FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Development of a monitoring and data communication system for application in Pavement Energy Harvesting

José Gonçalo Correia de Sousa Neto

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

FEUP's Supervisor: Prof. Miguel Velhote Correia

Pavnext's Supervisor: Eng. Francisco Duarte

July 19, 2021

Resumo

As cidades que aplicam os dados recolhidos por sensores e dispostivos eletrónicos na sua gestão de recursos e serviços são conhecidas por cidades inteligentes (*smart-cities*). A introdução deste conceito remonta aos anos 70 do século XX ("A Cluster Analysis of Los Angeles"), mas encontra-se hoje mais atual do que nunca. Barcelona e Amesterdão são exemplos de cidades que começam a recolher dados de forma a gerir os seus recursos e operações de forma eficiente, como a iluminação pública ou estacionamento inteligente.

Relacionado com a ideia de cidade inteligente, a sustentabilidade urbana é também foco de diversas cimeiras e encontros internacionais. A utilização de fontes de energia renováveis ou de meios de transporte "zero emissões" são implementações práticas deste conceito.

Deste modo, a Pavnext desenvolveu e patenteou a nível internacional um sistema de *energy harvesting* para aplicação em pavimentos rodoviários. Este sistema permite, em primeiro lugar, reduzir a velocidade de circulação de veículos sem depender da ação dos condutores e sem induzir impacto nos veículos, funcionando pela extração de energia cinética de veículos de forma não agressiva para os condutores e demais ocupantes. A energia captada é depois convertida em energia elétrica, com uma elevada eficiência de conversão, através da mesma tecnologia. Esta energia pode depois ser utilizada para diversas aplicações, quer no local, quer em diferentes pontos de consumo (sustentabilidade). Para além disto, foi desenvolvido um sistema que permite monitorizar múltiplos dados de tráfego e energia, os quais poderão ser utilizados para otimização de recursos energéticos e para melhorar os níveis de gestão da cidade, segundo a ideia de *smart city*.

O trabalho desenvolvido consiste num sistema de monitorização e informação que permite registar e comunicar os dados adquiridos pelos vários sensores implementados nos módulos da Pavnext. A componente *local-master* comunica com os vários módulos aplicados no pavimento, que funcionam como *slave*s, criando assim uma estrutura *master-slave*, escalável para algumas dezenas de módulos. Múltiplos *local-master* reportam a informação a um *master* que a reencaminha para uma *gateway* LoRa. Os dados recebidos no *master* são enviados para uma base de dados, sendo devidamente processados e apresentados numa aplicação *web*.

A arquitetura sugerida e implementada foi testada e avaliada nos protótipos da Pavnext, tendo sido obtidos os resultados pretendidos – fazer chegar a informação recolhida aquando da ativação dos módulos à aplicação de apresentação de dados.

Abstract

Cities that apply data collected by sensors and electronic devices in their management of resources and services are known as smart-cities. The introduction of this concept goes back to the 70th decade of the last century ("A Cluster Analysis of Los Angeles"), but it is part of our lives now more than ever. Barcelona and Amsterdam are examples of cities that started gathering data to manage resources and operations in an efficient way, like streetlights or smart parking.

Related with the idea of smart-city, urban sustainability is the theme of several summits and international meetings. Renewable energy sources or zero-emission transport are practical implementations of this concept.

Considering this, Pavnext developed and patented internationally an energy harvesting system for road pavements. This system allows the reduction of vehicles' speed without the driver's action while avoiding damage to the vehicle, by extracting its kinetic energy in a non-aggressive way. The energy gathered is converted, with a high efficiency rate, in electric energy, which can be used in many applications, either locally or not (sustainability). Furthermore, a traffic and energy data monitoring system was developed, which can be used to optimize energy resources and to improve the city's management, following the idea of smart-city.

The work developed consists of a monitoring and information system, responsible for storing and communicating the data acquired by the sensors implemented in Pavnext's modules. The *local-master* component communicates with the modules in the pavement who act as *slave*s, establishing a *master-slave* structure that is scalable for numerous elements. Multiple *local-master* report the information to a *master*. The data received by the *master* is forwarded to a LoRa gateway. The gateway sends the data to a database, which is later processed and displayed in a web application.

The architecture suggested and implemented was tested and evaluated in Pavnext's prototypes and the intended results were obtained – getting the information collected when activating the modules to the data presentation application.

Acknowledgments

I would like to thank Pavnext's supervisor, Eng. Francisco Duarte for his support during the last few months. I learned a lot from him and without his help I could not have complemented the task at hands.

The insights provided by Prof. Miguel Correia were also very important, guiding me on which aspects I should focus my time and resources. Thank you.

Obviously, I could not have done any of this without the support of my family and friends, to whom I thank as well.

José Neto

Contents

List of Figures

List of Tables

Abbreviations and Symbols

API	Application Programming Interface
ASC	Asphalt Solar Collectors
CAN FD	Controller Area Network Flexible Data-Rate
CCTV	Closed-Circuit Television
DBMS	Database Management System
IMU	Inertial Measurement Unit
ITS	Intelligent Transportation System
IoT	Internet of Things
LoRa	Long Range
M2M	Machine to Machine
MCU	Microcontroller
MEM	Microelectromechanical
NB-IoT	Narrowband IoT
REST	Representational State Transfer
RSU	Roadside Unit
TEG	Thermoelectric Generator
TFMS	Traffic Flow Monitoring System
UAV	Unmanned Aerial Vehicle
UHI	Urban Heat Island
V2X	Vehicle-to-Everything

Chapter 1

Introduction

Pavnext's prototypes are capable of generating electrical energy from vehicle's mechanical actuation of the hydraulic system. However, it lacks an architecture capable of getting the data from the ground to a graphical interface. This thesis will describe how that system was implemented, from the hardware to the software behind its operation.

1.1 Motivation

The smart city concept has became a prominent idea when planning a city's future and its management of resources and services. Road safety has been improving with the introducing of traffic safety measures in zones around schools, hospitals and residential areas. These measures are accomplished with the implementation of speed reduction techniques, such as speed bumps.

The Internet of Things can help this technologies reach another level, by providing useful data to the city's administration and citizens. By combining different structures a network of information can be formed, maximizing the city's efficiency and organization. It can be applied in situations such as traffic control and energy monitoring.

1.2 Objectives

As referenced earlier, Pavnext's solution lacks an infrastructure that gathers the information about a passing vehicle – its speed and energy generated – and presents it to its administrator. This project aims to fulfil that gap, increasing the product's value to potential stakeholders, such as cities interested in the smart city concept.

1.3 Conception

The hardware that was used in the system was selected after comparing the solutions available on the market. Circuit boards were projected, printed and soldered in order to assemble the subsystems into a compact and reliable design.

Once the scope of the project was determined, tests started being performed, evaluating the prototype's mechanical behaviour and its generation of energy. Then, as each subsystem was developed, it was integrated into the system and its functionalities properly assessed.

1.4 Thesis outline

The thesis is structured as follows:

1. Chapter 2 presents the relevant state of the art, focusing on Road Pavement Energy Harvesting technologies, traffic monitoring systems in smart-cities, IoT solutions, data processing/storage and databases, servers and hosting;
2. Chapter 3 presents a description of the work developed, detailing its software and hardware implementation;
3. Chapter 4 presents the results of the tests performed;
4. Chapter 5 concludes this report.

Chapter 2

State of the art

In this chapter, several Pavement Energy Harvesting technologies based on solar power or vehicles' mechanical energy will be introduced. Then, traffic monitoring systems in smart cities will be addressed given the importance of road safety in urban areas and the increasing traffic congestion.

The project's specifications require an acquisition system capable of estimating the energy generated by vehicles, so data/signal acquisition and local/global processing and storage technologies are addressed.

Lastly, IoT and database solutions are compared, drawing conclusions about their advantages and disadvantages.

2.1 Road Pavement Energy Harvesting

Sustainability is the idea behind many technological solutions developed in the last decades. Energy harvesting is one of the implementations of this concept, which can be described as the conversion of ambient energy present in the environment into electrical energy, typically generating power in the range of micro to milli-watts [11].

Based on works from Dawson et al. [5], Duarte and Ferreira [1] and Wang et al. [12], Wenjuan Sun et al. [7] summarized energy harvesting technologies for sustainable pavements.

The main sources of energy associated with Pavement Energy Harvesting are the continuous exposition of road pavements to solar radiation and vehicles loads [1], as depicted in figure 2.1.

2.1.1 Solar energy harvesting on road pavements

The demonstration of the photovoltaic effect by Edmond Becquerel in 1839 was followed by Charles Fritts invention of the first photovoltaic cell in 1883, with an efficiency of 1%. The following decades were rich in technological development and less than a century later, in 1958, it started being used to power space exploration equipment, specifically by the Earth-orbiting satellite Vanguard 1 [13].

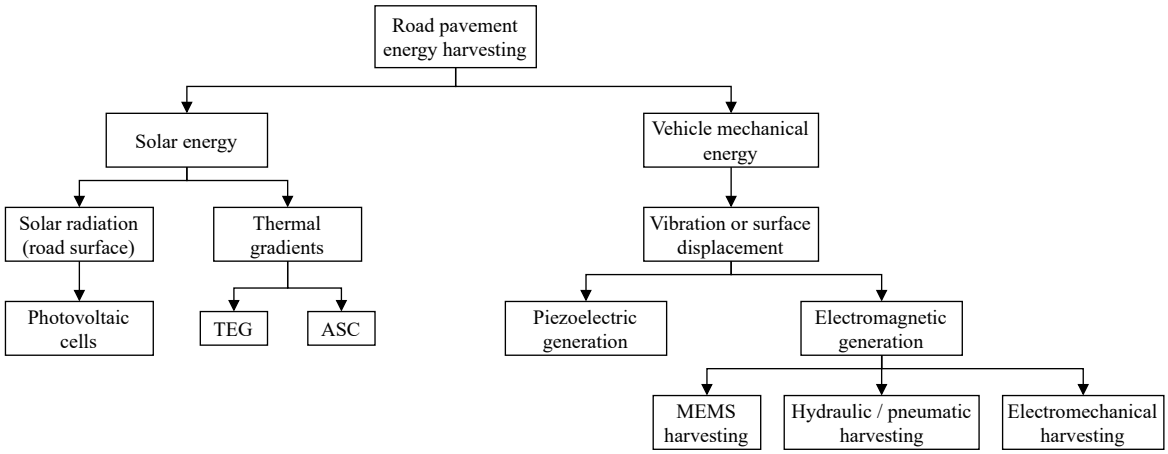


Figure 2.1: Road Pavement Energy Harvesting technologies (adapted from [1])

Solar energy harvesting is often associated with the photovoltaic effect and solar cells, but other methods can be implemented, as shown in the left side of diagram 2.1. Figure 2.2 extracted from [7] depicts the most prevalent methods to harvest solar energy, based on patent US7317405B2 [2] and works from Dawson et al. [5], Efthymiou et. al [3], Poon [4] and García and Partl [6].

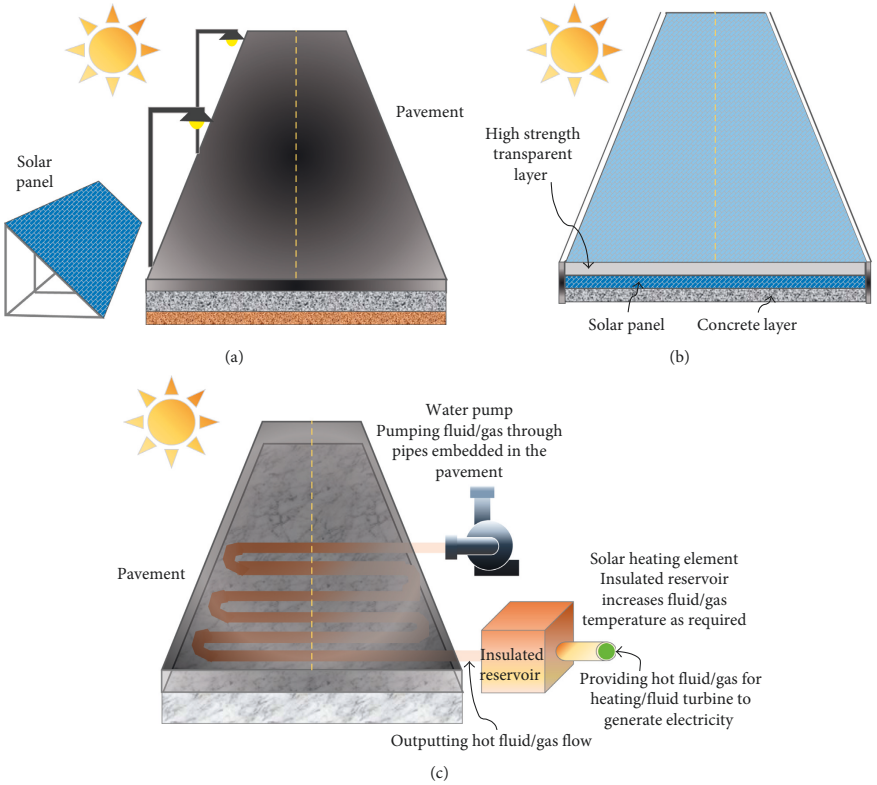


Figure 2.2: Solar energy harvesting technologies for pavements: (a) solar panel (based on [2]), (b) photovoltaic pavement (based on [3, 4]), and (c) heat extraction with the piping system (fluid/gas) in pavements (based on [5, 6]) (extracted from [7])

2.1.1.1 Photovoltaic technology

The installation of solar panels along roadways to power up road signs, such as traffic lights or speed limits signals is common nowadays. An example is depicted in figures 2.2 (a) and 2.3 from Soltráfego[1].

Figure 2.3: Solar-energy-powered traffic sign from Soltráfego (extracted from Soltráfego[1])

A different approach was followed by SolaRoad that developed a photovoltaic pavement surface (2.2 (b)) and implemented it in a bicycle path in 2014. Later, in 2019, it was tested in road pavements, as seen in figure 2.4 from SolaRoad[2]. Despite the technological advance imposed by this invention, the real-world implementation faced some criticism because of the damage it suffered in the following years and was removed in November 2020 [14]. A similar approach was followed by Solar Roadways [15].

Figure 2.4: SolaRoad's photovoltaic pavement surface in Haarlemmermeer (NL), 2019 (extracted from SolaRoad[2])

Due to its functional specification, photovoltaic technology is limited by atmospheric and illumination conditions. Despite this, it is considered to have the greatest peak productivity among all other technologies [7, 16].

2.1.1.2 Thermal gradients

Duarte and Ferreira [1] and Wenjuan Sun et al. [7] studied technologies to harvest solar energy using the thermal gradients registered in pavements, such as asphalt solar collectors (ASC) and thermoelectric generators (TEGs).

[1] Soltráfego's website
[2] SolaRoad's website

Asphalt solar collectors technology

In order to harvest the thermal energy generated by the absorption of solar radiation by road pavements, a heat extraction method using the piping system has been studied (Bobes-Jesus et al. [17]), as depicted in figure 2.2 (c) from [7].

New applications for ASCs other than merely generating electricity have been proposed. Nasir et al. 2020 [18] and Chiarelli et al. [19] studied the effect of ASC in urban heat island (UHI), a phenomena that occurs when *"a metropolitan area is a lot warmer than the rural areas surrounding it"* (National Geographic: Urban heat island[3]), up to 8 °C difference (Nasir et al. 2015 [20]). The study concluded that *"the ambient air temperature within the urban canyon was reduced by up to* 4.67 °C *and the surface was reduced up to 27% as a result of the addition of the urban road pavement solar collector"* [18].

De-icing techniques have also been developed using thermal energy harvested from pavements and applied in Switzerland (SERSO system) and the Netherlands (RES, 2007) [1].

Thermoelectric technology

Temperature differences between two points of a conductive material cause the electromotive force behind the Seebeck effect [21], which can be used by TEGs to produce electrical energy. Guo and Lu [22] identified two methods to harvest energy from pavements using thermoelectric technologies: a) pipe systems and b) thermoelectric modules in the pavement.

Hasebe et al. [23] obtained an output of 5 W with $\nabla T = 40.5\,°\text{C}$ using a TEG and a water piping system incorporated into the pavement.

Rohani et al. [24] tested a thermoelectric energy harvesting system both theoretical and experimentally with a prototype. The system is composed of a TEG and thermally conductive rods for heat transfer. Different generators, circuits and rod designs where investigated, obtaining 42 mW with $\nabla T = 47.8\,°\text{C}$ using the best device configuration.

Multiple studies have been conducted and other system implemented. However, the potential damage to the infrastructure and the low conversion rate of this technology make it not market-ready as of now [1, 22].

2.1.2 Vehicles' mechanical energy harvesting on road pavements

Road pavements are subject to mechanical forces caused by moving vehicles, inducing vibration or surface displacement. Different technologies can be applied to harvest the energy generated during this process: piezoelectric and electromagnetic.

[3] National Geographic: Urban heat island

2.1.2.1 Piezoelectric technology

The mechanical stress applied to certain materials is responsible for the generation of an electric charge, a phenomena known as the piezoelectric effect [25].

Duarte et al. [1] and Correia and Ferreira [26] performed a literature review on the development of piezoelectric technologies that can be applied to energy harvesting on pavements.

Zhao et al. [27] developed a solution that generated 1.2 mW at 20 Hz with a conversion efficiency lower than 15% for one vehicle passage. Wang et al. [28] obtained 50.41 mW during experiments of their design.

Different patents have been registered in this domain, which are described in table 2.1.

Table 2.1: Patents' description

Patent no.	Title	Authors	Year
WO2009098676A1 [29]	Energy harvesting	Haim Abramovich, Eugeny Harash, Charles Milgrom and Uri Amit	2009
CA2715129C [30]	Energy harvesting from roads and airport runways	Haim Abramovich, Eugeny Harash, Charles Milgrom, Uri Amit and Lucy Edery Azulay	2011
WO2015157377A1 [31]	Piezoelectric energy harvesting systems and methods	Ira L. Campbell	2015
CN112468019A [32]	Road surface impact energy harvesting device and calculation method	Xiaoguang Yua	2021

2.1.2.2 Electromagnetic technology

Another approach to generate electrical energy from vehicles are electromagnetic generators based on Faraday's law. This law states that an electric current is induced if an electric conductor is moved in relation to a magnetic field. The harvesting system can be based on hydraulic, pneumatic, electromechanical or microelectromechanical (MEM) systems.

2.1.2.3 Pavement Energy Harvesting technologies comparison

Table 2.2 – based on [12] – compares the different energy harvesting technologies, identifying their advantages and disadvantages.

2.1.3 Energy monitoring systems

The Pavement Energy Harvesting technologies discussed usually include a monitoring system capable of estimating the energy produced. Al-Turjman et al. [33] divided energy monitoring technologies in two categories:

Table 2.2: Pavement Energy Harvesting technologies comparison

Technology	Advantage	Disadvantage
Photovoltaic	High energy output	High cost; fragility to traffic loading
Thermal gradients (ASC | TEG)	Wide applicability (e.g. de-icing) UHI mitigation	Low efficiency High cost per unit
Piezoelectric	Mature technology (research and patents)	High cost for high energy output
Electromagnetic	Small size; does not require voltage to operate	Low voltage output

- Direct sensing: electric current and voltage are measured directly;
- Indirect sensing: the emitted energy is measured using interference engines, such as magnetic or acoustic sensors:
  - Magnetic sensors: detect changes in the magnetic field around the input power cable of the appliance and reports the amount of energy used;
  - Acoustic sensors: analyse the noise pattern of an appliance to determine its usage status. For example, a drop on the sound level can be perceived as the termination of the appliance.

Gopal et al. [34] proposed a IoT based solar power monitoring system using the direct sensing paradigm. The acquisition system was composed of voltage and current sensors connected to an Arduino Uno that was attached to a ESP8266 that provided data to a mobile application.

2.2 Traffic monitoring systems in smart cities

In the scope of the smart city concept, traffic management technologies have been developed in the last decades. Timed traffic lights have progressively been replaced by automatic algorithms to reduce the impact of transport on the urban environment, such as pollution caused by vehicles waiting unnecessarily for permission to cross [35]. Reducing traffic congestion and *bumper-to-bumper* situations leads to less consumption of fuel [36].

Traffic management is accomplished recurring to traffic monitoring systems capable of counting the number of cars, their frequency, direction, etc. Intrusive and non-intrusive sensing methods were identified by Omar et al. [37]. The former is composed of sensors installed on the asphalt, such as pneumatic road tubes or weigh-in-motion sensors. Non-intrusive technologies can be cameras and the associated image processing system, radar (laser or microwave), etc.

Omar et al. [37] proposed a traffic congestion monitoring and control system to be implemented in an intersection using a camera and a microcontroller.

Considering the limitations of fixed cameras' field of view, resulting in blind spots that can not be covered, Frías et al. [38] proposed a cooperative system using Unmanned Aerial Vehicles (UAVs). The algorithm calculates and sends to each vehicle its 2D real-world position, maximizing the driver's information about its surroundings.

Latif el al. [39] proposed an intelligent traffic monitoring and guidance system by using graph theory, formal methods and IoT.

Traffic Flow Monitoring Systems (TFMSs) have been used in situations such as vehicle location and theft prevention and to anticipate traffic congestion. TFMSs are part of the Intelligent Transportation System (ITS) concept, which aims to improve smart cities movement, security and road's throughput [40]. ITSs can implemented using the traffic signal control system, automatic number plate recognition, speed cameras, closed-circuit television (CCTV) systems, vehicle-to-everything (V2X) communication and Roadside Units (RSUs) [41].

The RSU processes and analyses data from cameras and radars in order to help optimize traffic flow and improve passenger and pedestrian safety [40]. Zheng et al. [42] studied the RSU placement problem for the TFMS, namely the minimum number of RSUs to place in order to maintain efficiency.

Urban mobility has been a decisive point when choosing where to live. City residents are tired of wasting time commuting on endless traffic jams. In February 2020, during her re-election campaign, Mayor Anne Hidalgo emphasized the importance of making Paris a 15 minute city. Introduced by Carlos Moreno, in this concept residents are able to meet most of their needs within a short walk or bicycle ride from their homes [43]. Moreno proposed four guiding principles: ecology, proximity, solidarity and participation [44].

2.3 IoT solutions

The Internet of things (IoT) as a concept does not have a standard definition, but it can be described as the network that links *things* – electronic devices, components, etc. – that carry sensors/actuators and that are connected to the Internet.

Since IoT has not been standardized as of now, there is not a definite structure to implement it. Sethi and Sarangi's [8] research concluded that two different architectures can describe its behaviour, as depicted in figure 2.5 from [8].

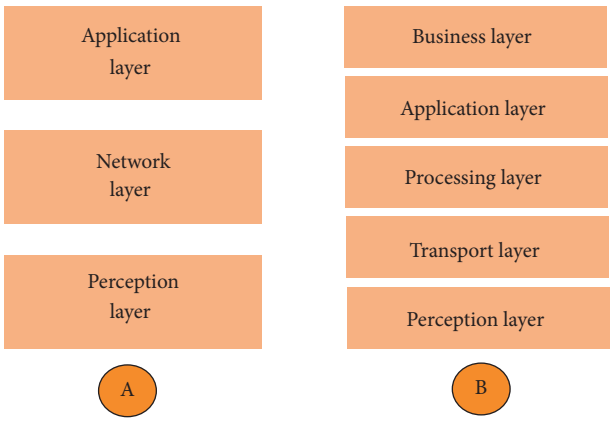


Figure 2.5: Architecture of IoT (A: three layers) (B: five layers) (extracted from [8])

The three-layer architecture consists of the following layers:

1. The perception layer is composed of sensors and gathers information about the environment;
2. The network layer connects the nodes and is responsible for the transmission of data;
3. The application layer delivers the data to the end user.

Sethi and Sarangi considered that the three-layer architecture lacked precision as technology evolved, so they proposed a new five-layer model, as seen in figure 2.5 b):

1. The transport layer acts as the bridge between the perception and processing layers, and can be implemented on top of protocols such as 3/4/5G or short (e.g., Bluetooth, Wi-Fi and ZigBee) to long (e.g., LoRaWan, Sigfox and NB-IoT) range wireless protocols;
2. The processing layer is responsible for storing, analysing and filtering the data from the sensors, resorting to databases or the cloud;
3. The business layer supervises the lower levels and assures user's security and privacy.

Mocnej et al. [9] proposed an unified protocol stack with five layers – physical, data link, network, transport and application – that synthesizes the most prevalent technologies used in the IoT, as depicted in figure 2.6 from [9].

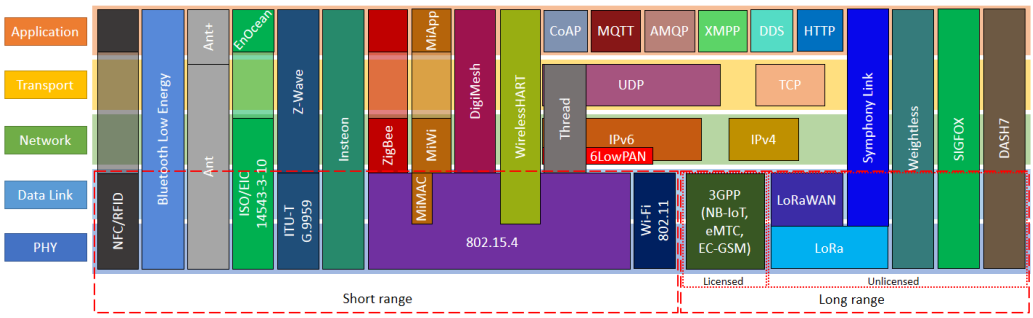


Figure 2.6: Protocol stack of IoT network technologies (extracted from [9])

Table 2.3 compares different wireless communication protocols, following the studies from Mocnej et al. [9], Al-Sarawi et al [45] and Lauridsen et al. [46]. The comparison criteria was:

- Frequency band – interval in the frequency domain;
- Range – communication distance covered by the technology;
- Data rate – maximum number of bits transferred per unit of time;

As seen in figure 2.6 and table 2.3, the IoT suffers from platform fragmentation, caused by the different technologies that can be used in the *things* network. The lack of standardization hinders the implementation of stable and secure ecosystems and makes difficult the interoperability of systems, but allows the user to select the most suitable and fitting solution to the desired application.

Table 2.3: IoT wireless communication protocols

Technology	Frequency Band	Range	Data rate
Bluetooth [9]	2.4 GHz	10 – 100 m	1 Mbps
Wi-Fi [47]	2.4 GHz (IEEE 802.11) 6 GHz (IEEE 802.11ax-2021)	10 m – 1 km	1 Mbit/s (IEEE 802.11) 9608 Mbit/s (IEEE 802.11ax-2021)
ZigBee [9]	2.4 GHz	10 – 100 m	250 kbps
LoRaWan [9]	868 MHz (Europe)	10 km+	27 kbps
Sigfox [45]	868 MHz (Europe)	10 – 50 km	100 bps (Uplink) 600 bps (Downlink)

2.4 Data processing and storage

The data gathered by information systems, such as the IoT, can be processed recurring to two distinct paradigms: edge and cloud computing. The difference between them is determined by the location where workloads are run.

Edge computing

The edge topology moves the computation, storage and networking closer to the source of data, decentralizing the system's workload. This concept was introduced as a solution to the network bandwidth overload and latency created by the increase of IoT devices.

Cloud computing

Cloud services are provided by a centralized system consisting of data centres available on the Internet. It can be used to store, process, compute and analyse massive amounts of data, without expensive hardware at the location where it is obtained.

Comparison

The edge topology architecture is inherently more private and secure, due to its distributed approach. In terms of scalability, the variety of devices in an edge configuration makes flexibility challenging to achieve.

Fog computing

There are systems that require features from both edge and cloud computing. To answer this, the fog paradigm was introduced. It compensates the lack of processing capacity of edge devices and reduces bandwidth problems associated with the cloud.

2.5 Databases, server and hosting

Databases can be hosted using two approaches: locally or by third party services responsible for hosting and maintaining the database. For this purpose, a survey was conducted of some of the most used solutions on the market, including IoT specific frameworks. Table 2.4 shows the

comparison between different hosting services that could be used to support the database. These comparison includes:

- The software used to implement the database (DBMS - database management system);
- The free-plan server capacity (either storage or write/read capacity);
- Pros and cons of the solution.

Table 2.4: Comparison between different hosting services

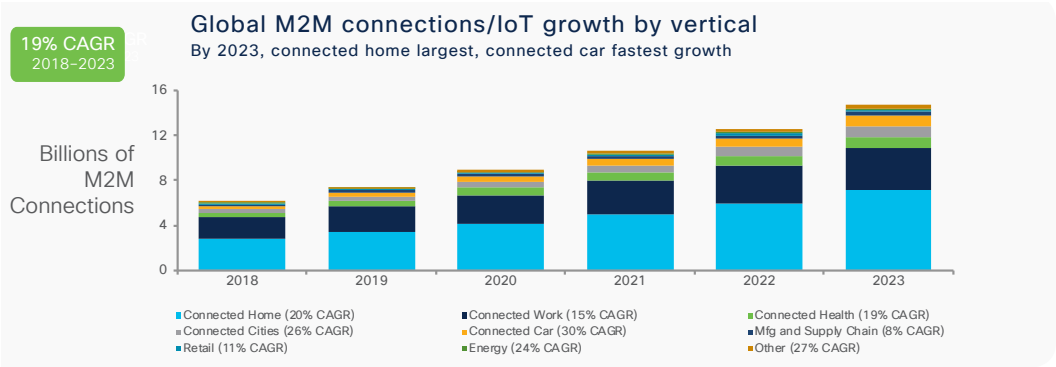
Hosting Service	DBMS	Capacity (free plan)	Pros	Cons
Heroku	PostgresSQL	512MB 10k rows	Polyglot platform (Java, Node.js, Scala, Clojure, Python, PHP, Go, etc.)	Has a limited free tier
Amazon RDS	PostgreSQL, mariaDB, MySQL, Amazon Aura, Oracle and Microsoft SQL Server	20GB 12 months	Supports multiple languages	Has a limited free use time
MongoDB Atlas	NoSQL	512MB	Fast Deployment Strong Community	Less flexibility in queries Paywalls in query functions
Google Cloud Firestore	NoSQL	1GB storage (50k reads and 20k writes)	Strong Community Large capacity for free	Limited querying capabilities
Redis Labs	NoSQL	30 MB	Ranked the 4th NoSQL database in user satisfaction and market presence (reliable)	Limited free plan capacity
FEUP's Database Server	PostgresSQL or MariaDB	–	No limit Free (for students)	No server downtime guarantees

2.6 Conclusion

This chapter started by introducing different Pavement Energy Harvesting technologies. Based on the classification presented, Pavnext's system can be placed in the category of electromagnetic generation – a hydraulic system with mechanical actuation – one of the technologies with the biggest development in the past years.

Then, the increasing interest in the smart city concept in modern urban areas was covered, with focus on traffic monitoring systems that help improving safety and reduce traffic congestion. Other than that, the amount of data that can be useful to cities' administrations is also an emergent topic in today's *age of information*, creating a huge opportunity to projects that know how to conciliate innovation and efficiency, attracting investment.

IoT applications have seen an increase usage for the past years. Cisco's Annual Internet Report (2018–2023)[4] estimates that by the end of 2021, 10.6 billions of IoT connections will be registered. Home applications represent the vast majority of use cases – home automation, home security and video surveillance. Figure 2.7 from 4 depicts the expected growth of IoT connections by 2023.



Source: Cisco Annual Internet Report, 2018–2023

Figure 2.7: Global M2M connection growth by industries (extracted from 4)

[4] Cisco's Annual Internet Report (2018–2023)

Chapter 3

Development of the monitoring and communication system

3.1 Introduction

Now that Pavnext's solution has been contextualised in the scope of smart cities, the project's specifications and implementation will be described. Pavnext's system can be applied in urban environments or on motorways[1]:

- Before crosswalks, roundabouts or traffic light signs: helps reducing vehicle's speed and produces energy that can be used in road illumination;
- Near schools and hospitals: the system can reduce vehicles' speed in roads close to schools or hospitals, which usually have a reinforced need of moving at low speeds;
- Residential areas with high flow of pedestrians and cyclists;
- Before tolls and on motorways.

The system is deployed in the pavement as shown in figure 3.1.

The structure represented in figure 3.1 is composed of several modules similar to the one that is depicted in figure 3.2 from Pavnext's website[1].

The harvesting system implemented is based on a hydraulic system with mechanical actuation. The prototype's mechanical system is depicted in figure 3.3 from [10]. The most recent prototype, where the tests were conducted, has $1.20 \times 0.35 \times 0.35$ m dimensions.

3.2 Requirements and specification

The monitoring and communication system depicted in figure 3.4 is composed of the following subsystems:

[1] Pavnext's website

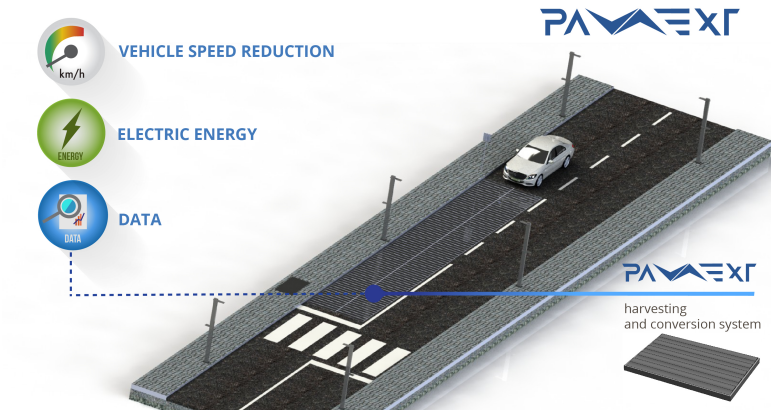


Figure 3.1: Pavnext's system

Figure 3.2: Pavnext's prototype alpha (extracted from Pavnext's website[1])

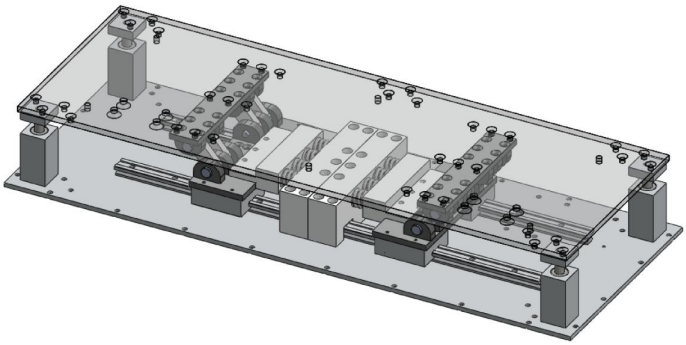


Figure 3.3: Pavnext's prototype mechanical system (extracted from [10])

- Each module has a *slave* microcontroller responsible for acquiring the data described in section 3.2.1;
- Multiple *slaves* report the data to a *local-master* responsible for a set of modules;
- The *local-masters* send the information to a *master*;
- The *master* communicates wirelessly via LoRa with a gateway.

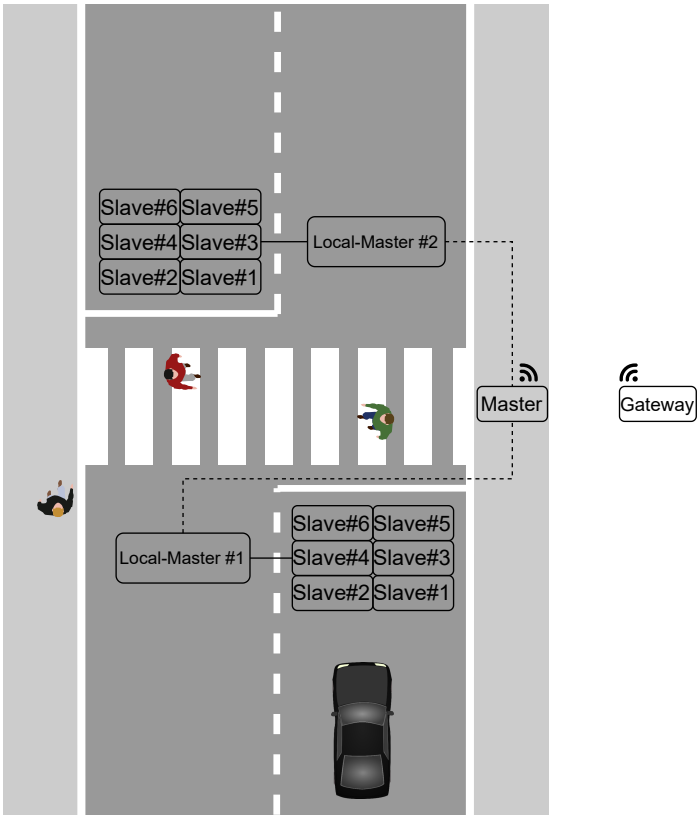


Figure 3.4: System's architecture

3.2.1 Data to be monitored

The main purpose of the product is to reduce the speed of the vehicles and generate energy. However, given the fact that this project is within the scope of the smart cities concept, it is important to generate valuable data for its administration and citizens.

This project will take advantage of the infrastructure already implemented on the pavement and collect valuable data that can be divided into two categories: functional and environmental.

3.2.1.1 Functional

In the age of information that we live in nowadays, data is key. Being able to maximize the amount of knowledge gathered by the product is important to end up with a diversified solution. Taking this into consideration, the objective is to measure the following parameters:

- Average power and total energy generated by a vehicle;
- Vehicles' speed;
- Maximum acceleration in three-axis, to verify the stability of the modules.

3.2.1.2 Environmental

Temperature and relative humidity sensors were included in order to evaluate the conditions inside the modules. This can help preventing malfunctions in the microcontrollers and the rest of electronic equipment, due to system failures or bad weather conditions.

3.2.2 Communication protocols

In order to avoid congestion and collisions in the network, the *slaves* – *local-master* and the *local-master*– *master* subsystems communicate via a specific protocol. At the same time, by fragmenting the communication modes, network failures will not impact the rest of the implementation. Only serial protocols were considered, because the number of wires is lower when implementing the physical layer, as depicted in figure 3.5.

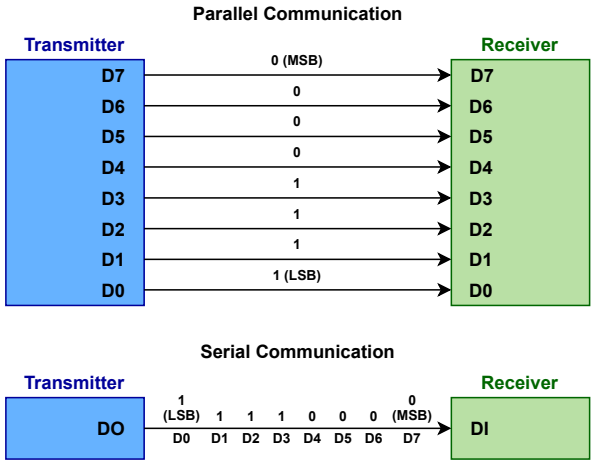


Figure 3.5: Parallel and Serial Communication

Further comparison between the two types of communication:

- Sharing a single communication link makes the serial option slower and more prone to interference between transmissions if the bit spacing is not properly configured;
- Increased possibility of crosstalk between parallel's multiple channels;
- Serial's sequential transmission is more suited for high frequency and long distance applications (due to noise and interference);
- A single communication link is simpler to assemble and maintain (less cables);

Table 3.1 presents various embedded systems' serial communication protocols and characteristics that suit different applications. Most of the serial protocols applied nowadays are implemented using a differential pair – two wires are used, carrying the same signal but with opposite polarity – the receiver decrypts it by doing the difference between the two lines. This method is more robust

than a single wire because interference modifies both signals in a similar way, but the difference between them remains the same [48].

Table 3.1: Communication protocols

Protocol	Type	Transfer Mode	Number of Devices	Max Distance (m)	Max Data Rate (bps)	Number of Signal Lines
RS-485	Asynchronous Serial	Half-Duplex	32	1200	10M	2
I^2C	Synchronous Serial	Half-Duplex	128 (7-bit address)	5	3.4M	2 + Common GND
SPI	Synchronous Serial	Full-Duplex	No physical limit	3	60M	3 Common + 1 *slave* select per node
CAN FD	Asynchronous Serial	Half-Duplex	128 (7-bit address) ∼110 physical limit	5000	1M	2

The decision to use two different communication modes for the subsystems resulted from the fact that sharing a communication channel would mean that a problem in the line would be hard to detect and compromise the whole installation.

The maximum distance of I^2C and SPI does not satisfy the needs of the application (maximum distance between the furthest *local-master* and the *master* of 20 m), so only RS-485 and CAN FD were considered and the choices are described in table 3.2.

Table 3.2: Subsystem's communication protocols

Subsystem	Protocol
slave – local-master	RS-485
local-master – master	CAN FD

The system's communication configuration can be visualized in figure 3.6.

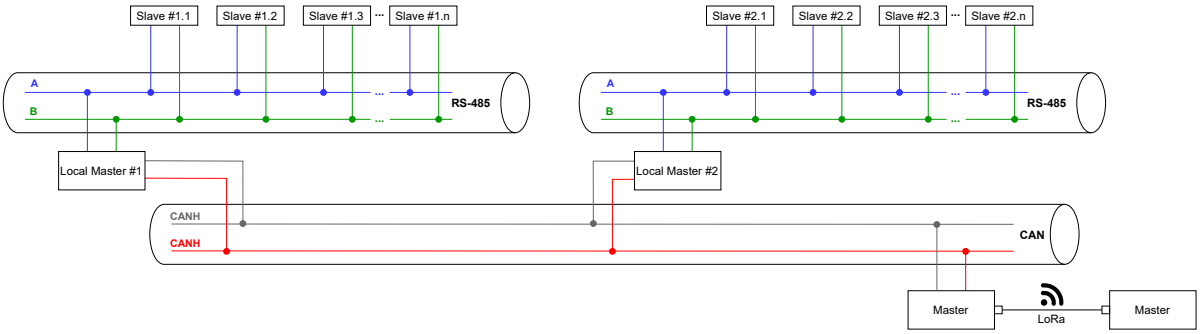


Figure 3.6: System's communication configuration

3.3 Conception and design

3.3.1 *Slave* system

The *slave* system is responsible for acquiring the data generated by a vehicle passage and transmitting it to its *local-master*.

3.3.1.1 Sampling period

The bottleneck of the system is the estimation of the energy generated. It is important to acquire data as fast as possible to obtain an estimation that resembles the generated values. It was determined that a period of 1 ms is the minimum to ensure a realistic estimation of the energy generated. The accelerometer is also sampled at this frequency. The temperature and relative humidity are read every ten vehicle detections.

3.3.1.2 Hardware

The *slave* system requirements are the following:

- Four Analog-to-Digital converter channels capable of satisfying the imposed sampling period;
- The selected communication protocol for the *slave-local-master* system was RS-485, so an USART interface is required;
- I^2C to communicate with different sensors (accelerometer and temperature and relative humidity);
- Initial snippets determined that at least 16 KB of program memory and 1 KB of RAM were necessary to run the fully fledged algorithms.

Table 3.3 lists microcontrollers that fit the criteria mentioned above and are on the same price range. Only 8-bit architecture MCUs were considered, due to their lower price and characteristics that fit the desired application.

Table 3.3: *Slave* microcontroller options [a]

Microcontroller	ADC	Flash Memory	RAM	Pins
ATmega324	8-Channel 10-Bit	32 Kbyte	2 Kbyte	44
ATmega328	8-Channel 10-Bit	32 Kbyte	2 Kbyte	32

[a] The list of microcontrollers available on the market was obtained using Mouser Electronics' catalogue.

The specifications of both solutions are identical and its price is the deciding factor. The ATmega328 microcontroller was selected due its lower price and more familiarity (e.g., Arduino UNO Rev3).

RS-485 communication

The communication between the *slave*s – *local-master* subsystems could be achieved using the USART interfaces provided by each system's microcontroller, but a multiplexing technique would need to be implemented. Having this in consideration, RS-485 transceiver was added, converting the USART signal to a differential one, enabling data transmission over long distances and allowing multiple devices to be connected to the bus.

Table 3.4 compares RS-485 transceivers. The MAX485 was selected because it is considered the *go-to* transceiver in different applications, such as USART to RS-485 converters, making it the better documented and tested option.

Table 3.4: RS-485 transceivers

Transceiver	Supply Voltage
SP485E MaxLinear	4.75 to 5.25 V
ISL81487 Renesas	4.5 to 5.5 V
SN75ALS1 Texas Instruments	4.75 to 5.25 V
MAX485 Maxim Integrated	4.75 to 5.25 V

Sensors

An analysis of the sensors needed was performed. Table 3.5 lists temperature and relative humidity sensors and their specifications.

Table 3.5: Temperature Sensors

Sensor	Temperature Range	Humidity Range	Operating Voltage	Communication
DHT11 Aosong	0 to 50 °C ± 2 °C accuracy	20 to 80 % ±5 % accuracy	3 to 5 V	1 Data Pin
DHT22 Aosong	−40 to 80 °C ± 0.5 °C accuracy	20 to 80 % 2 to 5 % accuracy	3 to 5 V	1 Data Pin
AM2320 Aosong	−40 to 80 °C ± 0.5 °C accuracy	0 to 99.9 % ± 3 % accuracy	3.3 to 5 V	I^2C
AHT20 Aosong	−40 to 85 °C ± 1 °C accuracy	0 to 100 % ± 3 % accuracy	3.3 to 5 V	I^2C
Si7021 Silicon Labs	−10 to 85 °C ± 0.4 °C accuracy	0 to 80 % ± 3 % accuracy	3.3 to 5 V	I^2C
HTS221 STMicroelectronics	−40 to 120 °C ± 0.5 °C accuracy	0 to 100 % ± 3.5 % accuracy	3.3 to 5 V	I^2C
SHTC3 Sensirion	−40 to 125 °C ± 0.2 °C accuracy	0 to 100 % ± 2 % accuracy	3.3 to 5 V	I^2C

After consulting table 3.5, the SHTC3 is the best option, due to its better range/accuracy and lower price compared to the HTS221, which was used because it was available in the laboratory, ready to be tested and implemented. Adafruit's breakout board, which includes the HTS221 and a power regulator is depicted in figure 3.7 from Adafruit's product overview[2].

Accelerometers measure both static (e.g. gravity) and dynamic (e.g. sudden starts/stops) acceleration. The main objective of the accelerometer added to the prototype is to detect passing

[2] https://www.adafruit.com/product/4535

Figure 3.7: Adafruit's HTS221 breakout board

vehicles. In order to do so, the algorithm makes use of interrupt functions included by accelerometer manufacturers, activated when the acceleration on an axis exceeds a certain level [49]. So, this interrupt capability is a requirement to take into consideration when choosing the accelerometer.

Other options, such as gyroscopes, which measure angular velocity, and magnetometers measuring magnetic fields can be used to help determine the stability of the installation (this method would need to be tested due to the magnetic properties of the modules). Inertial measurement units (IMUs) combine accelerometers, gyroscopes and magnetometers, providing two to nine degrees of freedom. Sensor fusion algorithms can make use of the raw data provided by IMUs to estimate orientation and position over time. Although interesting, this methods were not considered because of their high computational demands.

Table 3.6 lists accelerometers by price and details their specifications.

Table 3.6: Accelerometers

Sensor	Measurement Range	Interrupts	Sampling Rate (max)	Operating Voltage	Communication
LIS3DHST Microelectronics (3-axis accelerometer)	± 2 / ± 4 / ± 8 / ± 16 g	2	5.3 kHz	1.71 to 3.3 V	I^2C SPI
ADXL337 Analog Devices (3-axis accelerometer)	± 3 g	None	1.6 kHz (X and Y axis) 550 Hz (Z axis)	1.8 to 3.6 V	3 Analog Outputs (one for each axis)
MMA8452Q NXP Semiconductors (3-axis accelerometer)	± 2 / ± 4 / ± 8 g	1	800 Hz	1.95 to 3.6 V	I^2C
LSM6DSO STMicroelectronics (3-axis accelerometer and gyroscope)	± 2 / ± 4 / ± 8 / ± 16 g ± 125 / ± 250 / ± 500 ± 1000 / ± 2000 dps [b]	2	6.6 kHz	1.71 to 3.6 V	I^2C SPI
H3LIS331DL STMicroelectronics (3-axis accelerometer)	± 100 / ± 200 / ± 400 g	2	1 kHz	2.16 to 3.6 V	I^2C SPI
LIS331HH STMicroelectronics (3-axis accelerometer)	± 6 / ± 12 / ± 24 g	2	1 kHz	2.16 to 3.6 V	I^2C SPI
KX132Kionix (3-axis accelerometer	± 2 / ± 4 / ± 8 / ± 16 g	2	25.6 kHz	1.7 to 3.6 V	I^2C SPI
ADXL345Analog Devices (3-axis accelerometer)	± 2 / ± 4 / ± 8 / ± 16 g	2	3.2 kHz	2 to 3.6 V	I^2C SPI

[a] The price of the sensors was obtained from Sparkfun's shop.
[b] dgs = degree per second.

Looking at table 3.6, most of the accelerometers listed meet the requirements – 1 kHz sampling rate and one interrupt. Adafruit's ADXL345 breakout board with a voltage regulator and logic-level shifting was selected and is depicted in figure 3.8 from Adafruit's product overview[3].

[3] https://www.adafruit.com/product/1231

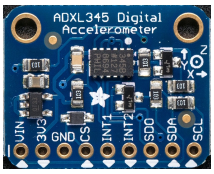


Figure 3.8: Adafruit's ADXL345 breakout board

Figure 3.9 depicts the *slave* board that was designed by me in the Altium Designer software and used in the tests conducted.

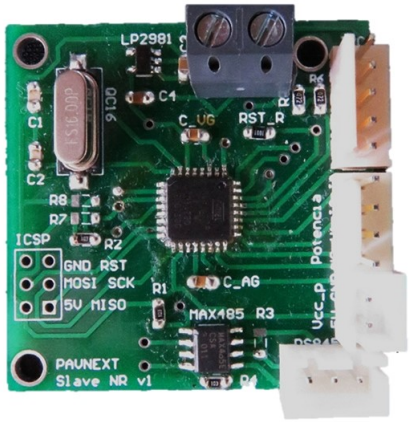


Figure 3.9: *Slave's* PCB

3.3.1.3 Software

The operating principle of the *slave* system can be described using the activity diagram depicted in figure 3.10.

When powered-up, the *slave* calls initialization routines for the USART and to verify the different sensors integrity by pinging them and checking the answer. If they are compromised, the *slave* notifies the *local-master* of malfunctions and powers off, waiting to be repaired.

If the initialization succeeds, the *slave* is put in a state waiting for a triggering event – a vehicle causes the z-axis acceleration of the module to reach a certain threshold. If the *local-master* signalled previously the *slave* that it can communicate, it starts acquiring the data related with the accelerometer and the generation of energy, namely the voltage and current produced by the generator.

When the *slave* stops collecting the data, it notifies the *local-master* that a vehicle was detected. Then, if the *local-master* answered with permission for a packet transmission, the *slave* sends it and waits for a new vehicle detection.

The data acquisition is described more detailed in the diagram depicted in figure 3.11.

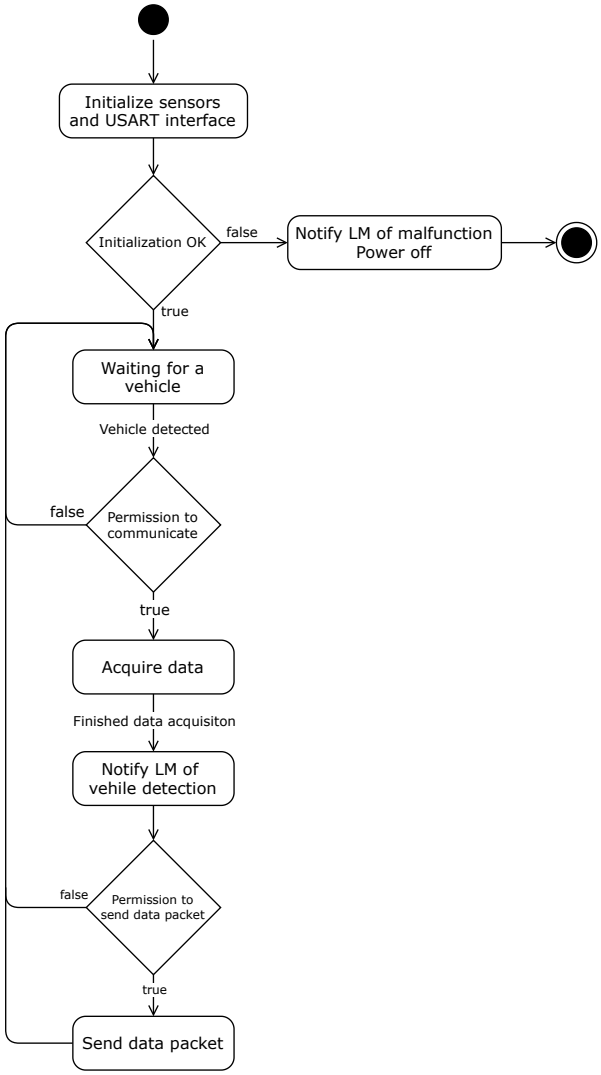


Figure 3.10: *Slave* UML activity diagram

Pavnext developed a board that outputs the values of voltage and the output voltage proportional to the AC current measured by a current sensor (Allegro ACS723) from each generator. Equation 3.1 presents the relation between the measured current and the output voltage.

$$I = 6.1597 \cdot V - 15.495 \tag{3.1}$$

The algorithm developed uses the four ADC channels to the read the voltage outputs of the board and estimates the energy generated.

Every millisecond after the accelerometer's interrupt, the ADC channels and the acceleration of the 3-axis are read. The acceleration values are compared with the previous readings to determine if they are new maximums. This parameter is important to detect damages to the installation caused by the vehicle.

The estimation of the energy generated is determined by calculating the instantaneous power

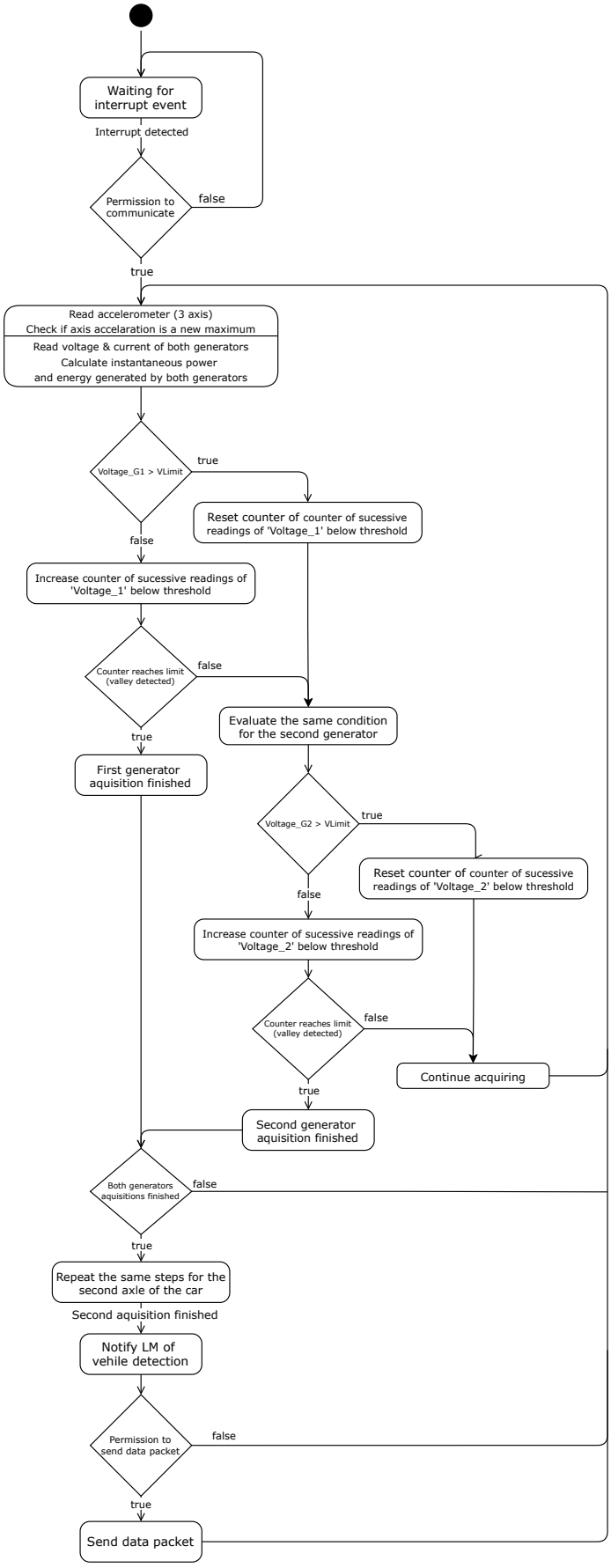


Figure 3.11: Detailed *Slave* UML activity diagram

(equation 3.2) and multiplying it by the interval between readings (around 1 ms) (equation 3.3).

$$P = VI \tag{3.2}$$

$$E = P \cdot \Delta t \tag{3.3}$$

The vehicle's speed is estimated by dividing the average distance between axles (also known as wheelbase) – 2.73 m – by the time elapsed between the two triggering events (equation 3.4).

$$\text{speed} = \frac{2.73\,\text{m}}{\Delta t} \tag{3.4}$$

Data packet

The packet that is sent to the *local-master* consists of ten data bytes and four control bytes, as depicted in figure 3.12.

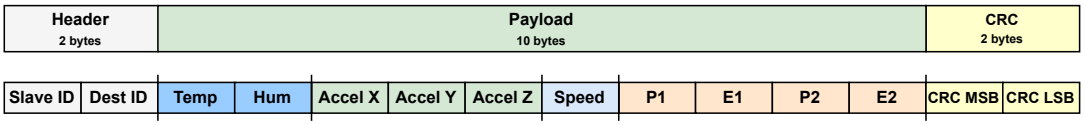


Figure 3.12: Data packet

The first two bytes identify the sender and the receiver, allowing the *local-master* to determine if the packet it has received is addressed to itself and from the correct slave.

The following ten bytes are the packet's payload, containing the data acquired. In order to fit the different values in a byte's 0 to 255 range, the following strategy was followed:

- The temperature sensor can measure in the range of $-40\,^{\circ}\text{C}$ to $120\,^{\circ}\text{C}$, so an offset of 100 is added;
- The relative humidity value is multiplied by 2 to increase accuracy;
- Tests using the accelerometer showed that the maximum value returned is around $40\,\text{m}\,\text{s}^{-2}$, so it is multiplied by 5 to increase accuracy;
- The vehicles' speed is calculated in m/s. 25 m/s – a relative high speed for the project's use cases – corresponds to 90 km/h, so the calculated speed can be multiplied by 100 to increase accuracy;
- The average power is multiplied by 5 to increase accuracy (during tests it was around 25 W, so even if its value doubles, it still fits in a byte without overflow);
- The estimation of the energy generated is around single digit values, so it can be multiplied by 100 to increase accuracy.

The integrity of the message is guaranteed by the CRC-16 present in the last two bytes.

The source code was documented using Doxygen and can be visited in the thesis' website documentation section[4].

3.3.2 *Local-master* system

The *local-master* acts as a bridge between the *slave*s and the *master*. It is responsible for six *slave*s, managing the flow of communication to avoid collisions and loss of information.

3.3.2.1 Hardware

The *local-master* system needs are the following:

- USART interface (RS-485 communication with the *slaves*);
- CAN controller (communication with the *master*).

Losing a *slave* due to malfunctions is not critical to the operation of the system, because there are at least five others that are acquiring data. However, that is not the case for the *local-master*, because without it the *slave*s can not communicate with the *master*. So, considering the fact that the *local-master* is important to the project's reliability, a development board was used instead of a microcontroller. A development board is equipped with all the necessary components, such as LEDs and IO pins, and was tested before being released to the public. But the main concern is the power system, which is included in development boards, sometimes with more than one regulator, helpful when working with devices that support different voltage levels.

The STM32G4 Nucleo-32 board – NUCLEO-G431KB – was used, because there were a few immediately available in the laboratory. The board is depicted in figure 3.13 from NUCLEO-G431KB's product overview[5] and its specifications are detailed in table 3.7.

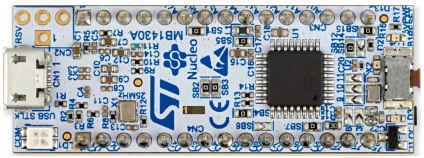


Figure 3.13: NUCLEO-G431KB (extracted from NUCLEO-G431KB's product overview[5])

CAN communication

The CAN node is constituted by a controller and a transceiver. The transceiver is the interface between the controller and the CAN bus, converting the single-ended logic used by the controller to the differential signal transmitted over the bus, as depicted in figure 3.14. Most of the MCUs

[4] https://web.fe.up.pt/ up201603192/thesis/doxygen
[5] https://www.st.com/en/evaluation-tools/nucleo-g431kb.html

Table 3.7: NUCLEO-G431KB specifications

Microcontroller	**Crystal oscillator**	**Flash Memory**	**RAM**	**Communication interfaces**
STM32G431KB (up to 170 MHz)	24 MHz (External)	128 Kbyte	32 Kbyte	CAN FD I^2C USART SPI

and development boards – as is the case for the NUCLEO-G431KB – only include the controller, so a transceiver needs to be added to the design.

Table 3.8 compares CAN FD transceivers. Their specifications are very similar, but the MCP2561 was selected because it was better documented and more widely used.

Table 3.8: CAN FD transceivers

Transceiver	**Supply Voltage**
ATA6560 Microchip	4.5 to 5.5 V
NCV7344 ON Semiconductor	4.75 to 5.25 V
MCP2561/2FD Microchip	4.5 to 5.5 V

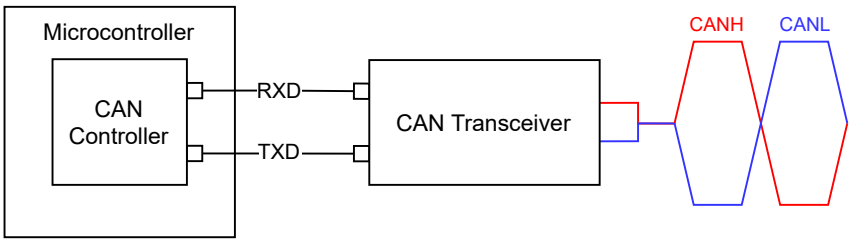


Figure 3.14: CAN communication

In case the communication between the *local-master* and the *master* is compromised, the algorithm stores the packets sent by the *slaves*. The storage capacity was doubled by adding Microchip's 23K256 – 32 Kbytes SPI Serial SRAM.

Considering that each *local-master* is responsible for six *slaves* and that a *slave* sends a data packet with a size of 14 bytes, each vehicle results in:

$$6\ \textit{slaves} \times 14\ \text{bytes} \quad = \quad 84\ \text{bytes} \tag{3.5}$$

With the extra RAM provided, the *local-master* has around 64 Kbytes available to store packets. It can store information about 761 vehicles passing:

$$\frac{64\ \text{Kbytes}}{84\ \text{Kbytes per vehicle}} \quad = \quad 761.90 \approx 761\ \text{vehicles} \tag{3.6}$$

Depending on the frequency of the vehicles, the *local-master* storage capacity can vary from 25 minutes in heavy traffic situations to more than a week if vehicles rarely activate the technology. Table 3.9 shows for how long the *local-master* can store packets without losing information.

Table 3.9: Estimation of the *local-master* memory capacity

Assuming a vehicle every **5 minutes**: **63:29:31**
Assuming a vehicle every **minute**: **12:41:54**
Assuming a vehicle every **half-minute**: **6:20:57**
Assuming a vehicle every **10 seconds**: **2:06:59**
Assuming a vehicle every **5 seconds**: **0:25:24**

Figure 3.15 depicts the *local-master* board that was designed by me in the Altium Designer software and used in the tests conducted.

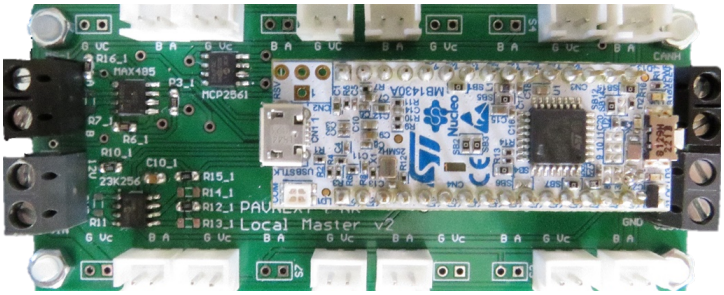


Figure 3.15: *Local-master's* PCB

3.3.2.2 Software

The algorithm implemented in the *local-master* can be described by the diagram in figure 3.16.

The *local-master* starts by initializing the communication interfaces (CAN FD and USART). If problems are detected, the *local-master* tries to notify the *master* and powers off.

If the initialization occurs without detecting errors, the *local-master* waits for the *master* to give it permission to communicate. When the authorisation is granted, the *local-master* signals the *slaves* that they can communicate as well.

Then, when the *slaves* start reporting vehicles activity, the *local-master* performs a loop requesting the data packets. If the CRC present in the packet is the same as the one calculated by the *local-master*, the packet is stored. After all the *slaves* send their packets, the *local-master* forwards them to the *master*.

The source code was documented using Doxygen[4].

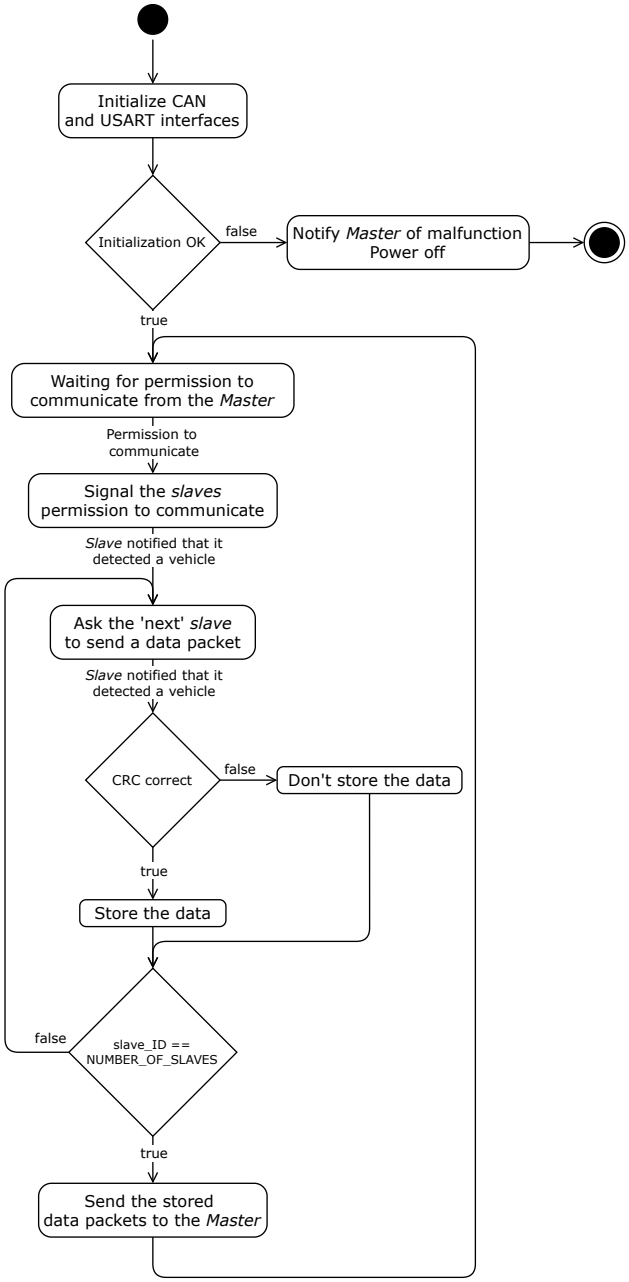


Figure 3.16: *Local-master* UML activity diagram

Slaves–local-master communication

The communication between the *slave*s and the *local-master* is described in figure 3.17.

1. The *local-master* tells the *slave*s that it is accepting communication.

2. *Slave* #1 is waiting for a car. When it arrives, the *local-master* is notified.

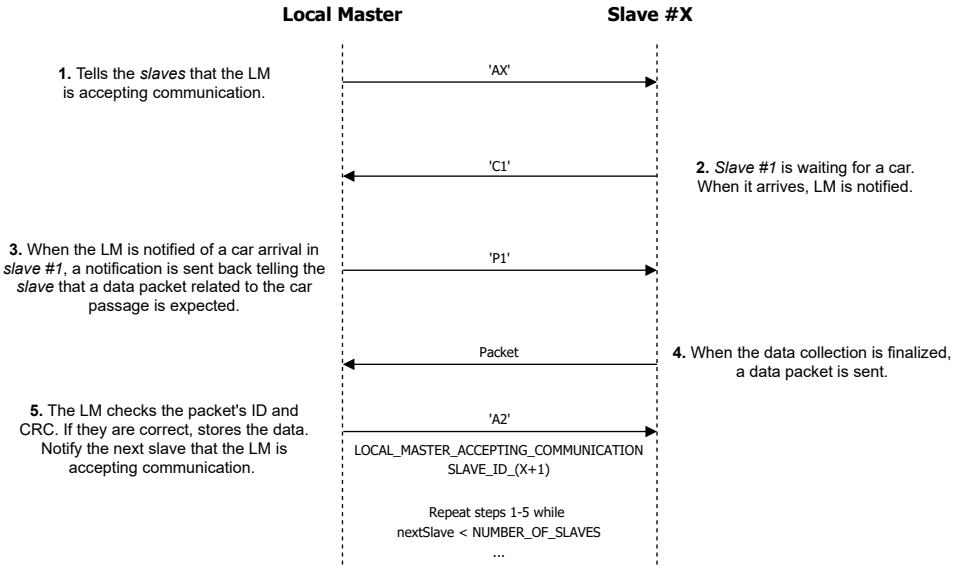


Figure 3.17: *Slaves – local-master* communication

3. When the *local-master* is notified of a car arrival in *slave* #1, a notification is sent back telling the *slave* that a data packet related to the car passage is expected.
4. When the data collection is finalized, a data packet is sent.
5. The *local-master* checks the packet's ID and CRC. If they are correct, stores the data.
6. Notify the next slave that the *local-master* is accepting communication.
7. Repeat steps 1-5 for the next five *slave*s.

3.3.3 *Master* system

The *master* is responsible for supervising the system, receiving – via CAN – the packets sent by the *local-master*s and sending them via LoRa to the gateway.

3.3.3.1 Hardware

The *master* system requirements are the following:

- CAN FD controller (communication with the *local-master*);
- LoRa communication.

LoRa transceiver

Development boards with LoRa interface are not widely available on the market and their cost is not competitive enough to justify the investment in this application. Considering this,

LoRa transceivers were the option chosen to implement the requirement. Table 3.10 lists LoRa transceivers by price and details their specifications.

Table 3.10: LoRa transceiver

Transceiver	Modulation	Frequency	Sensivity	Maximum Transmit Power	Range
HopeRF RFM95/96/97/98	(G)FSK, (G)MSK, LoRa and OOK	433 / 470 / 868 / 915 MHz	-148 dBm	+20 dBm	No data
Microchip RN2483	(G)FSK and LoRa	434 / 868 MHz	-146 dBm	+14 dBm	15 km (suburban) 5 km (urban)
Semtech SX1261/2	(G)FSK, (G)MSK and LoRa	434 / 490 / 868 / 915 MHz	-148 dBm	+22 dBm	+ 20 km
Semtech SX1280	(G)FSK, FLRC and LoRa	2.4 GHz	-132 dBm	+12.5 dBm	No data
Semtech SX1272	(G)FSK, (G)MSK, lora and OOK	868 / 915 MHz	-130 dBm	+20 dBm	16 km

[a] The list of LoRa transceivers available on the market was obtained using Mouser Electronics' catalogue.

All transceivers listed fulfil the requirements, although some have better features, such as a wider range of frequencies that can be used or a longer range. Ultimately, RF Solutions' LAMBDA62 with an integrated Semtech SX1262 was used, because there were some modules available to be used in the laboratory. The chip is depicted in figure 3.18.

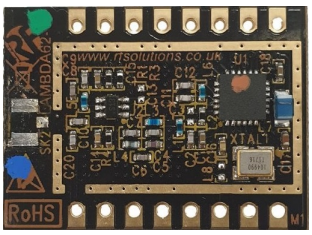


Figure 3.18: RF Solutions' LAMBDA62

The requirements of the *master* are not substantially different from *local-master*'s. The LoRa module needs SPI to be programmed, configured and controlled, but since the *local-master*'s development board has SPI interfaces, the same board was selected (with the addition of the same SRAM chip).

Figure 3.19 depicts the *master* board that was designed by me in the Altium Designer software and used in the tests conducted.

3.3.3.2 Software

The algorithm followed by the *master* is similar to the one implemented by the *local-master* and can be described by the diagram in figure 3.20.

The communication interfaces (CAN FD and LoRa) are initialized and the *local-master* is notified that it can communicate if the initialization is successful.

When all the *slave*s finish sending packets, the *local-master* signals the *master* that it has packets ready to be sent. The *master* answers, granting permission for the packets to be sent. If

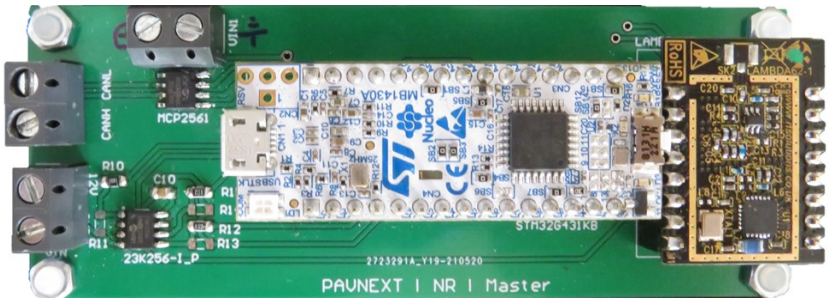


Figure 3.19: *Master's* PCB

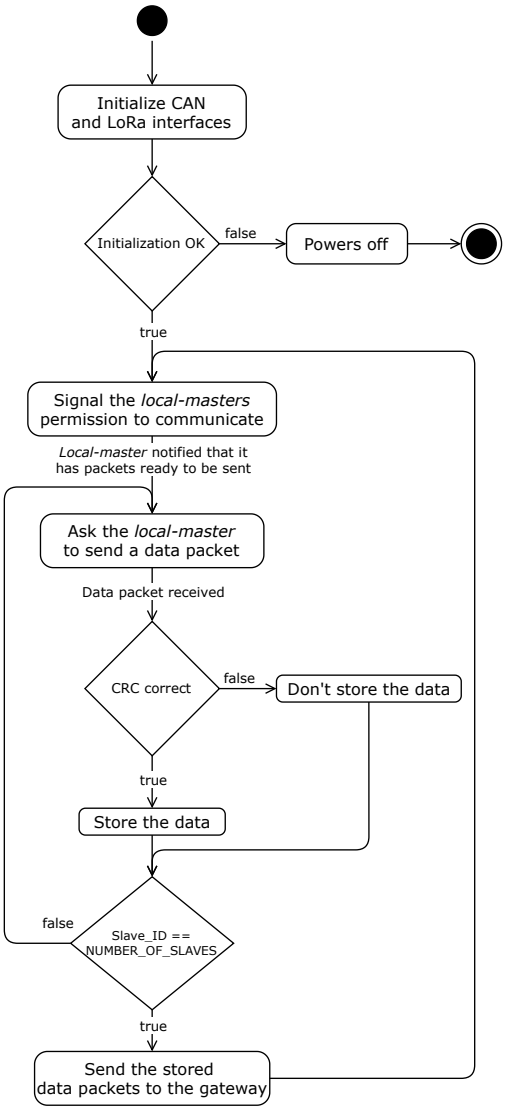


Figure 3.20: *Master* UML activity diagram

the calculated CRC matches the packet's, the packet is stored. When every packet is received, the stored ones are sent to the gateway.

The source code was documented using Doxygen[4].

3.3.4 Gateway

The gateway is a passive component of the system. Its purpose is to receive the data sent by the *master* via LoRa. The information contained in the packets is processed and decrypted, reverting the *slave*'s strategy of fitting the data into a byte.

3.3.4.1 Hardware

The gateway system needs are the following:

- LoRa communication;
- Wi-Fi connection (upload the data to the database).

Initially, development boards with both LoRa and Wi-Fi interfaces were considered, but were discarded due to their expensive cost. Then, the addition of a Wi-Fi module to the *local-master* board was contemplated. Table 3.11 lists Wi-Fi modules and their specifications.

Table 3.11: Wi-Fi modules

Module	Release year	Microcontroller	Clock Frequency
ESP8266	2014	Xtensa single-core 32-bit L106	80 MHz
ESP32	2016	Xtensa single/dual-core 32-bit LX6	160/240 MHz
ESP32-S2	2019	Xtensa single-core 32-bit LX7	240 MHz
ESP32-S3	2020	Xtensa dual-core 32-bit LX7	240 MHz

The ESP8266, although outdated when compared to the other options, fulfilled the requirements. However, this alternative implementation was not used, because there are boards that incorporate the Wi-Fi in their design, such as Espressif's development kits listed in table 3.12.

Table 3.12: Wi-Fi development kits

Kit	Microcontroller	Clock Frequency	Flash Memory	RAM	Additional Interfaces
ESP32-S2-DevKitM-1R	ESP32-S2-MINI-1-N4	80 MHz	4 MByte	320 KByte	I^2C SPI
ESP32-S2-Saola-1R	ESP32-S2-WROVER	160/240 MHz	4 MByte	320 KByte	I^2C SPI
ESP32-C3-DevKitM-1	ESP32-C3-MINI-1	240 MHz	4 MByte	400 KByte	I^2C SPI Bluetooth LE

The specifications of the kits are very similar, although the ESP32-C3-DevKitM-1 has slightly better ones and the addition of a Bluetooth interface. The selected board was the ESP32-S2-Saola-1R, because the SP32-C3-DevKitM-1 was out of stock. The board is depicted in figure 3.21 from ESP32-S2-Saola-1R's Mouser Electronics' product overview[6].

[6] https://pt.mouser.com/ProductDetail/Espressif-Systems/ESP32-S2-Saola-1R

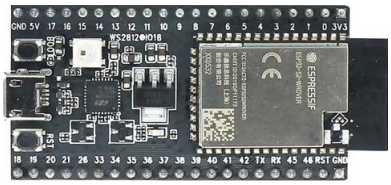


Figure 3.21: ESP32-S2-Saola-1R (extracted from ESP32-S2-Saola-1R's Mouser Electronics' product overview[6])

The LoRa communication was obtained based on the same approach used in the *master*, RF Solutions' LAMBDA62 transceiver.

3.3.4.2 Software

The algorithm implemented in the gateway can be described by the diagram in figure 3.22.

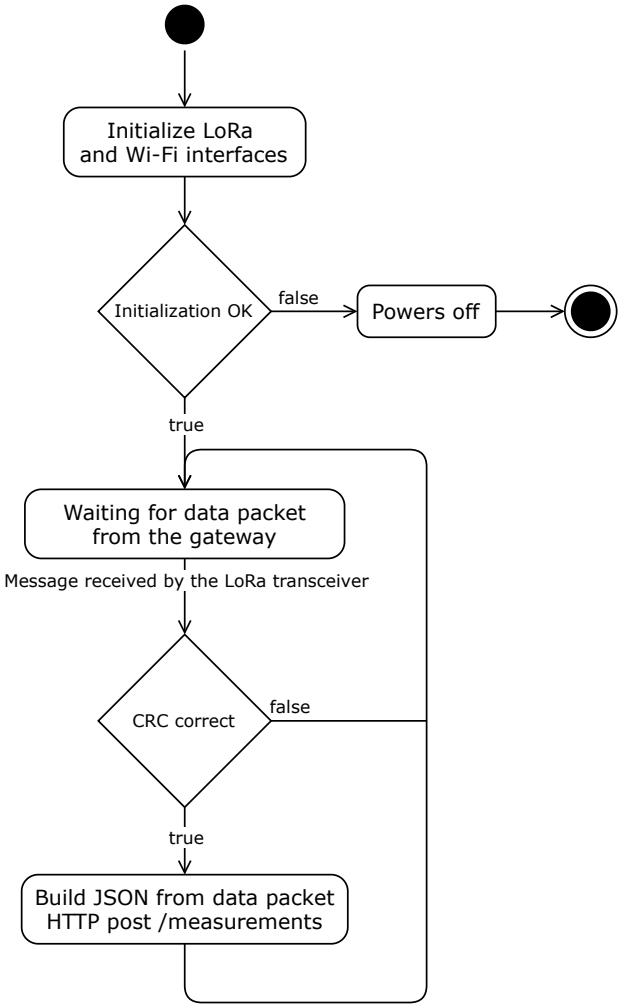


Figure 3.22: Gateway UML activity diagram

The gateway initializes its communication interfaces, then is put in a passive situation, waiting for packets from the *master*. Once the packets arrive, its CRC is evaluated and if it matches the calculated one, the data is processed, converted to JSON – the format expected by the REST APIs – and a HTTP post request is made, in order to upload the data to the database.

The source code was documented using Doxygen[4].

3.4 Data storage

The data storage subsystem connects the hardware and web application. Information flows in different directions in this bridge:

- The gateway sends measurements settings via HTTP to the database;
- The data presentation application exchanges information with the database by sending HTTP requests to the REST APIs. These APIs query data from the database using the Node.js server. This architecture is commonly called the PERN Stack.

3.4.1 Database

Table 2.4 compared different database hosting services. Initially, Amazon Relational Database Service (RDS) was considered the best option, because it could be easily integrated in later development stages, such as the back-end server service (AWS Elastic Beanstalk). This option was tested and validated, but its usage was stopped once the free plan limits were reached. The database is now being hosted in Heroku, a cloud platform.

The database subsystem can be divided into classes, as represented in the diagram in figure 3.23. Each class has its own attributes and operations and the relationships between them establish the structure of the system. In this case, this can be described as:

- The user owns one or more installations;
- Each installation produces data every time a vehicle activates the system.

3.4.2 Back-end server

Communication between the gateway and the web application is achieved by a back-end server developed using Node.js as the run-time environment and using the Express.js framework to handle HTTP requests, allowing the REST APIs to exchange information with the database in an architecture based on the PERN (PostgreSQL + Express + Node + React) stack.

The back-end server has been tested and validated using AWS Elastic Beanstalk has the hosting service. Once the free limit was reached, the application was transferred to the Google App Engine.

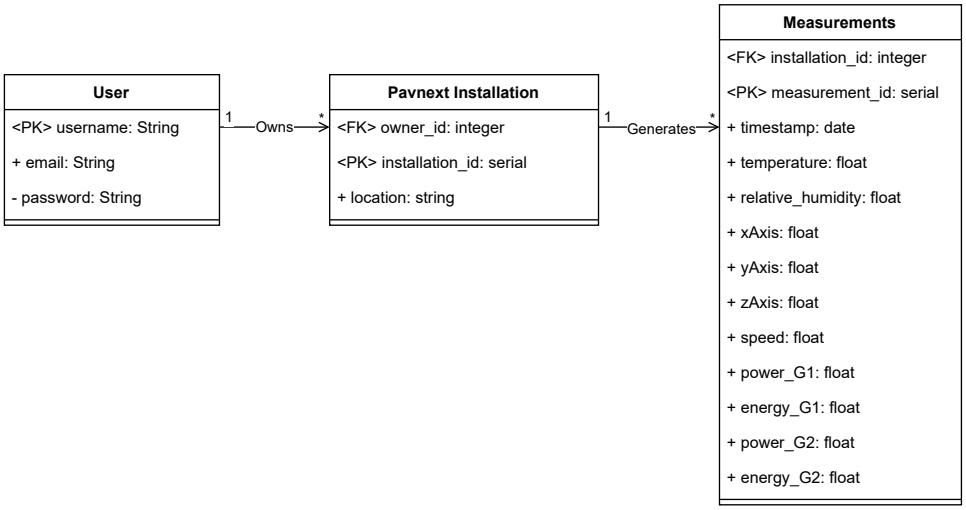


Figure 3.23: Database UML class diagram

3.5 Data presentation application

The web application was written in HTML, with JavaScript and PHP scripts that fetch the data from the database using the REST APIs. The charts were obtained using the Highcharts charting library.

The data presentation application is hosted in FEUP's personal webpage services and can be visited on the thesis' website[7]. A preview is depicted in figure 3.24.

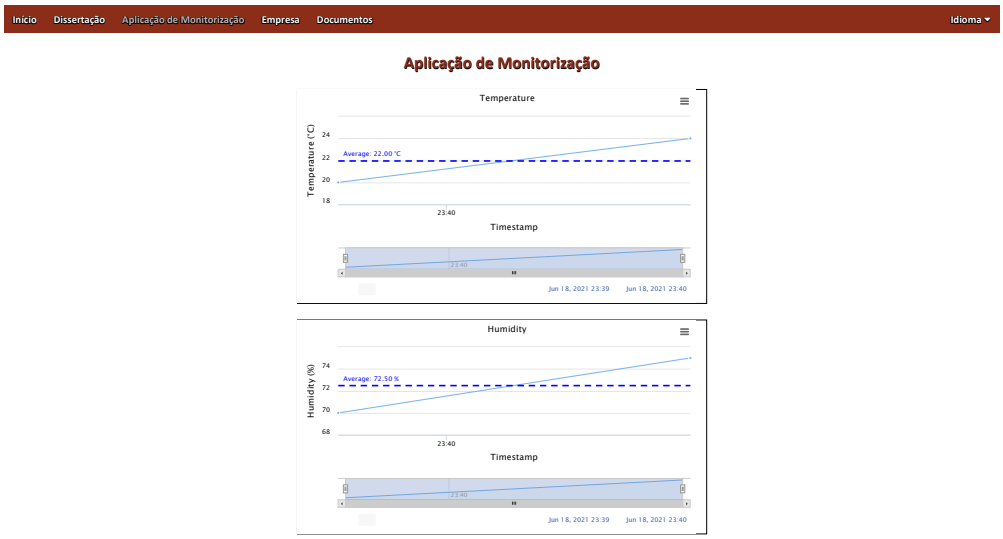


Figure 3.24: Data presentation application

[7] https://web.fe.up.pt/ up201603192/thesis/app/app.html

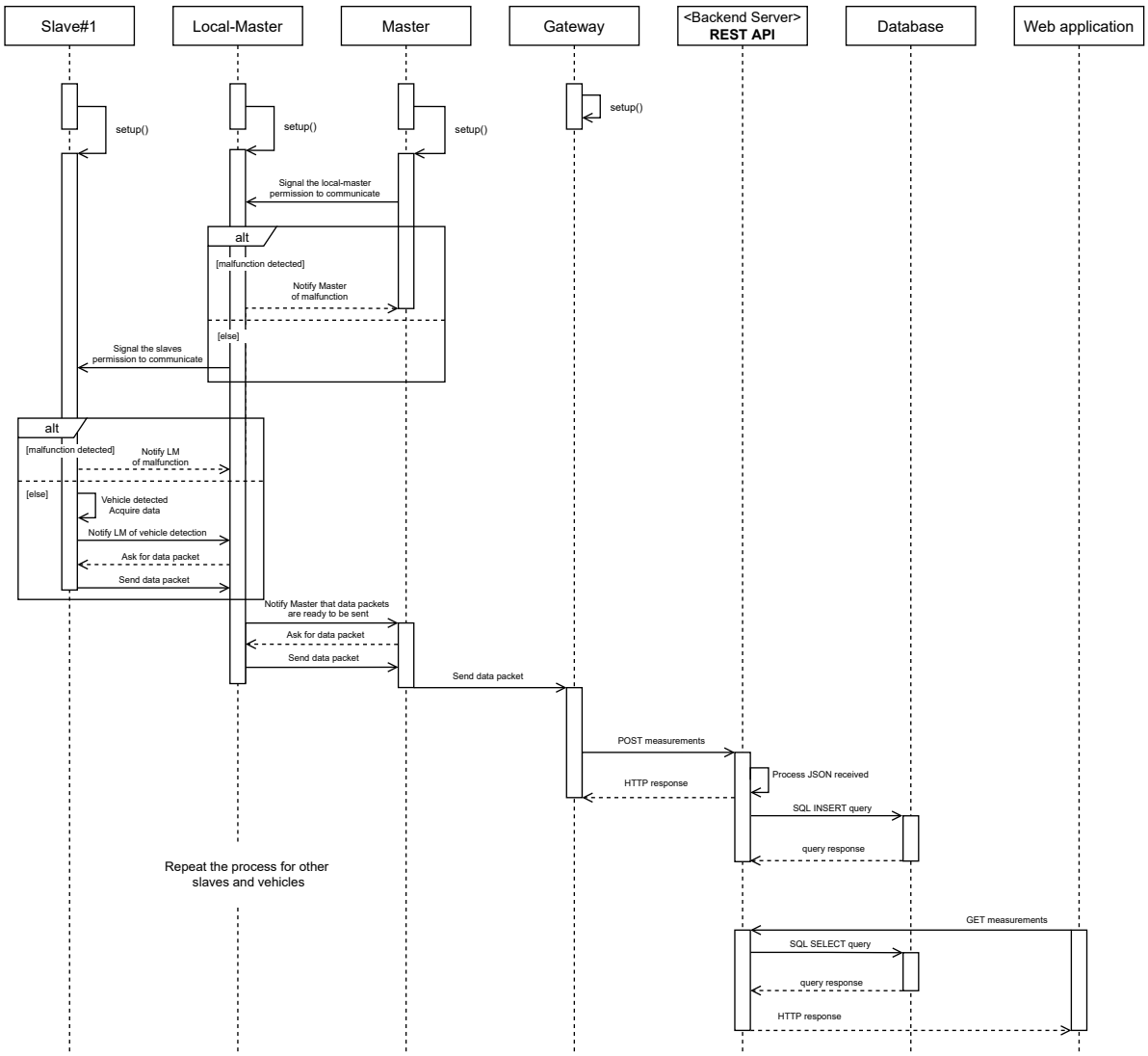


Figure 3.25: System's UML sequence diagram

3.6 Conclusion

The sequence diagram in figure 3.25 depicts the overall structure of the system, from the *slave*s to the web application. It represents a simplified version, with only one *slave* to make it readable, but the sequence can be expanded for multiple *slave*s and *local-master*s configurations.

3.6.1 Integration procedures

To access the information in the database, to assist in the development of the APIs and to check if the gateway was correctly inserting data into the database, a PostgreSQL management tool, PgAdmin 4, was used. This software includes a graphical user interface that simplifies the data visualization of a PostgreSQL server, as depicted in figure 3.26.

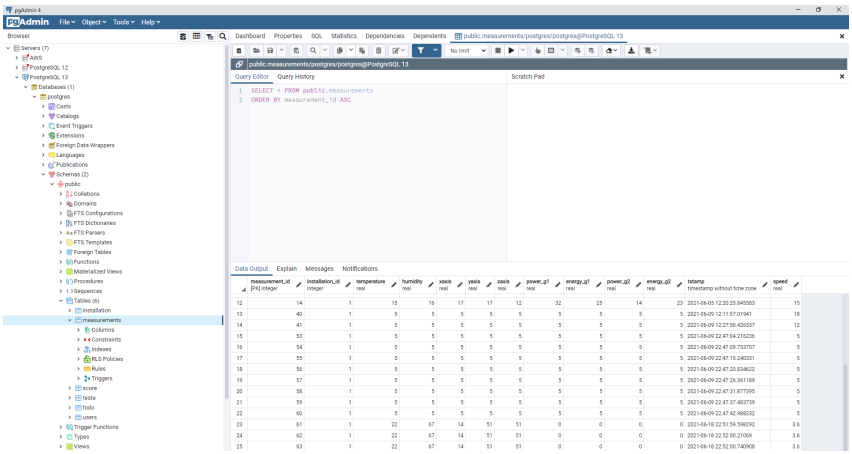


Figure 3.26: PgAdmin Graphical Interface

The data presentation application received information from the database through the PERN stack. Accordingly, the HTTP protocol allowed the connection between the web application and the PERN stack. In order to test this connection, the open source software Swagger was used, where the APIs and respective functionalities were listed, validating the implementation before integrating it into the web application. Its interface is depicted in figure 3.27.

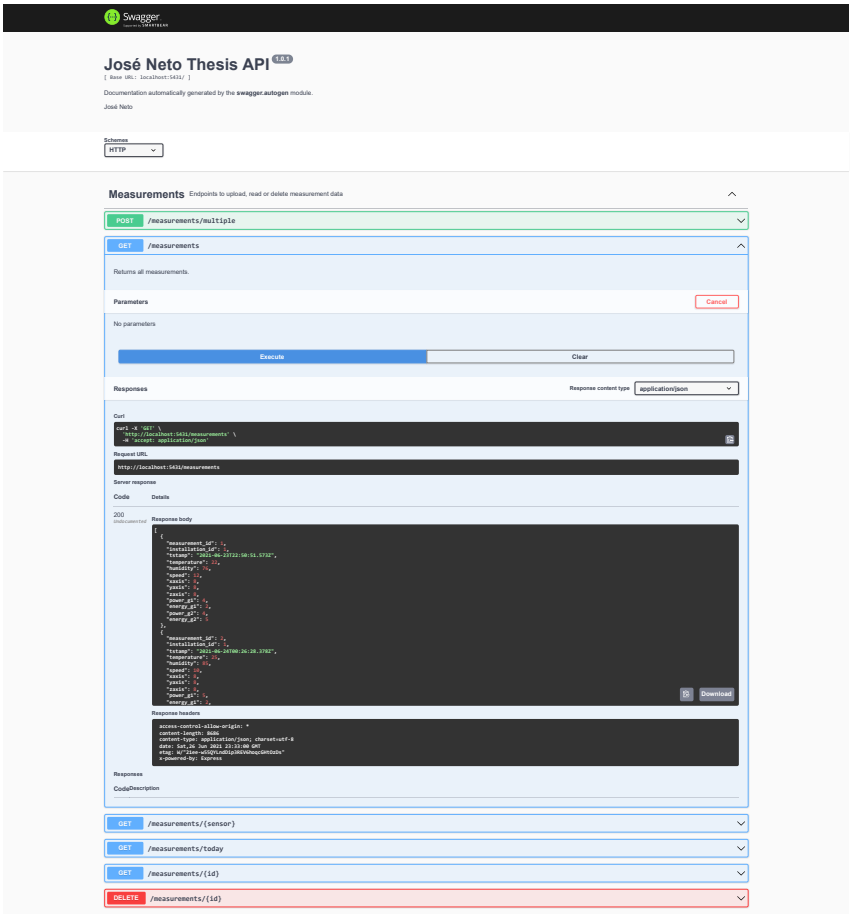


Figure 3.27: Swagger Graphical Interface

Chapter 4

Tests and results

4.1 Energy generated tests

Initially, the behaviour of the z-axis acceleration of the prototype was studied using one person jumping as actuator. The waveform is depicted in figure 4.1.

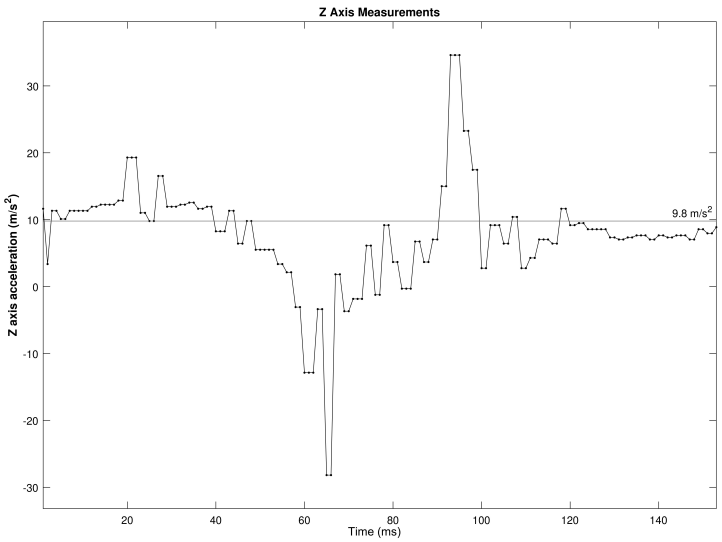


Figure 4.1: z-axis acceleration waveform

In order to validate the estimation of the energy generated different tests were performed. A MATLAB script that read a text file with the measurements sent from the *slave*'s via RS-485 was developed.

The first trials focused on testing the *slave*'s ADC by supplying a know waveform from a signal generator. The next step was to evaluate its performance on the prototypes.

The actuation of the generators inside the prototype creates a voltage wave as shown in figure 4.2. The voltage signal was acquired using one *slave* ADC channel.

As seen in figure 4.2, the voltage decreases over time until it becomes zero. The estimation of energy produced by a generator is considered to be over when six successive readings below

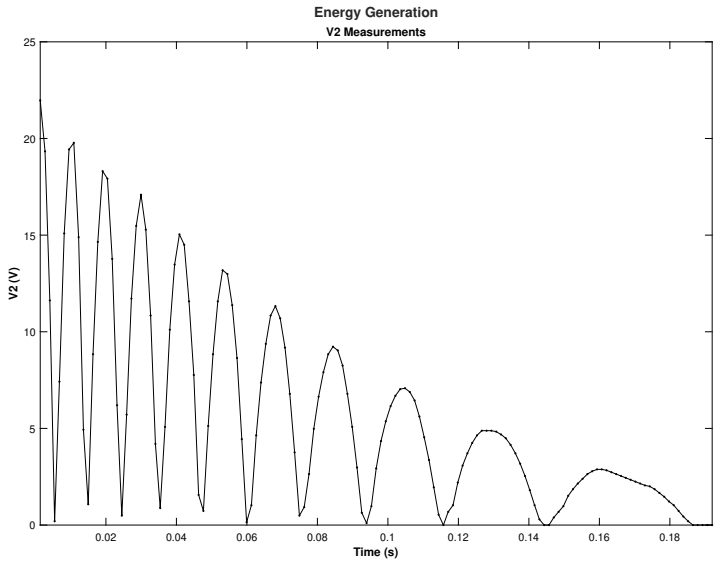


Figure 4.2: Generator voltage waveform

a threshold level (0.2 V) are detected. The reading is finished when the waves of both generators have this behaviour. The process is repeated for the vehicle's second axle.

The system will be deployed in cities, so the vast majority of vehicles has two axles, meaning that a passing vehicle causes two triggering events. Figure 4.3 was obtained using the MATLAB script and depicts a test performed in the laboratory using a prototype and one person jumping as actuator instead of a vehicle.

The interval between the triggering events was about 2.7 seconds, but the interval was cut in order to showcase the readings clearly.

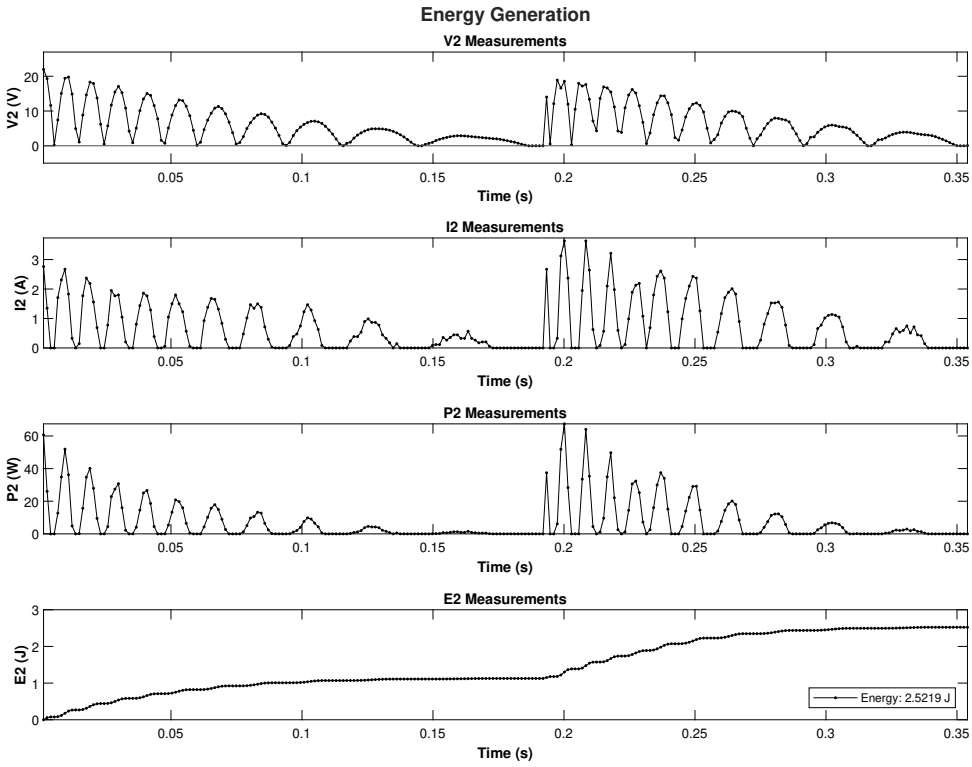


Figure 4.3: Power section tests

4.2 Overall system tests

After validating the data acquisition subsystem, the performance of the entire project was evaluated. Figure 4.4 depicts the prototypes used in the tests.

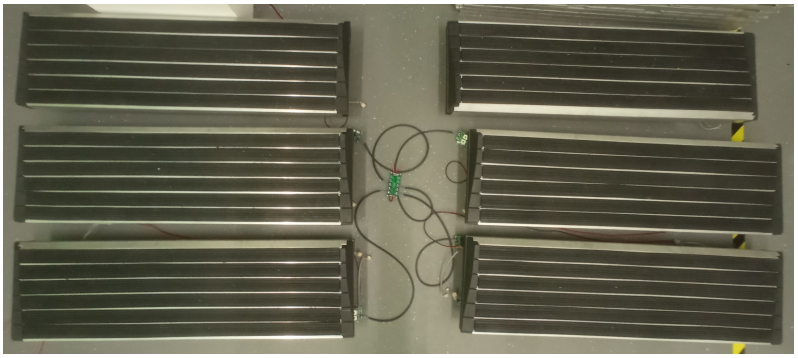


Figure 4.4: Prototypes used in the tests performed

Initially, the system was tested with six *slave*s and one *local-master*. All of the *slave*s were capable of acquiring the data and the other subsystems were able to make it reach the web application successfully. This experiment proved that the system is ready to be implemented outside of the laboratory and the next step is to use vehicles as actuators to test the implementation in a

non-controlled environment.

A similar experiment was conducted, this time with two *local-master*s, each with three *slave*s. The results were meaningful as well, as the system was capable of getting the data to the web application.

Chapter 5

Conclusion and future work

This dissertation focused on the development of a monitoring and data communication system for application in Pavement Energy Harvesting. For the past decades, this type of technologies have been the focus of scientific research and development all around the world.

The tests conducted validated the proposed structure, from the data acquisition by the *slave*s, their management by the *local-master*and the *master* to its presentation in the web application.

The work developed covered a wide range of concepts and knowledge related with electrical engineering. That was the condition that convinced me to accept the project and I'm satisfied with end result.

5.1 Future work

Some of the sensors used were not the first option due to the global chip shortage and limited stock, but their performance didn't compromise the results. However, other options should be considered to increase the product's value.

The estimation and calculation of the vehicles' speed can be improved. One possibility is to connect the *slave*'s interrupts to the *local-master*. By knowing the distance between consecutive *slave*s, the *local-master* can better estimate the velocity of the vehicles.

The data sent by the *slave*s could be validated by the *local-master* before being forwarded to the *master*. That way, malfunctions could be prevented and noticed if a lot of variance between *slave* readings was detected.

The stability of the prototypes can be better estimated by using IMUs and sensor fusion algorithms that fully describe an object behaviour in space.

As successful as the experiments performed were, the system needs to be tested using vehicles as actuators to fully understand its applicability in Pavnext's installations.

However, I'm confident that no significant changes to the proposed architecture will be needed, as the vehicles actuation only affects the behaviour of the mechanical aspect of the prototypes and not the software side that was the focus of this dissertation.

References

[1] Francisco Duarte and Adelino Ferreira. Energy harvesting on road pavements: state of the art. *Proceedings of the Institution of Civil Engineers - Energy*, 169(2):79–90, 2016. URL: https://doi.org/10.1680/jener.15.00005, doi:10.1680/jener.15.00005. Cited on pages ix, 3, 4, 5, 6, and 7.

[2] David Robin Green, Jim Ward, and Neil Wyper. Solar-powered wireless crosswalk warning system, Jan 2008. Cited on pages ix and 4.

[3] Chrysanthi Efthymiou, Mat Santamouris, Dionysia Kolokotsa, and Andreas Koras. Development and testing of photovoltaic pavement for heat island mitigation. *Solar Energy*, 130:148–160, 2016. URL: https://www.sciencedirect.com/science/article/pii/S0038092X16000785, doi:https://doi.org/10.1016/j.solener.2016.01.054. Cited on pages ix and 4.

[4] Linda Poon. Unlocking the hidden power of a highway, Mar 2017. URL: https://www.bloomberg.com/news/articles/2017-03-15/how-a-road-in-rural-georgia-is-bringing-power-to-highways. Cited on pages ix and 4.

[5] Andrew Dawson, Rajib Mallick, Alvaro García Hernandez, and Pejman Keikhaei Dehdezi. *Energy Harvesting from Pavements*, pages 481–517. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. URL: https://doi.org/10.1007/978-3-662-44719-2_18, doi:10.1007/978-3-662-44719-2_18. Cited on pages ix, 3, and 4.

[6] Alvaro García and Manfred N. Partl. How to transform an asphalt concrete pavement into a solar turbine. *Applied Energy*, 119:431–437, 2014. URL: https://www.sciencedirect.com/science/article/pii/S0306261914000257, doi:https://doi.org/10.1016/j.apenergy.2014.01.006. Cited on pages ix and 4.

[7] Wenjuan Sun, Guoyang Lu, Cheng Ye, Shiwu Chen, Yue Hou, Dawei Wang, Linbing Wang, and Markus Oeser. The State of the Art: Application of Green Technology in Sustainable Pavement. *Advances in Materials Science and Engineering*, 2018:1–19, jun 2018. URL: https://www.hindawi.com/journals/amse/2018/9760464/, doi:10.1155/2018/9760464. Cited on pages ix, 3, 4, 5, and 6.

[8] Pallavi Sethi and Smruti R. Sarangi. Internet of Things: Architectures, Protocols, and Applications, 2017. doi:10.1155/2017/9324035. Cited on pages ix and 9.

[9] Jozef Mocnej, Adrian Pekar, Winston Seah, Erik Kajáti, and Iveta Zolotová. INTERNET OF THINGS UNIFIED PROTOCOL STACK. *Acta Electrotechnica et Informatica*, 19(2):24–32, jun 2019. URL: http://www.aei.tuke.sk/papers/2019/2/04_Mocnej.pdf, doi:10.15546/aeei-2019-0011. Cited on pages ix, 10, and 11.

[10] Francisco João Anastácio Duarte. *Pavement energy harvesting system to convert vehicles kinetic energy into electricity*. PhD thesis, 00500:: Universidade de Coimbra, 2018. Cited on pages ix, 15, and 16.

[11] Tom J. Kazmierski and Steve Beeby. *Energy Harvesting Systems: Principles, Modeling and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2010. Cited on page 3.

[12] Hao Wang, Abbas Jasim, and Xiaodan Chen. Energy harvesting technologies in roadway and bridge for different applications – a comprehensive review. *Applied Energy*, 212:1083–1094, 2018. URL: https://www.sciencedirect.com/science/article/pii/S0306261917318500, doi:https://doi.org/10.1016/j.apenergy.2017.12.125. Cited on pages 3 and 7.

[13] Wikipedia contributors. Timeline of solar cells — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Timeline_of_solar_cells&oldid=1024353048, 2021. [Online; accessed 25-May-2021]. Cited on page 3.

[14] Wikipedia contributors. Solaroad — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=SolaRoad&oldid=1020340087, 2021. [Online; accessed 26-May-2021]. Cited on page 5.

[15] Solarroadways. https://solarroadways.com. [Online; accessed 25-May-2021]. Cited on page 5.

[16] T. Voigt, H. Ritter, and J. Schiller. Utilizing solar power in wireless sensor networks. In *28th Annual IEEE International Conference on Local Computer Networks, 2003. LCN '03. Proceedings.*, pages 416–422, 2003. doi:10.1109/LCN.2003.1243167. Cited on page 5.

[17] Vanesa Bobes-Jesus, Pablo Pascual-Muñoz, Daniel Castro-Fresno, and Jorge Rodriguez-Hernandez. Asphalt solar collectors: A literature review. *Applied Energy*, 102:962–970, 2013. Special Issue on Advances in sustainable biofuel production and use - XIX International Symposium on Alcohol Fuels - ISAF. URL: https://www.sciencedirect.com/science/article/pii/S030626191200637X, doi:https://doi.org/10.1016/j.apenergy.2012.08.050. Cited on page 6.

[18] Siti Diana Nasir, Weijie Xu, Becky Vital, Conrad Pantua, Bochao Zhou, John Calautit, and Ben Hughes. Urban road and pavement solar collector system for heat island mitigation: assessing the beneficial impact on outdoor temperature. *IOP Conference Series: Earth and Environmental Science*, 463:012038, apr 2020. URL: https://doi.org/10.1088/1755-1315/463/1/012038, doi:10.1088/1755-1315/463/1/012038. Cited on page 6.

[19] A. Chiarelli, A. Al-Mohammedawi, A.R. Dawson, and A. García. Construction and configuration of convection-powered asphalt solar collectors for the reduction of urban temperatures. *International Journal of Thermal Sciences*, 112:242–251, 2017. URL: https://www.sciencedirect.com/science/article/pii/S1290072916305877, doi:https://doi.org/10.1016/j.ijthermalsci.2016.10.012. Cited on page 6.

[20] Diana S.N.M. Nasir, Ben Richard Hughes, and John Kaiser Calautit. A study of the impact of building geometry on the thermal performance of road pavement solar collectors. *Energy*, 93(P2):2614–2630, 2015. URL: https://ideas.repec.org/a/eee/energy/v93y2015ip2p2614-2630.html, doi:10.1016/j.energy.2015.09. Cited on page 6.

[21] Wikipedia contributors. Thermoelectric effect — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Thermoelectric_effect&oldid=1022101685, 2021. [Online; accessed 27-May-2021]. Cited on page 6.

[22] Lukai Guo and Qing Lu. Potentials of piezoelectric and thermoelectric technologies for harvesting energy from pavements. *Renewable and Sustainable Energy Reviews*, 72:761–773, 2017. URL: https://www.sciencedirect.com/science/article/pii/S136403211730103X, doi:https://doi.org/10.1016/j.rser.2017.01.090. Cited on page 6.

[23] M. Hasebe, Y. Kamikawa, and S. Meiarashi. Thermoelectric generators using solar thermal energy in heated road pavement. In *2006 25th International Conference on Thermoelectrics*, pages 697–700, 2006. doi:10.1109/ICT.2006.331237. Cited on page 6.

[24] Philip Park, G.S. Choi, Ehsan Rohani, and Ikkyun Song. Optimization of thermoelectric system for pavement energy harvesting. *Asphalt Pavements - Proceedings of the International Conference on Asphalt Pavements, ISAP 2014*, 2:1827–1838, 07 2014. URL: https://www.researchgate.net/publication/288735756_Optimization_of_thermoelectric_system_for_pavement_energy_harvesting, doi:10.1201/b17219-220. Cited on page 6.

[25] Wikipedia contributors. Piezoelectricity — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Piezoelectricity&oldid=1023500401, 2021. [Online; accessed 30-May-2021]. Cited on page 7.

[26] Diogo Correia and Adelino Ferreira. Energy harvesting on airport pavements: State-of-the-art. *Sustainability*, 13(11), 2021. URL: https://www.mdpi.com/2071-1050/13/11/5893, doi:10.3390/su13115893. Cited on page 7.

[27] Hongduo ZHAO, Jian YU, and Jianming LING. Finite element analysis of cymbal piezoelectric transducers for harvesting energy from asphalt pavement. *Journal of the Ceramic Society of Japan*, 118(1382):909–915, 2010. doi:10.2109/jcersj2.118.909. Cited on page 7.

[28] Chaohui Wang, Jianxiong Zhao, Qiang Li, and Yanwei Li. Optimization design and experimental investigation of piezoelectric energy harvesting devices for pavement. *Applied Energy*, 229:18–30, 2018. URL: https://www.sciencedirect.com/science/article/pii/S030626191831064X, doi:https://doi.org/10.1016/j.apenergy.2018.07.036. Cited on page 7.

[29] Haim Abramovich, Eugeny Harash, Charles Milgrom, and Uri Amit. Energy harvesting [WO2009098676A1], August 2009. Cited on page 7.

[30] Haim Abramovich, Eugeny Harash, Charles Milgrom, Uri Amit, and Lucy Edery Azulay. Energy harvesting from roads and airport runways [CA2715129C], November 2011. Cited on page 7.

[31] L. CAMPBELL, Ira. Piezoelectric energy harvesting systems and methods [WO2015157377A1], October 2015. Cited on page 7.

[32] Xiaoguang Yuan. Road surface impact energy harvesting device and calculation method [CN112468019A], March 2021. Cited on page 7.

[33] Fadi Al-Turjman, Chadi Altrjman, Sadia Din, and Anand Paul. Energy monitoring in iot-based ad hoc networks: An overview. *Computers & Electrical Engineering*, 76:133–142, 2019. URL: https://www.sciencedirect.com/science/article/pii/S0045790618316082, doi:https://doi.org/10.1016/j.compeleceng.2019.03.013. Cited on page 7.

[34] Maisagalla Gopal, T Chandra Prakash, N Venkata Ramakrishna, and Bonthala Prabhanjan Yadav. IoT based solar power monitoring system. *IOP Conference Series: Materials Science and Engineering*, 981:032037, dec 2020. URL: https://doi.org/10.1088/1757-899x/981/3/032037, doi:10.1088/1757-899x/981/3/032037. Cited on page 8.

[35] MaŁ,ecki Krzysztof. The importance of automatic traffic lights time algorithms to reduce the negative impact of transport on the urban environment. *Transportation Research Procedia*, 16:329–342, 2016. The 2nd International Conference "Green Cities - Green Logistics for Greener Cities", 2-3 March 2016, Szczecin, Poland. URL: https://www.sciencedirect.com/science/article/pii/S2352146516306469, doi:https://doi.org/10.1016/j.trpro.2016.11.032. Cited on page 8.

[36] Aminah Hardwan Ahmed and Luca Zanotti Fragonara. Adaptive intelligent traffic control systems for improving traffic quality and congestion in smart cities. *International Journal for Quality Research*, 15(1):139–154, 2021. doi:10.24874/IJQR15.01-08. Cited on page 8.

[37] Tamer Omar, Daniel Bovard, and Huy Tran. Smart cities traffic congestion monitoring and control system. In *Proceedings of the 2020 ACM Southeast Conference*, ACM SE '20, page 115–121, New York, NY, USA, 2020. Association for Computing Machinery. URL: https://doi.org/10.1145/3374135.3385271, doi:10.1145/3374135.3385271. Cited on page 8.

[38] Carlos Justo de Frías, Abdulla Al-Kaff, Francisco Miguel Moreno, Ángel Madridano, and José María Armingol. Intelligent cooperative system for traffic monitoring in smart cities. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 33–38, 2020. doi:10.1109/IV47402.2020.9304649. Cited on page 8.

[39] Saba Latif, Hamra Afzaal, and Nazir Ahmad Zafar. Intelligent traffic monitoring and guidance system for smart city. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–6, 2018. doi:10.1109/ICOMET.2018.8346327. Cited on page 9.

[40] Arm designs: Intelligent roadside unit. https://www.arm.com/why-arm/innovation/arm-designs/smart-transportation. [Online; accessed 28-May-2021]. Cited on page 9.

[41] Wikipedia contributors. Intelligent transportation system — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Intelligent_transportation_system&oldid=1021627764, 2021. [Online; accessed 28-May-2021]. Cited on page 9.

[42] Huanyang Zheng, Wei Chang, and Jie Wu. Traffic flow monitoring systems in smart cities: Coverage and distinguishability among vehicles. *Journal of Parallel and Distributed Computing*, 127:224–237, 2019. URL: https://

www.sciencedirect.com/science/article/pii/S074373151830488X, doi:https://doi.org/10.1016/j.jpdc.2018.07.008. Cited on page 9.

[43] Wikipedia contributors. 15 minute city — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=15_minute_city&oldid=1023017490, 2021. [Online; accessed 28-May-2021]. Cited on page 9.

[44] Carlos Moreno. The 15-minute city [video]. https://www.ted.com/talks/carlos_moreno_the_15_minute_city. [Online; accessed 28-May-2021]. Cited on page 9.

[45] Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, and Mahmood Alzubaidi. Internet of Things (IoT) communication protocols: Review. In *2017 8th International Conference on Information Technology (ICIT)*, pages 685–690. IEEE, may 2017. URL: http://ieeexplore.ieee.org/document/8079928/, doi:10.1109/ICITECH.2017.8079928. Cited on pages 10 and 11.

[46] Mads Lauridsen, Huan Nguyen, Benny Vejlgaard, Istvan Z. Kovacs, Preben Mogensen, and Mads Sorensen. Coverage Comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km Area. In *IEEE Vehicular Technology Conference*, volume 2017-June. Institute of Electrical and Electronics Engineers Inc., nov 2017. doi:10.1109/VTCSpring.2017.8108182. Cited on page 10.

[47] Wikipedia contributors. Wi-fi — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Wi-Fi&oldid=1025469393, 2021. [Online; accessed 29-May-2021]. Cited on page 11.

[48] Sanjeeb Mishra, Neeraj Kumar Singh, and Vijayakrishnan Rousseau. Chapter 10 - sensor interfaces. In Sanjeeb Mishra, Neeraj Kumar Singh, and Vijayakrishnan Rousseau, editors, *System on Chip Interfaces for Low Power Design*, pages 331–344. Morgan Kaufmann, 2016. URL: https://www.sciencedirect.com/science/article/pii/B9780128016305000104, doi:https://doi.org/10.1016/B978-0-12-801630-5.00010-4. Cited on page 19.

[49] Accelerometer, gyro and imu buying guide. https://www.sparkfun.com/pages/accel_gyro_guide. [Online; accessed 16-June-2021]. Cited on page 22.