-Machine communication and its impact on telecommunications industry", 2011 Proceedings of the 34th International Convention MIPRO, Opatija, 2011, pp. 380-385.

[4] Muhuri, P.K., A.K. Shukla, and A. Abraham, Industry 4.0: A bibliometric analysis and detailed overview. Engineering Applications of Artificial Intelligence, 2019. 78: p. 218-235.

[5] Mohamed, N., J. Al-Jaroodi, and S. Lazarova-Molnar, Leveraging the Capabilities of Industry 4.0 for Improving Energy Efficiency in Smart Factories. IEEE Access, 2019. 7: p. 18008-18020.

[6] LoRa Server, open-source LoRaWAN network-server, Available: https://www.loraserver.io/. [Acessed June 3, 2019]

[7] A. Banks and R. Gupta, "Message Queuing Telemetry Transport (MQTT) - version 3.1.1.", OASIS standard, 29 October 2014.

[8] Pycom, FiPy multi-protocol module, FiPy datasheet, Available: https://pycom.io/product/fipy/. [Acessed June 3, 2019]

[9] *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods*, ETSI EN 302 220, 2012.

[10] N. Sornin (Semtech), M.L.S., T. Eirich (IBM), T. Kramp (IBM), O.Hersent (Actility), LoRaWAN™ Specification. 2015, LoRa Alliance.

[11] Microchip/Atmel, Single-Phase High-Performance Wide-Span Energy Metering IC, Atmel M90E26 datasheet, 2014.

[12] Cayenne Low Power Payload (LPP), Available: https://www.thethingsnetwork.org/docs/devices/arduino/api/cayennelpp.html. [Acessed June 3, 2019]

References

1. *LoRaWAN Overview*. Available from: https://www.thethingsnetwork.org/docs/lorawan/.
2. *FiPy Docs*. june 2019]; Available from: https://docs.pycom.io/datasheets/development/fipy.html.
3. Committee, L.A.T., *LoRaWAN™ 1.1 Specification*. 2017.
4. Pycom. *FiPy*. june 2019]; Available from: https://pycom.io/product/fipy/.
5. Analytics, I. *The Top 10 IoT Segments in 2018 – based on 1,600 real IoT projects*. 2018; Available from: https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/.
6. Xia, F., *Wireless Sensor Technologies and Applications*. Vol. 9. 2009. 8824-30.
7. IEEE. *Towards a definition of the Internet of Things(IoT)*. 2015; Available from: https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf.
8. AVNET. *Sigfox is the world's leading provider of connectivity for the Internet of Things (IoT)*. Available from: https://www.avnet.com/wps/portal/apac/resources/technical-articles/article/iot/sigfox-is-the-worlds-leading-provider-of-connectivity-for-the-internet-of-things/.
9. Muhuri, P.K., A.K. Shukla, and A. Abraham, *Industry 4.0: A bibliometric analysis and detailed overview*. Engineering Applications of Artificial Intelligence, 2019. **78**: p. 218-235.
10. Ray, P.P., *A survey on Internet of Things architectures*. Journal of King Saud University - Computer and Information Sciences, 2018. **30**(3): p. 291-319.
11. Kocakulak, M. and I. Butun. *An overview of Wireless Sensor Networks towards internet of things*. in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. 2017.
12. Farhan, L., et al. *A Concise Review on Internet of Things (IoT) -Problems, Challenges and Opportunities*. in *2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), 18-20 July 2018*. 2018. Piscataway, NJ, USA: IEEE.
13. Statista. *Number of network connected devices per person around the world from 2003 to 2020*. [cited 2018 28 november]; Available from: https://www.statista.com/statistics/678739/forecast-on-connected-devices-per-person/.
14. Gartner. 2014 [cited 2018 27 november]; Available from: https://www.gartner.com/newsroom/id/2905717.
15. Galetić, V., et al. *Basic principles of Machine-to-Machine communication and its impact on telecommunications industry*. in *2011 Proceedings of the 34th International Convention MIPRO*. 2011.
16. *Disruptive Civil Technologies Six Technologies with Potential Impacts on US Interests out to 2025*. 2008.
17. *China sets 5b yuan fund for IOT industry.Available from:* http://www.chinadaily.com.cn/bizchina/2011-08/23/content_13172322.htm

18. *Definition of Internet*. Available from: https://en.oxforddictionaries.com/definition/internet.
19. *Internet Live Stats*. 3rd february 2019]; Available from: http://www.internetlivestats.com/.
20. Evans, D., *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*. 2011.
21. *IoT Number of Connected Devices Worldwide*. Available from: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.
22. *Global Internet of Things Market Size*. Available from: https://www.statista.com/statistics/485136/global-internet-of-things-market-size/.
23. *IoT European Platforms Initiative*. Available from: https://iot-epi.eu/projects/.
24. *IoT and Blockchain Convergence: Benefits and Challenges*.
25. Elijah, O., et al., *An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges*. IEEE Internet of Things Journal, 2018. **5**(5): p. 3758-3773.
26. Euractiv. *'Farming 4.0' at the farm gates*. 2016; Available from: https://www.euractiv.com/section/agriculture-food/opinion/farming-4-0-digital-technology-at-the-farm-gates/.
27. Comission, E., *Industry 4.0 in agriculture: Focus on IoT aspects*. 2017.
28. WiseCrop. Available from: https://www.wisecrop.com/pt.
29. Beyer, S.M., et al., *Pattern-of-Life Modeling in Smart Homes*. IEEE Internet of Things Journal, 2018. **5**(6): p. 5317-5325.
30. Marche, C., M. Nitti, and V. Pilloni. *Energy efficiency in smart building: A comfort aware approach based on Social Internet of Things*. in *2017 Global Internet of Things Summit (GIoTS)*. 2017.
31. Li, W., et al., *Implemented IoT-Based Self-Learning Home Management System (SHMS) for Singapore*. IEEE Internet of Things Journal, 2018. **5**(3): p. 2212-2219.
32. Mohamed, N., J. Al-Jaroodi, and S. Lazarova-Molnar, *Leveraging the Capabilities of Industry 4.0 for Improving Energy Efficiency in Smart Factories*. IEEE Access, 2019. **7**: p. 18008-18020.
33. *LoRa Alliance*. june 2019]; Available from: https://lora-alliance.org/.
34. SigFox. Available from: https://www.sigfox.com/en.
35. Ingenu. Available from: https://www.ingenu.com/technology/rpma/
36. *Weightless*. june 2019]; Available from: http://www.weightless.org/.
37. Semtech, *ETSI Compliance of the SX1272/3 LoRa Modem (AN1200.10)*. 2013.
38. Domazetović, B., E. Kočan, and A. Mihovska. *Performance evaluation of IEEE 802.11ah systems*. in *2016 24th Telecommunications Forum (TELFOR)*. 2016.
39. *Wi-Fi CERTIFIED 6™ coming in 2019*. Available from: https://www.wi-fi.org/news-events/newsroom/wi-fi-certified-6-coming-in-2019.
40. Atmel, *Single-Phase High-Performance Wide-Span Energy Metering IC, Atmel M90E26 datasheet*. 2014.
41. Atmel, *Single-Phase Energy Metering IC, M90E26 Application Note*. 2014.
42. Instruments, T. *SPICE-based analog simulation program*. june 2019]; Available from: http://www.ti.com/tool/TINA-TI.
43. *Trace Width Calculator*. 23/06/2019]; Available from: https://www.4pcb.com/trace-width-calculator.html.
44. *LoRaTools*. [cited 2019 may]; Available from: https://www.loratools.nl/#/airtime.
45. *ATM90E26 Breakout*. Available from: https://www.tindie.com/products/whatnick/atm90e26-breakout/.
46. *DHT11 Humidity & Temperature Sensor*. [cited 2019 may]; Available from: https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.