

Development of an automated bicycle parking spot for a smart parking system

Samuel Davi Werner

Work oriented by:

Prof. PhD. Paulo Jorge Pinto Leitão

Prof. Dr. César Rafael Claire Torrico

Bragança, Portugal

2019/2020

Development of an automated bicycle parking spot for a smart parking system

Samuel Davi Werner

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master's Degree in Industrial Engineering under the scope of the Double Degree agreement with UTFPR. This work was oriented by professor PhD. Paulo Jorge Pinto Leitão, from IPB, and professor Dr. César Rafael Claure Torrico, from UTFPR.

Bragança, Portugal

2019/2020

"If I have seen further it is by standing on the shoulders of Giants."

- *Isaac Newton*

Dedication

This work is dedicated, first, to my family. Eliane and Leandro, you have always been a continuous source of inspiration. You encouraged me to achieve my most improbable aspirations and gave me a reason to inspire someone else, Helena.

To professor Carl Sagan, who lit a candle in a world haunted by demons. His passion for science, the universe, and everything else met me in the most chaotic times, flourishing in me much of who I turned out to be. I hope I never lose his kind and curious approach to Nature and humankind.

Ana, you taught me how to be a man in the most dubious and frightening world. You stood by my side in moments I thought I was alone. You inspire me to never give up on my patience, kindness, and balance, even in the most adverse situations. May we always hold our remarkable love for each other.

Salete, my mother, you are the giant among giants. From your shoulders, I stood to see the world. I am forever grateful for having you. Everything I achieve comes from your unbeatable resilience and tireless endeavor.

Acknowledgement

I begin thanking the direct supervisors that worked alongside me in this period. To professor Paulo Leitão and César Torrico for the great examples of professionalism, countless meetings and pieces of advice, and for the chance of developing projects under your supervision. Also, to Lucas Sakurada and Wellington Maidana for the guidance and partnership. Furthermore, I appreciate the efforts of the professors that made the Double Degree program possible, both from UTFPR and IPB, extending my appreciation to the institutions themselves for this opportunity.

I thank all my friends from Brazil, who encouraged me to board on this journey to the other side of the ocean. Also, I remember all the friends I met in Portugal, who helped me in many ways through the development of this work. Finally, I will never forget any of the 4ESQ roommates, my Portuguese family of Brazilians, and the moments we spent together, from marathoning the Star Wars series to the hours-long Age of Empires LANs.

Thank you to my family and my godfather. I never considered to be alone, even in the most adverse times, because I knew you were there for me. Your trust and presence support all the progress I achieve.

Abstract

Smart parking systems are promising solutions for a set of traffic-related problems in major cities across the world. The goal of those systems is to guide users through paths in which they spend less time, resources, and release fewer greenhouse gases to find a parking spot. To this end, deployers develop Cyber-physical Systems that generally comprise embedded electronics materials, Internet of Things technologies, and Artificial Intelligence concepts. This work combines ESP8266 microcontrollers and Raspberry Pi microprocessors through MQTT communication protocol to implement its architecture, a few possible different options for the actuator are also presented, and a project for the power supply by low-current photovoltaic panels is documented. Therefore, the goal is to work over some options and ideas for the physical implementation of the low-level electronics physical stage of a smart parking Cyber-physical System. The results include validated actuator options, a small photovoltaic generation sizing, and the deployment of a microcontroller routine capable of properly operate as a physical asset controller enabling scalability.

Keywords: Smart parking systems; Cyber-physical systems; Internet of Things; Embedded electronics.

Resumo

Sistemas de estacionamento inteligentes são soluções promissoras para uma gama de problemas relacionados a tráfego de automóveis em grandes cidades do mundo. O objetivo destes sistemas é guiar seus usuários por caminhos pelos quais os mesmos gastam menos tempo, recursos e liberam menos gases contribuintes para o efeito estufa a fim de encontrar um local de estacionamento. Para este fim, desenvolvedores implementam Sistemas Ciber-físicos que geralmente incluem materiais de eletrônica embebida, tecnologias de Internet das Coisas e conceitos de Inteligência Artificial. Este trabalho combina os microcontroladores ESP8266 e microprocessadores Raspberry Pi pelo protocolo de comunicação MQTT a fim de implementar sua arquitetura definida, também apresenta algumas possíveis opções para a implementação de um atuador e o projeto para suprir o consumo de eletricidade por painéis fotovoltaicos de baixa corrente. Portanto, o objetivo é trabalhar em possíveis opções e ideias para a implementação física da etapa de eletrônica de baixo nível de um Sistema Ciber-físico para estacionamentos inteligentes. Os resultados incluem opções validadas de atuadores, um dimensionamento de geração fotovoltaica de baixa potência e o desenvolvimento de uma rotina para o que o microcontrolador aja como um controlador local e permita escalabilidade.

Palavras-chave: Sistemas de estacionamento inteligentes; Sistemas Ciber-físicos; Internet das Coisas; Eletrônica embebida.

Contents

Abstract	xi
Resumo	xiii
Acronyms	xxiii
1 Introduction	1
1.1 Context and motivation	1
1.2 Objectives	3
1.3 Document structure	4
2 Related works	5
2.1 Internet of Things and machine-to-machine	5
2.1.1 Cyber-physical Systems (CPS)	7
2.1.2 Industry 4.0	8
2.2 Smart cities	10
2.3 Smart parking systems	12
2.3.1 Demand for intelligent systems	12
2.3.2 Academic solutions	16
2.3.3 Commercial options	16
3 Study object and system’s architecture	19
3.1 Description of the study object	19

3.1.1	Overview, characteristics, and features	20
3.1.2	System operation	23
3.2	Architecture	24
3.2.1	Multi-agent Systems (MAS)	25
3.2.2	Cyber-to-physical link	27
3.2.3	Linear solenoid	30
3.2.4	Electromagnets	31
4	Physical implementation	33
4.1	Actuator options	33
4.1.1	Linear-to-rotational mechanism	34
4.1.2	Solenoid piston	39
4.1.3	Electromagnetic locker	40
4.1.4	Stepper motor	41
4.1.5	Electronic drivers	42
4.2	Sensor operation	44
4.3	Embedded power supply	45
4.3.1	Photovoltaic generation	45
4.3.2	Batteries and BMS	46
4.4	Microcontroller	47
5	Results	49
5.1	Software resources	49
5.2	Mechanical design	50
5.3	Magnetic distributions	54
5.4	Electronic simulations	58
5.5	Photovoltaic panels and batteries sizing	62
5.6	Microcontroller routine	64
6	Conclusions and future work	69

A FEMM data	79
B ESP8266 code	83

List of Tables

2.1	Vehicles per thousand people in 2007 and 2017 [30].	13
2.2	The five most congested cities in the world [38].	15
3.1	The sequence of standardized messages for MQTT communication.	28
5.1	Solenoid's parameters to define the electronic drivers' specifications	58
A.1	Mechanism's magnetic simulation data from FEMM treated with UARM eq.	79

List of Figures

2.1	Four stages of Cyber-physical Systems (CPS). Adapted from [20].	8
2.2	Industrial revolutions and its starting points. Adapted from [23].	9
2.3	Smart city sample model. Adapted from [27].	10
2.4	Europe’s GHG emissions trend projection and targets [35].	14
2.5	Patents on bicycle’s locking mechanisms [49]–[52].	17
3.1	Frontal, side, superior and isometric views of the studied parking spot. . .	20
3.2	Generic scheme of an agent-based smart parking architecture [11].	21
3.3	Defined structure for the agent-based CPS smart parking.	22
3.4	Scheme of the system’s structure of control.	24
3.5	Scheme of the project scope.	25
3.6	Smart parking MAS schematic. Adapted from [11].	26
3.7	MQTT standard operation flux [54].	28
3.8	MQTT simplified tests scheme.	29
3.9	Generic solenoid scheme.	30
3.10	Terms of the solenoid equations.	31
3.11	Unlocking forces with an electromagnet.	32
4.1	Views of the three parts of the mechanism.	34
4.2	Simplified magnetic field and circuit design.	35
4.3	Isometric and lateral view of the mechanism’s three key positions.	35
4.4	Isometric and lateral view of the open and closed mechanism.	36
4.5	Generic ideal magnetic field flux density and force versus distance.	37

4.6	Considered design for the magnetic simulations.	38
4.7	All the drivers of mechanical movement assembled.	38
4.8	Solenoid piston simplified mechanical structure.	39
4.9	Electromagnet locking simplified mechanical structures.	40
4.10	Simplified scheme of a stepper motor in a motorized smart parking spot.	41
4.11	Three possible topologies for the implementation of the electronic drives. (a) Darlington transistor, (b) H bridge, (c) Relay.	42
4.12	Pull-up resistor structure for the sensor.	44
4.13	Proposed power flow.	45
4.14	Current flow during expected two operation states.	46
4.15	ESP8266 NodeMCU Lua Wi-Fi pinout layout.	47
4.16	ESP8266 MCU proposed schematic.	48
4.17	ESP8266-01 series module Wi-Fi pinout layout.	48
5.1	Mechanical project for the mechanism proposal.	50
5.2	Mechanical project for the double solenoid proposal.	51
5.3	Mechanical projects for the electromagnet and stepper motor proposals.	52
5.4	Mechanical designs scheme and proposed operation.	53
5.5	Drawing of the simulated solenoid.	54
5.6	Forces diagram analysis to treat the data from FEMM.	55
5.7	Physical behavior of the components from treated FEMM simulation data. (a) Force on the piston (FEMM) vs. distance between piston and solenoid; (b) Acceleration and speed of the piston (UARM eq. predicted) vs. time; (c) Traveled distance (UARM eq. predicted) vs. time.	56
5.8	Electronic circuits as simulated in LT Spice.	59
5.9	Simulated behavior of the (a) Darlington and relay, and (b) H bridge.	60
5.10	Simulated power losses of the (a) Darlington transistor and (b) H bridge.	60
5.11	Flowchart of the MCU code.	64

Acronyms

BMS	Battery management system
BSS	Bike sharing systems
CPS	Cyber-physical Systems
EU	European Union
FPC	Fixed path cylinder
GHG	Greenhouse gases
GPIO	General-purpose input/output port
HSCpD	Hours spent in congestions per driver
IoT	Internet of Things
LMG	Linear movement guide
M2M	Machine-to-machine
MAS	Multi-agent Systems
MCU	Microcontroller Unit
MQTT	Message Queuing Telemetry Transport
Mt CO₂ e.	Million tones of carbon dioxide equivalent
PAC	Physical Asset Controller
PCB	Printed circuit board
PV	Photovoltaic
PWM	Pulse width modulation
RC	Rotational component
UARM eq.	Uniformly accelerated rectilinear motion equations
WSN	Wireless Sensors Network

Chapter 1

Introduction

On the edge of industry technology development, there are new engineering techniques transforming processes completely, focusing on massive human-free communication and autonomous decision-making. Those form the base of what is called the Internet of Things and Industry 4.0, concepts and technologies enabling deployment of complex systems in fields such as smart cities, energy conservation, and agriculture [1]. This project is framed in the context of a concept known as Cyber-physical Systems (CPS), which consists of integration between software for control and decision-making, multiple devices networking, and physical electronic circuits capable of sensing, acting, and communicating [2]–[4]. The focus of this work is on the physical implementation of a smart parking system for cyclists aiming to change the way they interact with parking spots.

1.1 Context and motivation

Major cities across the world have increasingly struggled with a recurrent problem: traffic. From the Americas to Asia, there are thousands of crowded streets where cars are not able to fit or move efficiently. Cars accumulating in road traffic cause several different problems, such as air and noise pollution, a higher chance of accidents, pressure on natural resources, more fuel consumption, greater policing costs, and a significant waste of time for the drivers [5], [6].

In this sense, new technologies are in development to reduce such problems. Some governments are already seeking to change these scenarios, and solutions start to appear locally. In Amsterdam, for example, the government is world-widely known for cycling incentives [7]; likewise, cities like London, Oslo, Hamburg, Madrid, and Barcelona recently announced long-term plans to ban cars from downtown areas [8]. The IoT trend is one of the possibilities that offer potential solutions for both cases: first, to enhance bike infrastructure but also to offer means to decrease car traffic.

The feasibility of the Smart Parking system is a matter of great importance. Bicycle parking is almost omnipresent in European cities of all sizes, so automation plans must be potentially scalable accordingly. Additionally, the project is expected to offer a low-cost implementation so it can succeed and justify its adoption and relevance. Along with that, the software's versatility for different types of vehicles is a desirable feature.

Smart Parking systems are implemented from many distinct materials and techniques. A literature review around the latest reported smart systems verifies the diversity of approaches used for their deployment. For example, [9] proposes the substitution of position sensors by an image recognizing algorithm based on machine learning to detect free spots. Further, there is [10], which uses an Android platform for the management of parking spots and users. Finally, this project itself is based on the use of industrial Multi-agent Systems (MAS) in the context of smart parking [11], [12]. The results are promising and, although this is still a topic in scientific maturity status, Smart Systems shows signs of being, soon, a viable context-changing technology [13].

In essence, the goal of CPS systems based on IoT technologies is to rethink ordinary objects and ultimately transform their use. Thus, smart parking systems emerge from the demand for efficiency in the transport sectors. If a car, bike, bus, or any vehicle, already has a reserved parking spot when reaching its destination, there is a significant decrease in parking searching time. The claim that the Internet of Things concept can help solve such a big problem in today's society, like car congestion, is a big statement. However, there are high hopes that one of the driving technologies of the fourth industrial revolution can also bring improvements for everyday applications [1].

1.2 Objectives

This work is focused on the electronic stage design, validation, and implementation conceptualization of a bicycle parking dock. For this, the connection between the microcontrollers: a) agent-based programmed running virtually in the cloud; and, b) used in the physical implementation as the physical asset controller, is essential. Moreover, all electronic instrumentation and automation shall be properly interconnected taking sensors, actuators, controllers, and user interfaces into account. Further, it is expected an effort for the minimization of overall spending to propose a low-cost solution. Enumerating, the main objectives of this project are:

1. Implement an effective communication between the cloud computing microcontroller through a Message-Queue Telemetry Transport (MQTT) broker connected to the local network and the physical asset controller;
2. Propose different actuator models, calculate, and validate reliable, low-cost, and secure padlocks for parked bicycles;
3. Find solutions for the sensing and data flow in real-time to enable reliable control of multiple bicycle padlocks; and,
4. Provide means for scale implementation.

1.3 Document structure

This work is ordered in six chapters. It begins with chapter 1, *Introduction*, which approaches the contextualization and motivation, the objectives, and this structure detailing.

Chapter 2, *Related Works*, discusses the relevant related academic research and commercial options in the field of smart parking systems. The first section analyses the Internet of Things (IoT) concept and machine-to-machine (M2M) technology, followed by Smart Cities, and finished by a specific Smart Parking Systems inquiry.

Chapter 3, *Study object and system's architecture*, displays the studied parking and the approach adopted to deploy this project. It is divided into a description of the study object and the techniques and concepts present on the defined architecture.

Chapter 4, *Physical implementation*, explains the proposed actuators structure and sensor operation, the chosen power supplier, the defined routine for the microcontroller, and the electronic drivers options.

Chapter 5, *Results*, presents the results obtained from the previously designed structures and architectures. It also lists the multiple software used for the development of simulations aiming to validate this work's designs and implementation choices.

Chapter 6, *Conclusions and future work*, summarizes the whole project providing a conclusion and ideas for future works in the same research field.

Lastly, this document also contains two Appendices. Appendix A comprises the data obtained from the simulations and treated according to physical principles, and Appendix B consists of a transcription of important excerpts from the microcontroller codes.

Chapter 2

Related works

This chapter seeks to offer a recent and multi-sourced but concise literature review on the relevant related works in research fields such as the Internet of Things, Industry 4.0, Smart Cities, Cyber-physical Systems, and Smart Parking Systems providing its definitions and state-of-the-art. Finally, this work is inserted in the exposed context to express its relevance and placement in such study domains.

2.1 Internet of Things and machine-to-machine

The Internet of Things (IoT) is a concept that has been under media and science spotlights for the past decade. Although it can be considered relatively new for the scientific community, it is an old dream of automation technicians since the early 1980s. The first-ever internet-connected device was a soda vending machine placed in the Computer Science Department's of Carnegie Mellon University, it could remotely inform its stock and whether the beverage was cold or not. From that to the present date, the idea of an interconnected virtual world of devices spread, and the rapid scientific and industrial development of fields such as microcontrollers, power electronics, and wireless communications have been enabling increasingly complex applications [13].

This concept is, in simple terms, a technology for the promotion of internet-connected devices. These are designed to be simple enough to enable scalability, but complex to the point of being able to form a network with other devices, collect data, and occasionally make decisions. These study objects are called *things* and connected in a system of decentralized processing units that comprehend their environment and communicate [14].

Likewise, machine-to-machine is, today, an increasingly relevant field for research and development projects. Its most general definition is "communication between devices with minimum or no human intervention at any phase", i.e., reconfiguration, operation, and maintenance. M2M implementation makes many technologies possible, for example, decentralized decision-making, real-time data acquisition/broadcast, smart homes, cities, and mobility. In fact, this is a core technology for the development of Cyber-physical Systems (CPS), specially in the context of the Industry 4.0, as it provides the essential communication easiness. Both M2M and IoT hold great importance in the future of Industry 4.0 and automation systems [15], [16].

It is noteworthy that the IoT concept usually denotes a collection and sharing of data using a cloud-connected device or database as controller and calculator, while M2M stands for direct communication between independent devices that belong to the same system. The first is typically more scalable but deal with highly diversified types of data. M2M, in its turn, is usually implemented with standardized communication protocols and serves as a tool for the deployment of enclosed Smart Systems. Each of these two technologies have its particular fields of applications, but projects are often labeled both as IoT and M2M applications at the same time [16].

This work is framed both in the concept and the technology once it operates with non-human communication between devices, as described in subsection 3.1.1, i.e., machine-to-machine. Also, as said in subsection 1.1, the project consists of the automation of an everyday object, intending to transform its use from an ordinary passive bicycle parking spot to an active and intelligent parking system, allying multidisciplinary internet-based technologies and concepts. Hence, it is framed as an IoT appliance as well.

2.1.1 Cyber-physical Systems (CPS)

CPS are structures with linked hardware and software. Its purpose is to combine sensors, actuators, and embedded electronics aiming to add new capabilities to physical objects. These systems are superior to regular control because of the technologies applied to the combination of physical and cyber elements, which provides enhancements such as reconfigurability, intelligence, communication, and scalability. Namely, the system comprises physical objects to be controlled and virtual processing to enable analysis and decision-making. The first collects data and sends for processing, while the second feeds it with control signals and runs control algorithms. CPS can be understood as intersecting closed-loop of physical and virtual environments [17]–[19].

The *smart objects* have different levels of intelligence for each intended application, as it can sense its surroundings, process data, engage routines, or even trigger actions on its environment. In fact, for the past years, the fast advances in embedded systems have been sustaining the spread of cyber-physical appliances for ambient assisted living, entertainment, logistics optimization, energy management, and industrial automation [1].

As said by [13], the CPS implementation trend is easily recognizable and is expected to keep on rising, as now the Internet interconnects a large volume of heterogeneous devices. For example, the spread of wireless sensor networks (WSNs), which consists of many sensor nodes capable of collecting physical world data and wirelessly dispatching it so this information is used for control or monitoring purposes. Such technologies are being increasingly absorbed by the industry.

Figure 2.1 shows a schematic of the academic standard CPS' layers. First, the *things*: the physical objects of the system, for example, devices, machines, and environments. Next, in the first CPS' stage, there are sensors, actuators, and hardware. Then, the internet gateways, where the system acquires data and connects to the cyber world. In the third stage are pre-processing and connection between the physical and cyber world (also known as "Edge"). And finally, the last step consists of a data center and cloud computing [20]. This project focuses on the implementation of the first two stages.

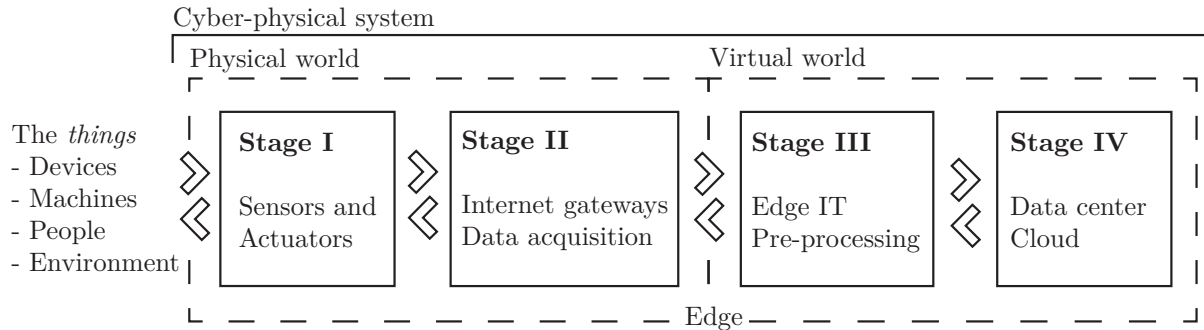


Figure 2.1: Four stages of Cyber-physical Systems (CPS).
Adapted from [20].

2.1.2 Industry 4.0

A trend of research in many interdisciplinary fields consists of the association of technologies and concepts such as IoT, M2M, CPS, and Big Data analysis aiming to build enough structure to provide new capabilities to smart production systems. These previously unseen features include more refined integration between systems, more flexible control, remote algorithm-based computing, and broader communication. Industry 4.0, or fourth Industrial Revolution, is based on the implementation of such topics intending to enable a more autonomous, more efficient, and customized productive process. Notably, this is the first-ever so-called Industrial Revolution studied *a priori* and not observed as a past event. This is an inherently interdisciplinary topic, involving many study domains such as industrial, production, electrical, and mechanical engineering, along with information technologies, business, and management [21], [22].

The remarkable Industrial Revolutions brought many paradigm changes to the industrial environment. These new ideas led to the complete transformation of equipment, processes, and factories organization in the course of decades. As seen in figure 2.2, the first revolution introduced steam-powered machines; the second was distinguished by electromechanical equipment on assembly lines; the third, in its turn, was characterized by automation and connection of computers. Finally, Industry 4.0 is a contemporaneous concept, and so there is no clear consensus on an universally accepted

definition [21]. This document considers the definition addressed in the previous paragraphs of this section, based on [23].

The fourth Industrial Revolution, as said, was distinctly early identified and studied. The German Federal Ministry of Education and Research published in 2013 its recommended practices for the deployment of projects included in this topic [23]. This publication became a landmark for other governments, and the German inclination to sustain the quick development of Industry 4.0's technologies along with its fast positive results fueled a wave of research and industrial systems deployment both inside and outside the country sustained to date [22], [24].

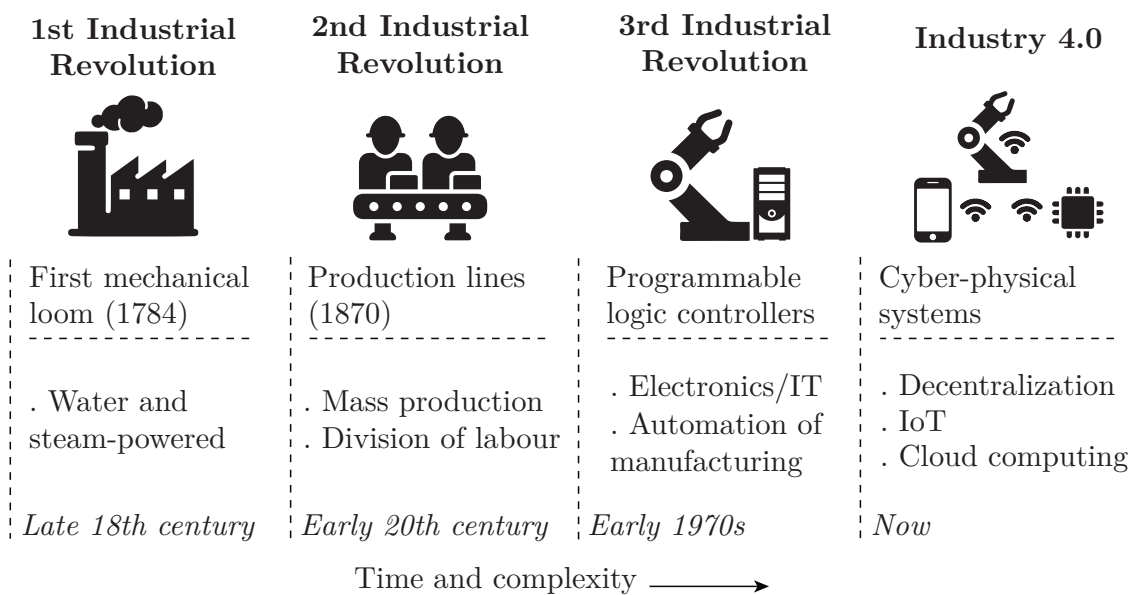


Figure 2.2: Industrial revolutions and its starting points. Adapted from [23].

2.2 Smart cities

"Smart cities" is a very general and contemporary topic. Researchers have been developing the first studies on this subject for the past decade, and so its definitions still cannot be considered consensus on the scientific community. However, several authors contributed, in many ways, by publishing their particular characterizations. Taking this into account, Smart Cities are defined by [25] and [26] as urban systems that use automation and information technologies intending to build better, more stable, efficient, and resourceful infrastructure and management for cities and citizens. Hence, public services have better rates on its interactiveness, accessibility, and efficiency.

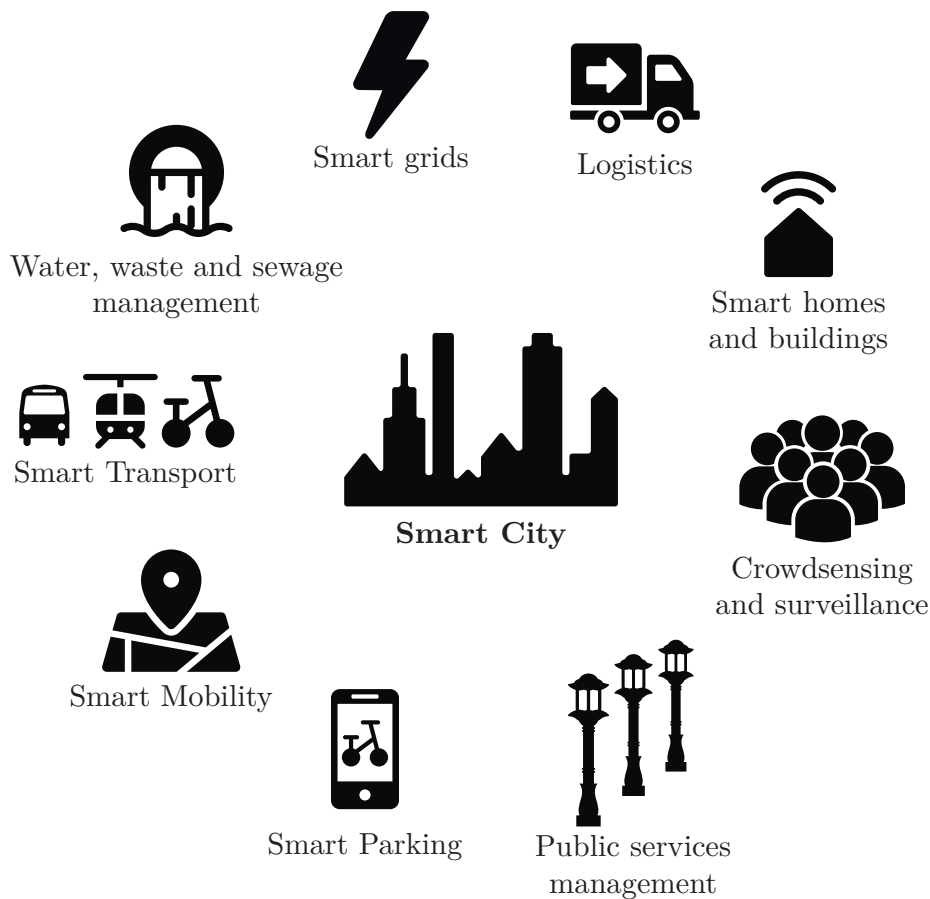


Figure 2.3: Smart city sample model.
Adapted from [27].

For that matter, as shown in figure 2.3, deployers of Smart Cities must deal with naturally complex and multidisciplinary concepts. In order to establish a reliable and comprehensive system, they often must implement state-of-the-art communications, data management, and embedded electronics technologies. Paraphrasing [27], the demand for brand-new, enhanced solutions is high given factors like population constant change, limited financial resources, and political inertia. Technologies such as Internet of Things, agent-based CPS, and similars drive smart cities planning, which enable better flow of goods, enhance vehicle traffic and allow the deployment of more environmentally friendly systems. A Smart City is defined by [6] as an innovative city that has reached a point of broad integration, with ICT and other means combined to improve quality of life, and efficiency of urban operations and services.

As a naturally omnibus concept, Smart Cities invariably embrace many different types of study objects (from devices to environments), so defining architectures, standards, and critical requirements is vital. Along with that, the large-scale range of elements (sensors, actuators, wireless technologies, internet connection, and processors) present on such systems demands uniformity of data and hierarchization to allow data analysis. Also, cybersecurity is seen as a critical component of this context, aiming to ensure responsible gathering, storage, and treatment of data [26], [27].

2.3 Smart parking systems

This section intends to approach the current works on smart parking systems demand and present scope of research. Its focus is to search for techniques, topologies, and implementation steps presently used for the deployment of such systems aiming to build a general knowledge on the field's latest academic and commercial solutions.

The exploration of the smart parking systems field of research is in its first steps. As an advent of the IoT technologies, CPS are a state-of-the-art concept. Naturally, there is a reduced number of available studies that include all of this work's objectives and solutions. Hence, specific approaches such as the physical implementation of bicycle parking tend to count on fewer preliminary studies material. The focus of the recently published articles is on the virtual control and communication of the smart parking systems, both already deployed in this project. Still, there are isolated studies regarding steps of the physical stage of implementation.

Contrarily, the commercial market presents many exciting features that could be inspirational on the project of smart parking systems. Naturally, companies do not share specific information around their devices and protect them with intellectual property regulation laws, which can complicate the replication. Nevertheless, it is valid to observe the solutions these organizations utilize to inspire a new solution design.

2.3.1 Demand for intelligent systems

Intelligent parking systems emerge from both the actual and forthcoming demand for more efficient parking. Traffic congestion is recognizably one of the main problems in the management of major cities since the increasing number of gridlocks and limited number of parking spots causes fuel over-consumption, time delays, and driver fatigue. Moreover, it is also responsible for problems like higher pollution rates and invariably prompt socio-economic problems, especially in developing countries, as discussed in the following subsections [28], [29].

Pollution

Countries have been witnessing a continuous addition of cars to their total fleet [30]. The rapid growth, shown by table 2.1, in vehicle numbers appears in industrialized countries but is even stronger in developing countries, such as Brazil, China, and India. These nations present economic growth, rising incomes, and intense urbanization for the last few decades [31]. Along with that, these nations are notably gigantic. Summed, they count 2.93 billion inhabitants, representing 39% of the world's population [32]. Therefore, the steady enhancement in their quality of life rates embraces a vast amount of people.

Road transport represented, in 2012, 17% of global CO₂ emissions. Many studies are in development to establish a pollution prospect, and predictions point to a peak in carbon emissions by 2020 and a gradual decrease until 2050 in a business-as-usual scenario. Although the number of cars should grow until 2050, developed countries will significantly decrease their carbon footprint in the next years. However, studies suggest that developing countries will gradually increase their overall greenhouse gas (GHG) pollution in parallel to a fast reduction in the gap between their and developed countries energy usage. Finally, even with the expected shrinkage in global emissions, only scenarios with great mitigation efforts seem to meet long-term sustainability goals [33].

Table 2.1: Vehicles per thousand people in 2007 and 2017 [30].

Country or region	Vehicles per thousand people		
	2007	2017	Increase (%)
<i>India</i>	12.3	36.3	195
Africa	24.0	38.4	60.0
Asia, Far East	63.4	112	76.7
Asia, Middle East	101	150	48.5
<i>China</i>	30.3	156	415
Central and South America	128	176	37.5
<i>Brazil</i>	132	210	59.1
Europe, East	271	373	37.6
Europe, West	588	612	4.08
USA	845	831	-1.66

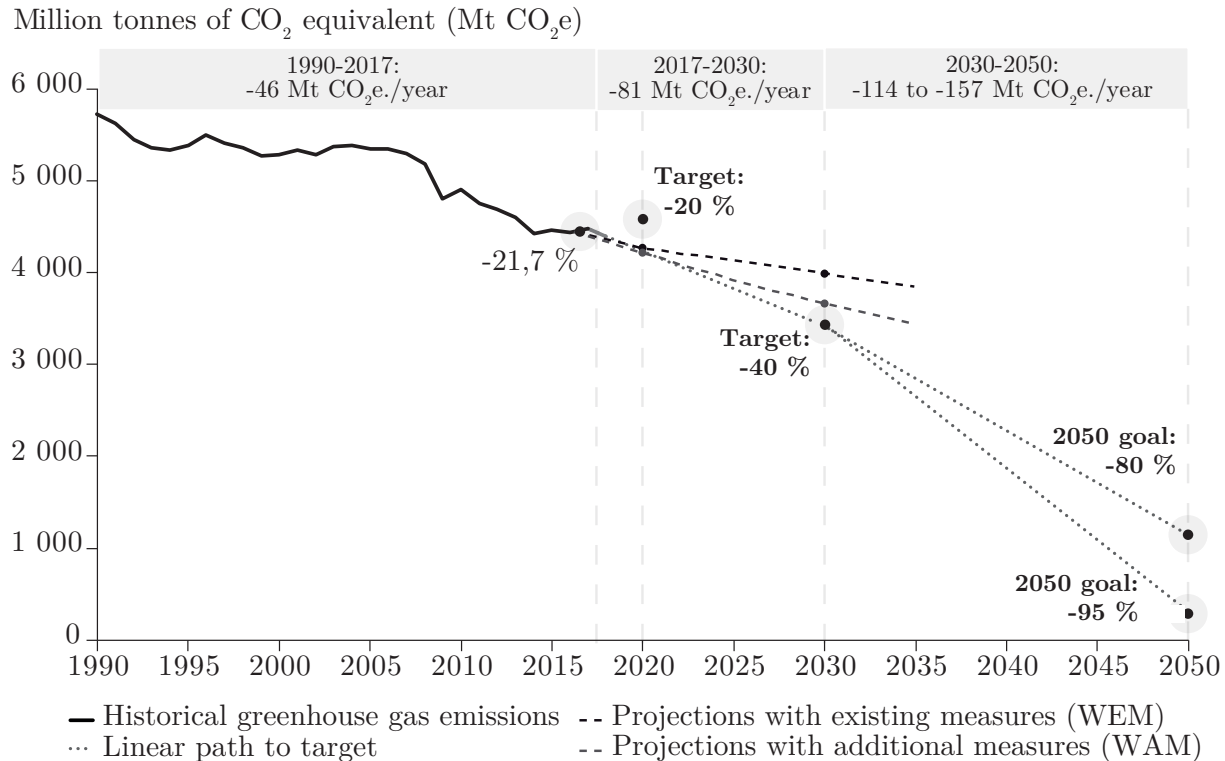


Figure 2.4: Europe’s GHG emissions trend projection and targets [35].

Europe leads the GHG mitigation efforts worldwide [34]. The European Union (EU) has been setting targets since the Paris Agreement in the 1990s, aiming to achieve progressively lower rates of emission (see figure 2.4). From 1990 to 2020, the EU set a target of mitigating 46 Mt CO₂ e. per year, accomplished early, in 2012; from 2020 to 2030, the purpose is to drop by 81 Mt CO₂ e. per year, and from that until 2050 the goal is to reduce 85 to 90% of global emissions since the year of 1990. Predictions, however, indicate that the EU nations will miss the next targets [35].

In 1997, [36] estimated that finding a parking spot in Westwood, a small business district of Los Angeles, took 730 tons of CO₂ equivalent, 95 thousand hours, and 178 thousand liters of gasoline every year. A 2018 study on 22 downtown areas of the United States suggested that cruising for parking took from 8 to 74%, averaging 34%, of the car’s running time [37]. Smart parking appear in this context as a solution to improve efficiency, lower fuel consumption, and mitigate GHG emissions significantly in more efficient spots management and, also, on projects to encourage alternative transportation [28], [29].

Gridlocks

The progressive increase in the total vehicles fleet (see table 2.1) incites problems such as waste of time and more pollution, less available parking spots, and higher car accident rates. Globally, road traffic crash rates remain a crucial problem, and projections suggest little change in the foreseeable future [6]. As compiled by [38], an enormous amount of resources is spent yearly on car congestion-related incidents: 88 billion dollars in the United States, almost 7 billion pounds in the United Kingdom, and close to 3 billion euros in Germany only in 2019.

Table 2.2 shows the world’s most congested cities. Notably, all cities cited on the table are from developing countries, most of them in South America. Governments, usually from high-income countries, are moving in the direction of building sufficient infrastructure to overcome the traffic problem. The Spanish city of Santander monitors real-time status of parking spaces, air pollution, and traffic conditions. Los Angeles, in the US, uses Big Data analysis to enable second-by-second actions on traffic congestion [6].

However, measures to mitigate gridlocks are not always effective. A study from 2011 affirm that adding extra lanes to existing streets is unlikely to relieve congestion [39]. A viable solution, then, is to diversify transport means. Examples are seen in NYC, Seattle, and London; where policies were successfully deployed aiming to influence citizens into choosing cycling to decrease the number of cars in the streets [40]–[42]. Bicycles naturally occupy less space than cars. Hence, each person that transitions from driver to cyclist saves significant space in a crowded urban environment. Smart parking systems fit in this scenario as a tool for a more organized and coordinated transition from cars to bicycles.

Table 2.2: The five most congested cities in the world [38].

Urban area	Country	Region	HSCpD	2018-2019
Bogotá	Colombia	South America	191	+3%
Rio de Janeiro	Brazil	South America	190	-5%
Mexico City	Mexico	North America	158	+2%
Istanbul	Turkey	Asia	153	+6%
São Paulo	Brazil	South America	152	+5%

2.3.2 Academic solutions

Many academic studies consider the bike-sharing systems (BSS), where a company offers multiple vehicles and users rent it for a limited period. Although these purposes are still useful for the deployment of a purely smart parking system, it is relevant to highlight that this project must offer a stand-alone solution, unlike the BSS where the bicycle's structures are designed to help the physical assets in locking and reservation processes.

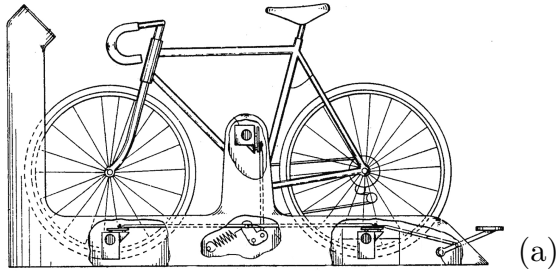
That said, analyzing the available solutions, some interesting ideas come up. The paper [43] defines an architecture for the deployment of BSS through many different IoT technologies. GPS, sensors, RFID tags, and smart locks are proposed as physical assets for this implementation. [44] and [45] present another compelling proposition for the physical sensing stage, the use of a wireless visual-aided vehicle presence sensor. These projects suggest sensing the spot availability through artificial intelligence image data processing, which hugely increases complexity but offers less maintenance and adds more precision.

[46], [47], and [48] discuss reservation-based parking systems, bringing a closer approach to this project specifically. The first, [46], documents an appliance of a presence sensor as an edge device that detects occupied spots and communicates with the system's controllers through WiFi. [47] implements the previous edge device in a spot reservation app for smartphones in an Arduino-based low-cost system. Finally, [48] simulates one of these systems for cars in downtown Los Angeles and reached great results in traffic reduction. This paper's architecture consists of a proposition of a paid web server reservation system with ZigBee sensors, wireless radio, and Bluetooth communication. Interestingly, this deployment confirms the user's identity through its cellphone Bluetooth in loco.

2.3.3 Commercial options

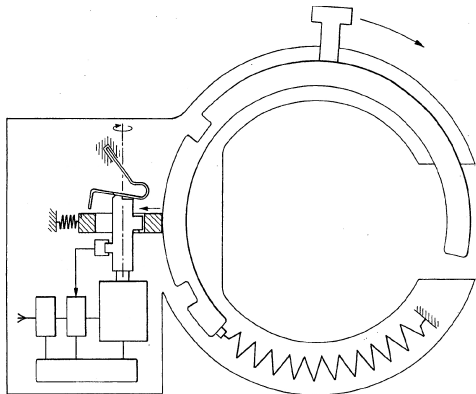
As previously addressed, the commercial field has diverse developed solutions for the physical handling of a bicycle's parking. In the last few years, startups offering BSS thrived in major cities. Therefore, the exploration of new material and methods rapidly

United States Patent [19] Patent Number: 5,278,538
Ainsworth et al. [45] Date of Patent: Jan. 11, 1994



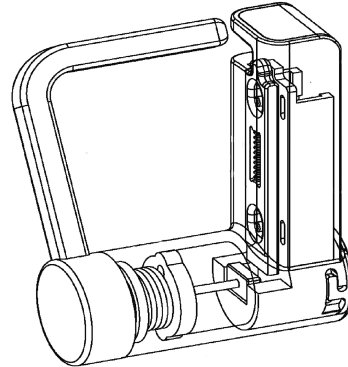
EP1760232B1
EP 1 760 232 B1

EUROPÄISCHE PATENTSCHRIFT



(c)

(12) United States Patent (10) Patent No.: US 9,679,429 B2
Duncan et al. (45) Date of Patent: Jun. 13, 2017

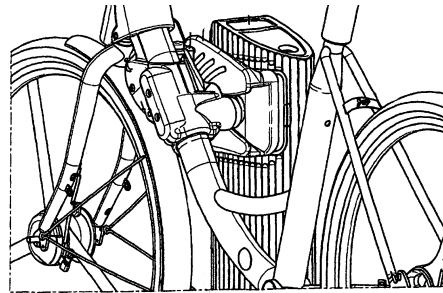


(b)



EP001820722B2
(11) EP 1 820 722 B2

NOUVEAU FASCICULE DE BREVET EUROPEEN
Après la procédure d'opposition



(d)

Figure 2.5: Patents on bicycle's locking mechanisms [49]–[52].

grew. For the successful deployment of companies that provide bike-sharing services, characteristics such as a portability, fastness, reliability, and affordability are mandatory for the lockers. So, in the past few years, many patents on bicycle lockers were registered, and the improvement in the mentioned characteristics is clear.

1994's, figure 2.5 (a), bicycle security system is large, mostly mechanic, and occupies a large volume of physical space [49]. Differently, patent (b) is condensed and presents automated features [50]. It must be locked by hand but unlocks via a wireless signal, an improvement towards enhanced automated lockers. Subfigure (c) illustrates another possibility for the same appliances [51]. It is also based on manually locking but can be unlocked wirelessly using an internal electromagnetic actuator. Alternative (d) is a socket, where a bicycle's structure fixed mechanical part fits, enabling features like charging electric bicycles' batteries and ensure locking [52].

The overall majority of the commercial patents in use locks the bicycles by passing a piston through the wheel's rim like the examples shown in subfigures (a), (b), and (c). Although a few options propose locking the bicycles with the frame structure, like in subfigure (d), or wrapping it with a resistant material. Noting that bicycles' wheels are largely standardized while its frame structures are not, this project focuses on the first option to universalize its possible users.

As said, the lockers are evolving from purely mechanical locking structures to more automated ones. The intent of this project is to go a step further, proposing ideas for the full automation where the user does not need to interact with the locker at all other than fitting the bike in the spot.

Chapter 3

Study object and system's architecture

The following section addresses a closer observation of the system. At first, there is a description of the study object intending to explicit its overall characteristics, possible features, and general operation. Following, there is a section for the explanation of fundamental concepts of this work. Then, the final segment discusses Multi-agent Systems (MAS), integration between cyber and physical elements, and the operation and appliances of linear solenoids.

3.1 Description of the study object

The studied object of this work is the transformation of a standard bicycle parking spot, seen in figure 3.1, from an ordinary passive object into an intelligent and automated Cyber-physical System. This conversion should grant characteristics such as secure locking, parking spot reservation, control and management of spots under specific conditions, pinpoint available spots for users, and promotion of alternative transportation means. It is expected that this system suit applications for more heterogeneous vehicles with only hardware modifications and minor software adjustments.

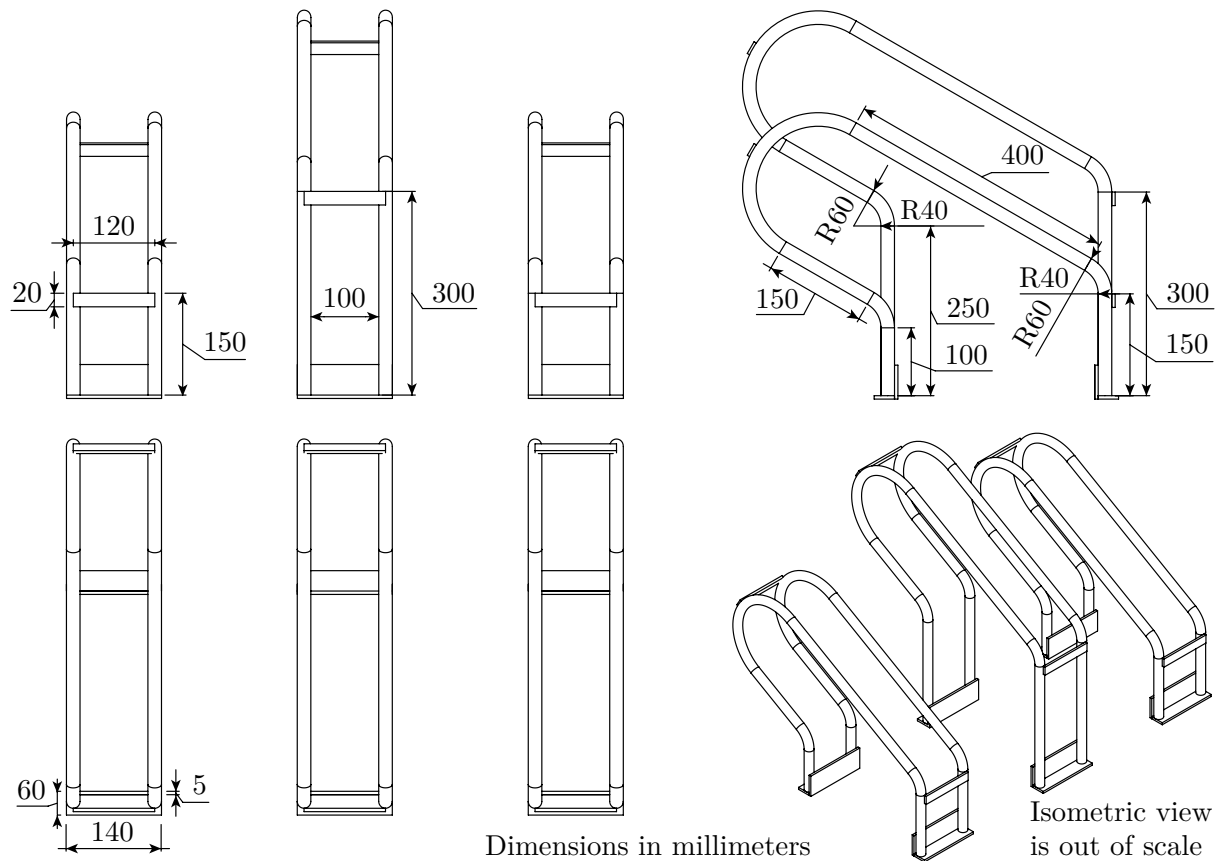


Figure 3.1: Frontal, side, superior and isometric views of the studied parking spot.

3.1.1 Overview, characteristics, and features

This project is part of an on-going partnership between IPB and UTFPR for the deployment of smart parking systems. Hence, the architecture of the virtual section of the work is already defined as published by [11]. Figure 3.2 illustrates the steps for the development of this smart parking system. Users would download a smartphone application for the interface with the system; driver and spot agents, in its turn, interact through an established MAS architecture; finally, the last step is the physical stage of the system, the focus of this work.

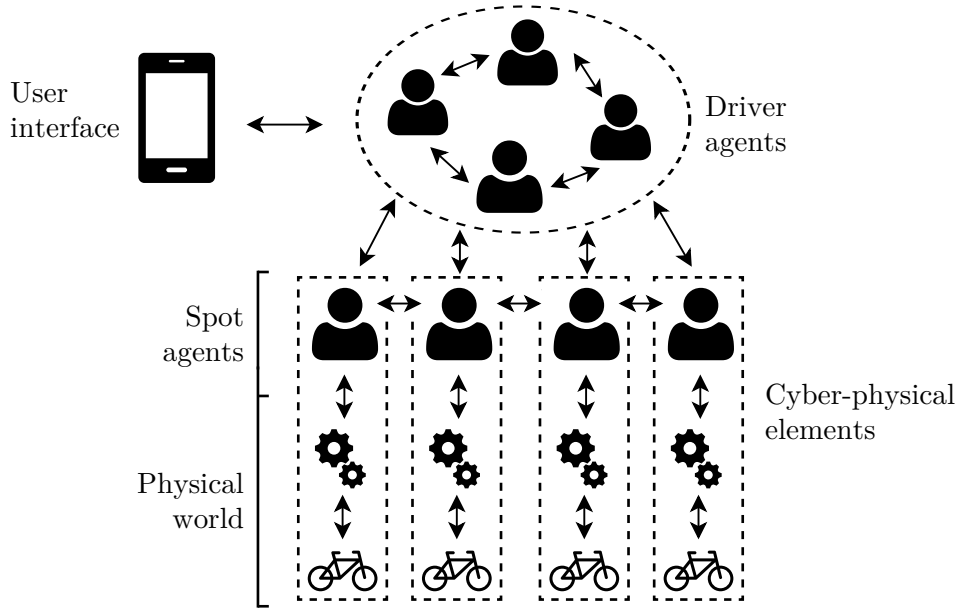


Figure 3.2: Generic scheme of an agent-based smart parking architecture [11].

The settled plan demands a well-defined organizational structure for the driver agents proper operation on management and negotiations. For this, there is a need for technologies such as cloud computing and machine learning to be added in order to ensure flexible, robust, and adaptative handling of dynamic tasks. The architecture embodies a micro-society of intelligent, autonomous, and cooperative agents to form a MAS. Each agent represents a system's object, namely parking spots and drivers. The overall operation of the system consists of the interaction between the several drivers and spot agents, supported by negotiation strategies. The spot agent represents the physical spot and acts as a bridge of the cyber and physical components.

This project focus on the conceptualization for the physical implementation of the smart parking for bicycles in question. Hence, it concentrates on the "physical world" area of figure 3.2, from the connection bridge between the Spot Agent and the physical asset controllers to the final lockers for the secure placement of the vehicles. Thus, the project comprises the proper connection of wireless data transmission, the implementation of a microcontroller routine for the control of spots, and the lockers mechanical design.

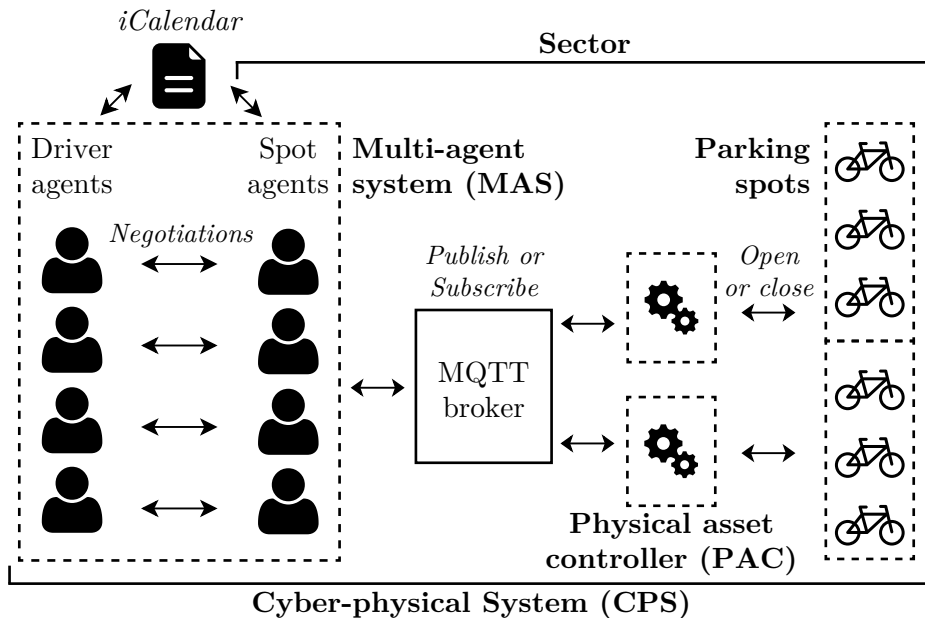


Figure 3.3: Defined structure for the agent-based CPS smart parking.

Figure 3.3 shows the defined structure for the deployment of this agent-based CPS applied to a smart parking system. The first stage consists of an established MAS architecture, where driver and spot agents negotiate with one another following holonic principles to define the most suitable path to the parking management. Then, the iCalendar file stores the reservation information after a round of negotiations between the driver and spot agents. In [11] this subject is addressed in more detail. Next, the MQTT broker connects spot agents to the physical asset controllers (PAC). The first is implemented in a cloud-connected device, and the second is a programmable control device, responsible for the management of the system’s physical elements. The PAC is, then, connected by wires to the sensors and actuators of the spots. The sensor is a pull-up connected pushbutton acting as a debounced interruption source for the microcontroller to verify the presence of a bicycle on the spot. Also, there is an electromagnetic locker, the system’s actuator, which follows the PAC’s commands. Section 4.1 addresses its design options, validation, and operation characteristics.

3.1.2 System operation

This subsection addresses the choices related to materials and the project's schematics. The decisions demand the adoption of some project parameters in order to obey the project's architecture defined above and search for low-cost options and materials' capabilities fitting this project.

Note, in figure 3.3, it is defined that each physical asset controller manages three parking spots to enable scalability. This happens as a result of the GPIOs availability in the chosen microcontroller model, ESP8266 development board. Each PAC could control more or fewer parking spots at once, depending on the number of available ports for the connection of peripherals and capacity of the chosen microcontroller unit. Another approach is the alternative of using ESP8266 series module, which would be capable of controlling a single spot at a time. This work uses the three spots management scheme, but minor software adaptations allow the conversion to single-spot control as well. Section 4 discusses further details on the materials' characteristics, programming, and placement on the system.

Figure 3.4 shows the information flow through the different elements of the system. As said, this project focuses on the physical implementation of the smart parking system. Hence, the components presented on the cyber stage were defined previously by the project's past deployers. This stage consists of two Raspberry Pi microprocessors. One for the implementation of the agent-based that emulates cloud computing MPU that hosts the negotiations forming the basis of the system's operation. The other is the MQTT Broker, which manages the whole system's data flow.

Both cloud-computing and PAC devices wirelessly connect to the MQTT Broker. This form a bridge for the information sharing between these devices, which enables the chain of command structuring from spot agents to the PAC, and data exchange from decentralized sensors to the main program running in the Cloud. In its turn, the physical devices, such as the actuator's electronic drivers and sensors of each spot, connect to the PAC through wires to exchange command and interruption signals.

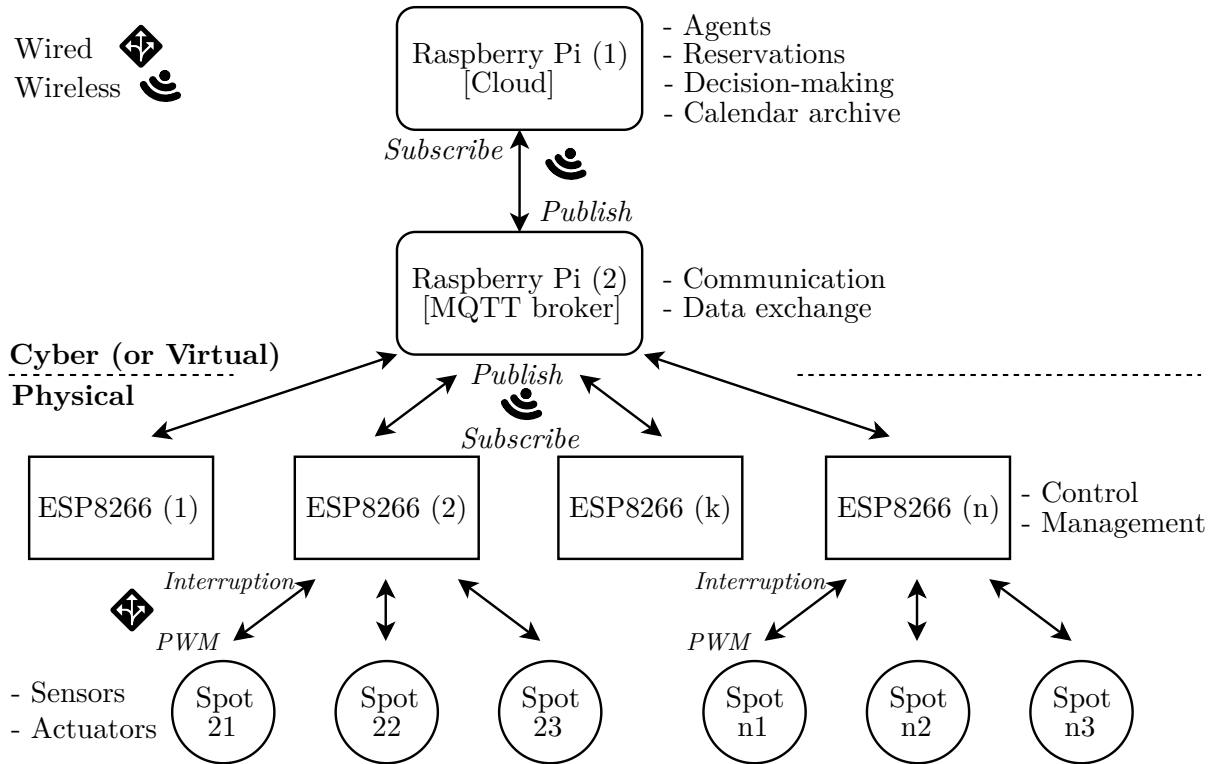


Figure 3.4: Scheme of the system's structure of control.

3.2 Architecture

This subsection addresses the system's conceptual basis. Each operation stage comprises a series of techniques combined to deliver proper functioning. So, it opens from a broad perspective with the Multi-agent System architecture, then approaches the applied communication protocol (MQTT), and concludes with an explanation around the electromechanical concepts of the lockers' designs. The previously discussed project's structure serves as a starting point for this discussion.

In figure 3.5 overview, the project structure shows the system's organization. MPU 1 and 2 are Raspberry Pi, previously addressed and deployed. MPU 1 comprises agents routines, reservations, decision-making, and calendar management tasks, while MPU 2 manages the MQTT links acting as the system's broker. The microcontrollers identified with letters ("MCU k"), in turn, are ESP8266 designated as physical asset controllers, and they control all physical features of the system. The electronic drivers are referred to as

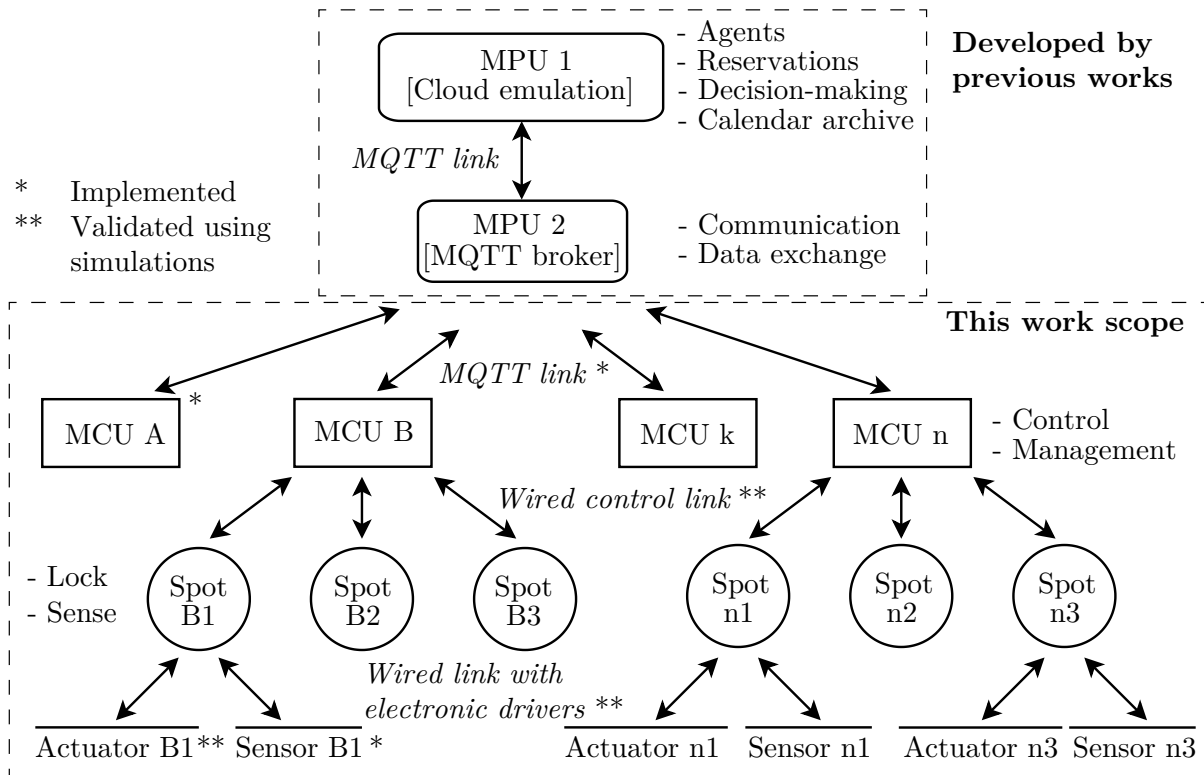


Figure 3.5: Scheme of the project scope.

"Spot k1", and they control their respective sensors and actuators. This work scope begins with the MQTT link between broker and MCU k and ends proposing a few implementation options for the sensors and actuators. This work intends to develop simulations to validate various recommendations of actuator and electronic driver options, as well as a simple and reliable sensor configuration.

3.2.1 Multi-agent Systems (MAS)

MAS are a solution applied to CPS aiming to incorporate an intelligence level into its operation. An agent is an autonomous software component gifted with an interoperable interface, and it often interacts with other agents cooperatively or for self-interest [11]. Some characteristics define the agent as a singular and intelligent entity, which are autonomy, responsiveness, proactivity, social skills, and learning ability.

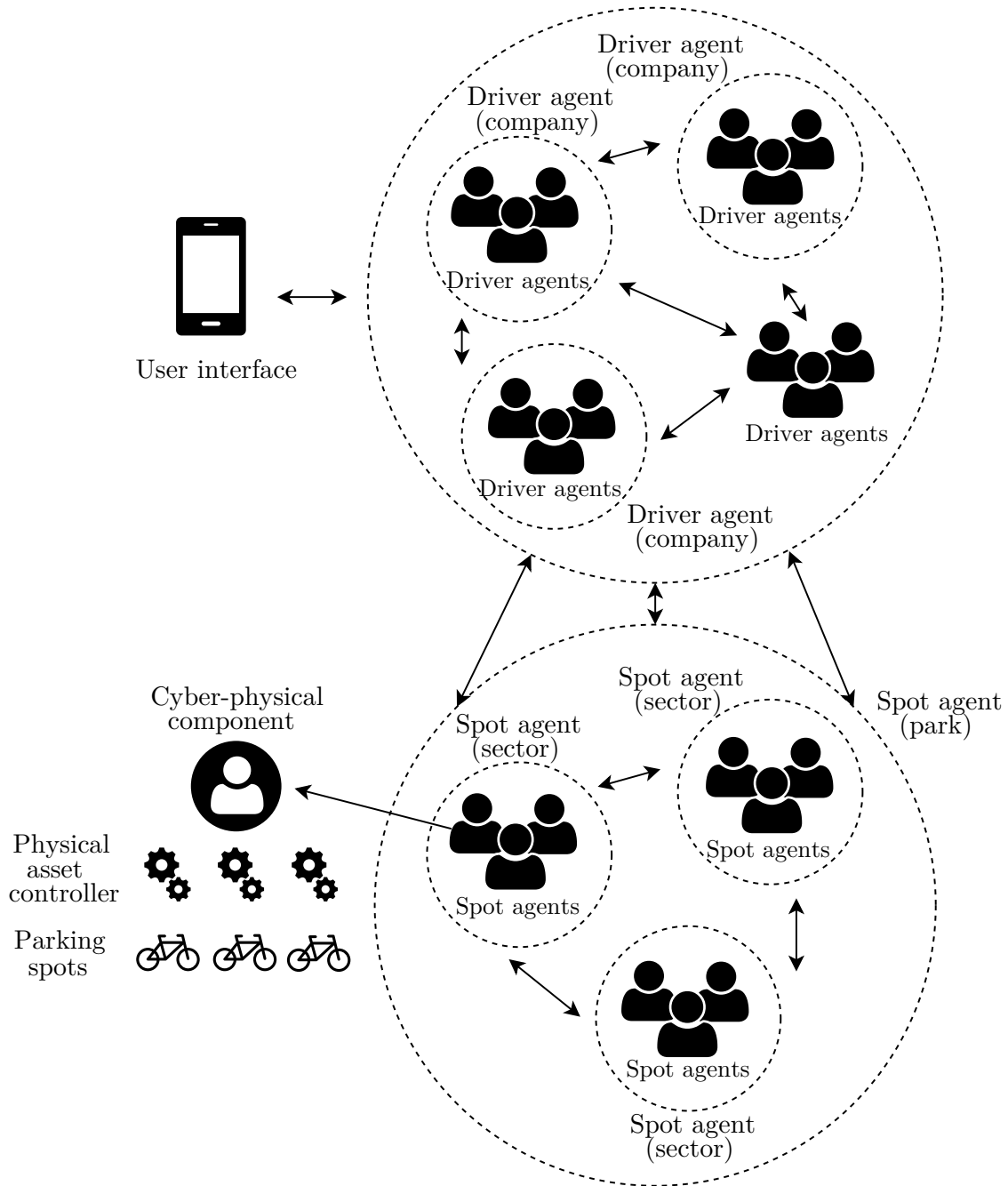


Figure 3.6: Smart parking MAS schematic. Adapted from [11].

Autonomy due to its ability to make decisions without human intervention and on its own. *Responsiveness* since it can perceive and respond to the world through sensors and actuators. It shall take the initiative to reach its objectives or anticipate environment changes, hence *proactive*. It cultivates *social skills* with other agents as a means to achieve its goals. Lastly, it *learns* and adapts to the changing environment and needs of the system's users [53].

Figure 3.6 illustrates the Multi-agent System architecture applied to smart parking systems. Ultimately, MAS are arrangements where multiple agents work and interact, reaching for personal or social goals. It typically adds advantages such as scalability, flexibility, robustness, responsiveness, and reusability. On the other hand, the implementation of this architecture is challenging since environments are continually transforming. Also, there is a need for constant conflict resolution and information exchange among the autonomous units, demanding a high level of cooperation [53].

3.2.2 Cyber-to-physical link

Several protocols exist for the deployment of CPS. The communication protocols supposed to best fit this system were approached and studied before this particular work started by other researchers. The considered interface practices were FIPA-ACL, MQTT, Modbus, and OPC UA. After a study and results comparison, the chosen protocol was MQTT. The investigation suggested that although the client-server approach provided faster response time, the publish-subscribe option presented the best performance. This conclusion comes from the fact that the second option has significantly better scalability characteristics [11].

Moreover, on-device approaches added both faster means of communication and better deterministic data acquisition in every case, but the need for scalability and monitoring play a more critical role in smart parking systems. Many low-level platforms cannot host software agents, so remote applications are frequently preferred. Also, the reusability feature provided by standardizing protocols is convenient in scalable systems. Hence, the chosen and implemented communication means was the MQTT protocol [11], [54].

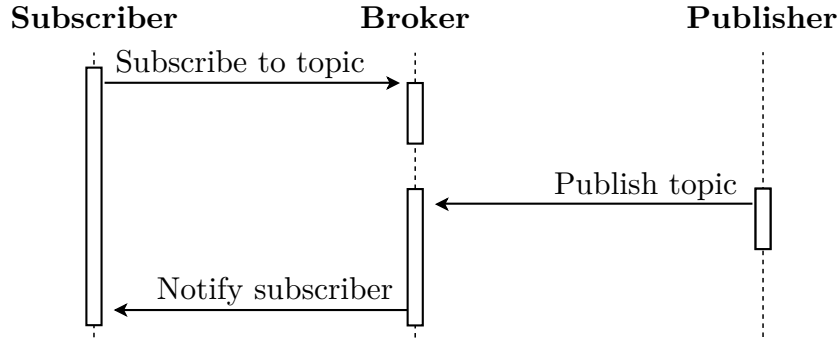


Figure 3.7: MQTT standard operation flux [54].

IBM invented and developed, in the late 1990s, the Message Queue Telemetry Transport (or MQTT) communication protocol. It was first conceived for the data correspondence between sensors in gas pipelines and orbiting satellites. As shown in figure 3.7, the MQTT practice supports asynchronous communication between parties since it decouples the source and receiver of a message in both space and time, so it happens to be scalable in unreliable environments. Despite its name, it does not work based on message queuing but with the publish-subscribe model instead [55].

The MQTT protocol implementation occurred in the link between the spot agents and the cyber-physical component, as illustrated in figure 3.6. Data format was standardized between the agents and the PAC according to table 3.1. It was done in a way that both the cloud-running agents and the physically present controller understand each other's commands and data. The agents can send `.char` messages like "arriving" or "departing" aiming to send commands to the electronic parking spot. After this first interaction, the agents address a second message to the PAC with a confirmation ID, to which the

Table 3.1: The sequence of standardized messages for MQTT communication.

Sender	Receiver	Message (.char)	Triggers
Agents	PAC	<i>arriving</i>	Verify spot, unlock, wait sensor confirmation
		<i>departing</i>	Unlock, wait sensor confirmation
		<i>ID</i>	Read ID, create a MQTT topic, publish
PAC	Agents	<i>error</i>	Error

PAC reads and responds. The PAC can also send an error notification in case of vacancy superposition, where the agents asked to free an already vacant spot, for example.

For the deployment of this communication mean, the Mosquitto message broker was installed and run in a Raspberry Pi device connected to the same network as the ESP8266 acting as the PAC. The tests were successful for all standardized messages from table 3.1 from the agents and all coded routines such as incoming user, departing user, and error messages were tested and triggered correctly, as addressed in subsection 5.6, following the flow shown in figure 3.8.

Basically, the deployed MQTT broker device works as an information mediator. In the first moment, devices subscribe to topics to receive data related to that subject. When a device provides data to the same topic, the broker diffuses the information to every device

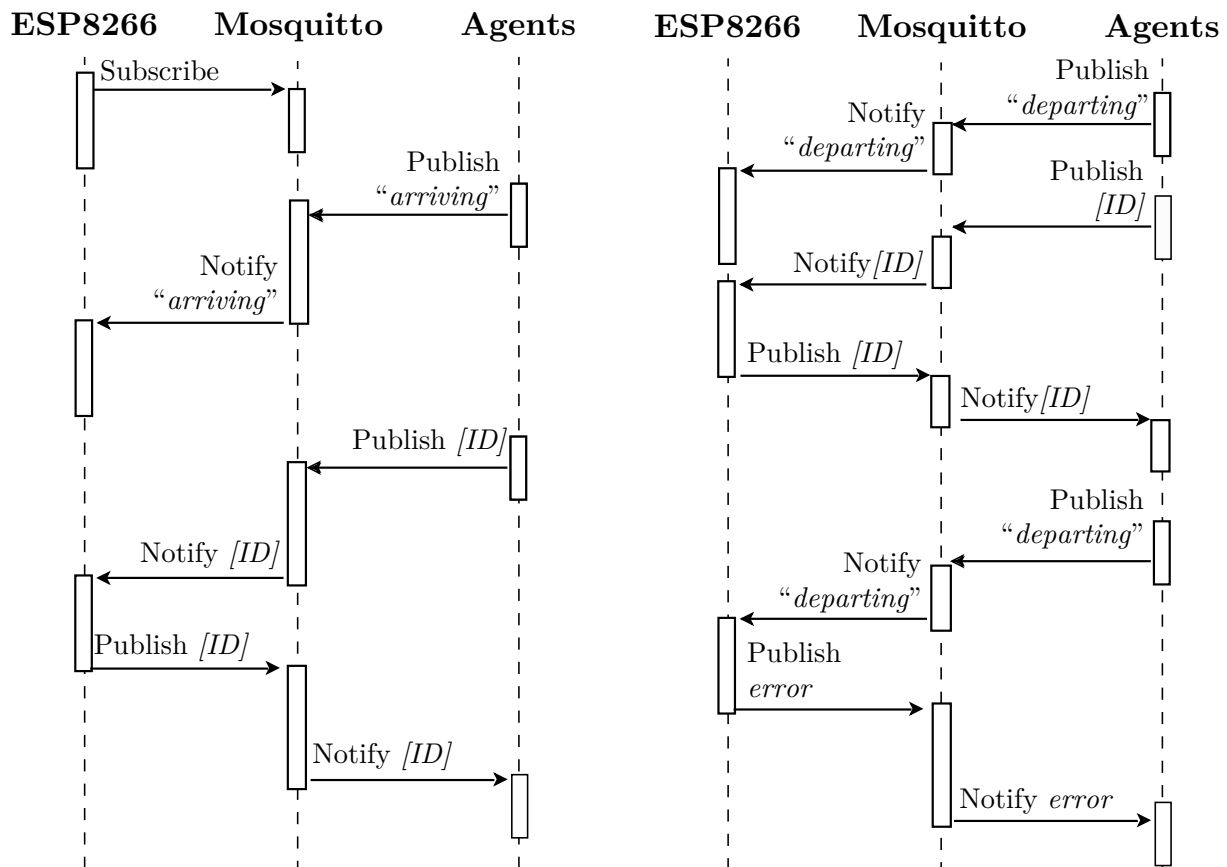


Figure 3.8: MQTT simplified tests scheme.

interested in it, i.e. previously subscribed to that topic. This way, as commented above, the information source (publisher) and receiver (subscriber) are fully decoupled, which enables higher control and security over data flux and storage.

3.2.3 Linear solenoid

The linear solenoid is a simple and widespread technology in many applications, such as home appliances, electronic doors, and automated lockers. It consists of using electric current as a driving source for linear movement. For this, as seen in figure 3.9, a conductor wire, usually copper or aluminum, is rolled into a winding to form a solenoid. A ferromagnetic piston is then concentrically coupled with the winding so that it acts as a path for the magnetic flux induced by the electrical field of the winding.

This project intends to use linear solenoids as a way to obtain a human-free linear movement inside the actuator mechanism aiming to trigger the mechanical locking operation. Section 4.1 presents further details on the subject. As it is an established technology, the mathematical formalization is known, and this work addresses the simulation without going into more theoretical details than the final formulas.

More information about this subject is available in [56]. That said, the following equations are vital to proper understanding and modeling a solenoid. The first is the result of the integral calculating the magnetic field in the P point (B_P) while the second derives from energy analysis to provide a force expression, also for the P point (F_P). Notably, both of the equations depend only on constants, current, and geometrical terms.

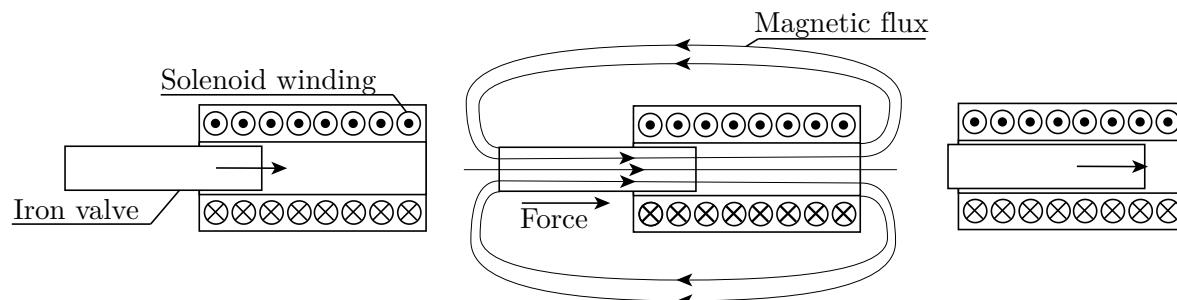
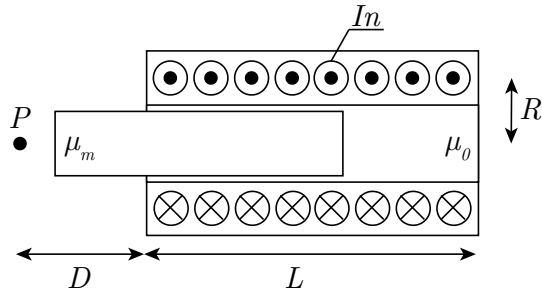


Figure 3.9: Generic solenoid scheme.



μ_m : magnetic permeability of the material
 μ_0 : magnetic permeability of the air
 I_n : electrical current
 D : distance from edge to the point of interest
 L : length of the winding
 R : radius of the winding
 P : point of interest

Figure 3.10: Terms of the solenoid equations.

Figure 3.10 illustrates all the terms of the equations and figure 4.5 is the graphical plot of these equations for better visualization.

$$B_P = \frac{\mu_0 \cdot I_n}{2} \cdot \left[\frac{D + L}{\sqrt{(D + L)^2 + R^2}} - \frac{D}{\sqrt{D^2 + R^2}} \right] \Rightarrow B_{edge} = \frac{\mu_0 \cdot I_n}{2} \cdot \frac{L}{\sqrt{L^2 + R^2}}$$

$$F_P = \frac{\pi \cdot R^2 \cdot B^2}{2 \cdot \mu_0} \cdot \left[\frac{\mu_m}{\mu_0} - 1 \right]$$

3.2.4 Electromagnets

Magnets are materials found in nature that naturally induce a magnetic field. Michael Faraday, a Britannic researcher, developed his law of induction in the early 19th century and, later on, James Clerk Maxwell expanded it, including magnetodynamics analysis. This law, known as Maxwell-Faraday Law of Induction, came to make explicit the intrinsic relation between electro and magnetic fields. Electromagnets mimic the cited magnetic materials with a magnetic field induced by an electric current flowing through a winding using the induction principle. The solenoid rearranges its poles temporarily forming a magnetic physical behavior. This way, electromagnets attract other opposite poles magnets or ferromagnetic materials.

This work uses the Maxwell-Faraday fields relation as an alternative to address the problem of locking the users bicycles in the actuator design context. The idea consists of an electromagnet that provides linear movement to a ferromagnetic piston with its

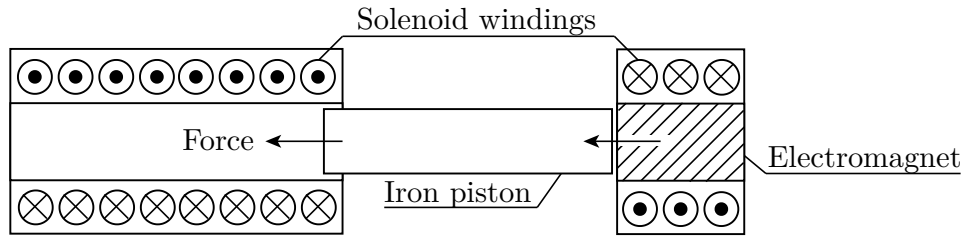


Figure 3.11: Unlocking forces with an electromagnet.

magnetic characteristics. To lock and unlock the system, current pass through a winding positioned around a fixed induced magnet, at the same time that a solenoid push or pulls the piston in the opposite direction. Figure 3.11 illustrates this process. To this end, the electronic driver has to be carefully designed. There are two options, enabling current flow in both directions or carefully timing the current supply so that the induced forces do not overlap. Subsection 4.1.5 discusses this topic in more detail.

Magnets' mathematical formulations are often intricate and time costly. Its inherent non-linear, frequently hysteresis-like, behavior hinders a simple approach and calculation. Also, it has a natural dynamic behavior. Again, [56] addresses in much more detail the mathematical formulation around electromagnetic induction, energy, and forces. However, this project chooses to develop its designs directly based on finite element simulations, which is a faster and safe alternative to design and test solutions to the addressed problems. Further, in section 4.1, this document discusses the electronic drivers, and mechanical performances of the actuator options.

Chapter 4

Physical implementation

This section discusses all the steps taken to implement the previously shown system's models, software, and hardware in four stages. It begins with the actuator propositions, compiling three mechanical projects, its structural and electronic driver design, and standard operation; following, the text approaches the chosen sensor; next, there is a discussion around the power supply to enable autonomy and facilitate scalability; finally, it ends with the explanation of overall functioning and connections of the microcontroller.

4.1 Actuator options

The actuator of a smart parking system consists of an automatic locker for the users' bicycles that receives a PWM signal from the controlling MCU. It must guarantee security, a low level of maintenance, easy scalability, while fitting the physical space of the structure. That said, this subsection focus on lockers design suggestions, illustrates the expected operation and the best range of applications of each one. A mechanism that converts linear to rotational movement is the first considered design. Next, the document considers a solenoid piston. Moreover, an electromagnetic locker is studied. And, at last, a stepper motor is also taken as an option. The final subsection approaches the electronic driver options as well.

4.1.1 Linear-to-rotational mechanism

The inspiration for this design is the mechanism present on many ballpoint pens. In simple terms, it converts the linear movement applied to one end into a rotational movement inside the mechanism's parts due to a peculiar path design. It consists of three parts here called the fixed path cylinder (FPC), the linear movement guide (LMG), and the rotational component (RC).

The FPC is a shell with a distinctive design to induce spinning on the rotational component to lock and unlock it in every 90 degrees turn. In its turn, the LMG conducts the rotational component back and forth, and also promotes the rotational movement along with the FPC. Figure 4.1 illustrates the rotational component, the linear movement guide, and the fixed path cylinder designs, respectively.

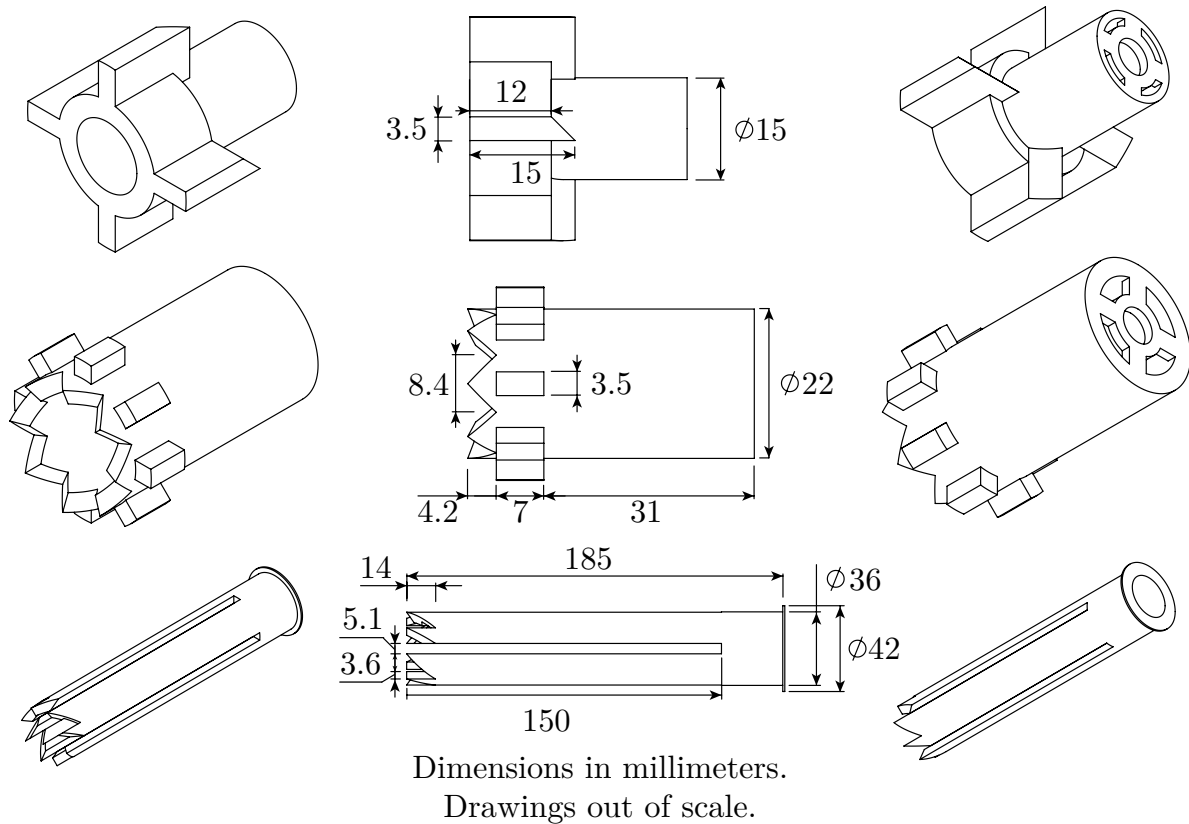


Figure 4.1: Views of the three parts of the mechanism.

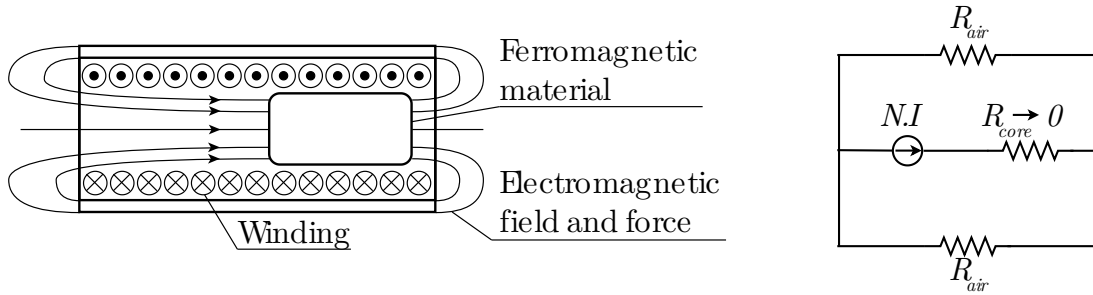


Figure 4.2: Simplified magnetic field and circuit design.

A winding is placed on the end of the FPC to provide a magnetic force from an induced field and automatize the system. The LMG is pushed and pulled by the ferromagnetic material subjected to the winding's magnetic force. An alternative to simplify the project is to attempt merging the winding and FPC's parts. Figure 4.2 illustrate the descriptions above and the resultant magnetic circuit. R denotes magnetic reluctance; N , the number of turns of the winding; and, I , the current amplitude. The piston's ferromagnetic reluctance is negligible because of its inherent high magnetic permeability values compared to that of the air.

Figure 4.3 shows the three key positions of the mechanism. At first, all parts are at the beginning of the path. The guide component moves linearly to push the rotational forward and, with the fixed cylinder's help, induce the 90 degrees revolution on this part. The symmetric characteristic of the paths designed inside the fixed cylinder ensures that when the guide moves, it locks and unlocks repeatedly. Note that the rotational piece must be, at all times, subjected to a force in the opposite direction of the LMG's movement.

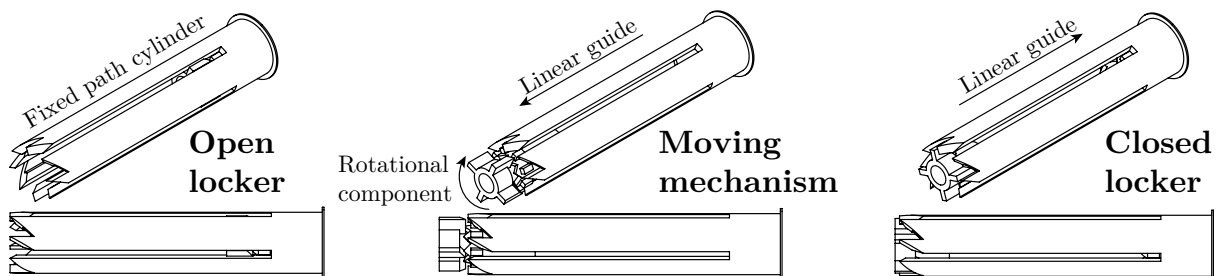


Figure 4.3: Isometric and lateral view of the mechanism's three key positions.

Placing

The mechanism's design must point upwards, in a way that the required contrary force acting in the system is the parts' weight itself. This way, there is no need for spring or elastic parts addition. The FPC points up, guiding the movement of the LMG and RC against the gravitational pull. Placed below the FPC is a winding consisting of a fixed ferromagnetic shell, that provides a low reluctance path for the magnetic field, and a ferromagnetic piston that promotes the LMG and RC parts' movements.

Figure 4.4 shows the mechanism's open and closed physical states. The cylinder placed on top of the RC part will then rotate a simple gear system that converts the upwards linear movement to a lateral linear movement. The gears design can influence the movement range of the winding's piston, the amplitude of the force, and the robustness of the mechanical locking. The ratio between the rays confer these characteristics and is an essential piece of the project.

Alternatively, horizontal placing is possible for this project. This way, the mechanical locking is provided by the mechanism's design itself. On the other hand, this choice requires a greater horizontal extent and would only be viable with major design changes of the available parking spots. The project would also demand the addition of an elastic part (spring or rubber band, for example) to ensure the proper RC's rotational movement. Due to these factors, this placing option will not be further considered in this work.

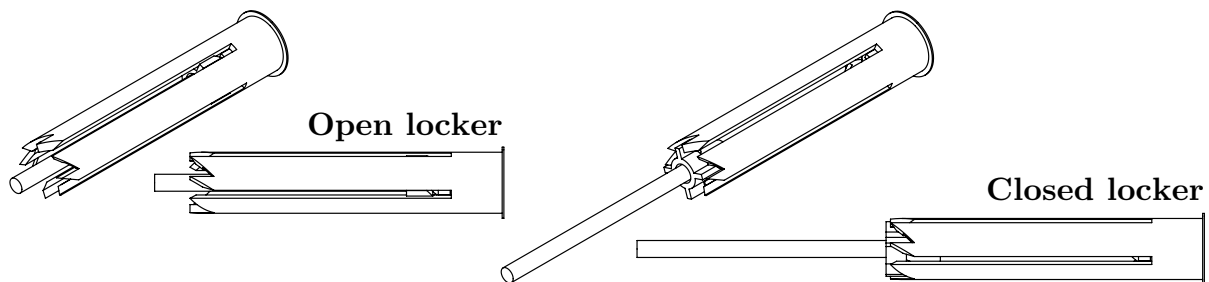


Figure 4.4: Isometric and lateral view of the open and closed mechanism.

For the physical validation of this project, a few mathematical formulations will follow. Taking into account the previously listed equations, in subsection 3.2.3, it is possible to determine the behavior of the magnetic field flux density and the force that will push the ferromagnetic material, respectively. For this plot, the first step is to define the project's parameters. The magnetic permeability of the air is a constant given by $4\pi \cdot 10^{-7} H \cdot m^{-1}$. The current was considered to be $10 A$, more detailed calculations on the drivers are addressed in subsection 4.1.5. Finally, the mechanism's design provides values for the solenoid's length, $2.5 cm$, and its diameter of $1.2 cm$. Also, the steel magnetic permeability is considered to be $500 \mu_0$. This formulation is the theoretical base of all three actuator designs.

The units were hidden and replaced by per unit (p.u.), hence all values are normalized. The real physical values differ from this ideal formulation and depends on deeper and more careful simulations. They are addressed in subsection 5.3. Figure 4.5 was plotted to provide a visual interpretation of the force and magnetic flux dynamics on each point of the distance between the ferromagnetic material and the solenoid is variable. The physical elements are based on figure 3.10. Note that the force provided is positive for half of the movement range of the solenoid's length. When the ferromagnetic piston reaches the middle point, the solenoid begins to push it back. The actuator is expected to operate in the dashed line.

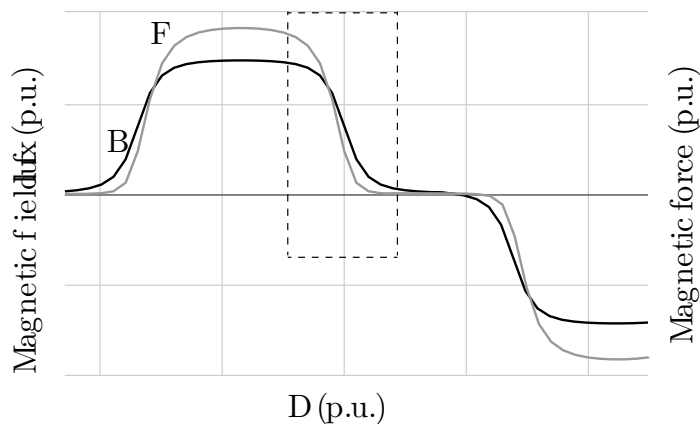


Figure 4.5: Generic ideal magnetic field flux density and force versus distance.

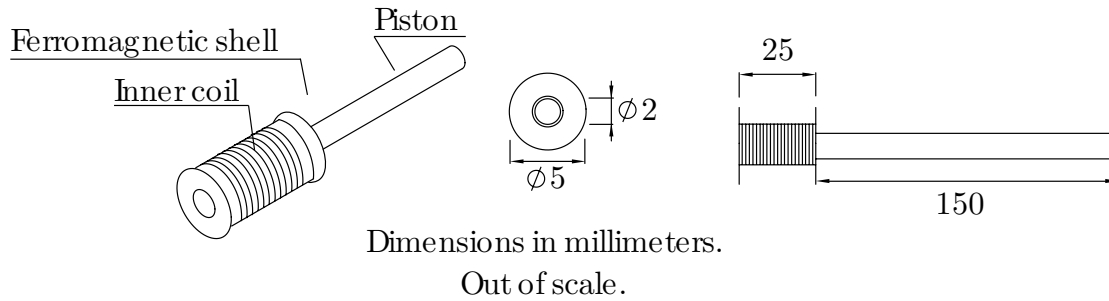


Figure 4.6: Considered design for the magnetic simulations.

Mechanical validation

Further validations of the system’s behavior do not come from algebraic calculations. As said, the dynamics present on this system are highly variable for each analyzed point, and the formulas disregard many physical elements. Such neglects overshadows the mildest behaviors, which impair the results. So, this work used an open-source finite elements software, FEMM, aiming to extract better results. This software provides a magnetic field flux visualization and the force calculations using the weighted stress tensor algorithm. Simulations were run for each millimeter of the movement. Figure 4.6 shows the design used for this simulation, while figure 4.7 demonstrates the winding and mechanism’s assembly.

From that on, Newton’s laws of movement were used to obtain, as precisely as possible, the movement’s performance. For this analysis, it is important to remember that the mechanism will be pointing upwards, so gravity will pull the parts downwards constantly, then the solenoid must provide enough force so that the RC part reaches the top of the FPC. Further discussions and FEMM results are addressed by section 5.3.

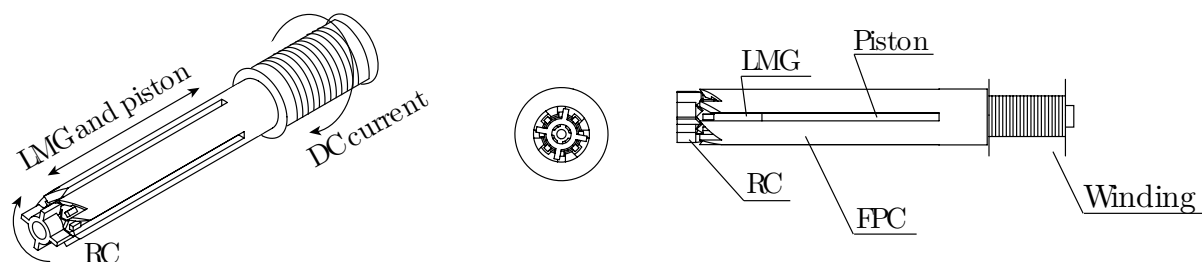


Figure 4.7: All the drivers of mechanical movement assembled.

4.1.2 Solenoid piston

Similarly to the previously mentioned linear-to-rotational mechanism, the solenoid piston option is based on electromagnetic force principles. It consists of a winding that induces a force in a ferromagnetic piece that moves linearly inside of it. The mechanical design is identical to the winding step of the mechanism spoken above. The peculiarity is that it does not intrinsically provide mechanic locking. Instead, a locking design placed at the end of the piston's path is essential for its operation.

This part's physical behavior validation is the same as the mechanism's since it has the same driving winding system. Otherwise, considering a constant gravitational opposite force in the secondary solenoid, as illustrated in the previous subsection, this structure choice demands the addition of a part that provides this function. The obvious choice is a spring, so the opposing force is, in this case, variable according to Hooke's Law: $F = -k \cdot \Delta x$. Therefore, the linear movement behavior depends on the chosen spring's specific elastic constant, k .

Figure 4.8 illustrates the concept of this design's choice. Note that a spring is added to the secondary winding which is placed on the piston path's end as a locking tool. This secondary solenoid feature can draw more implementation challenges to the project since it would add more dynamic elements to the system.

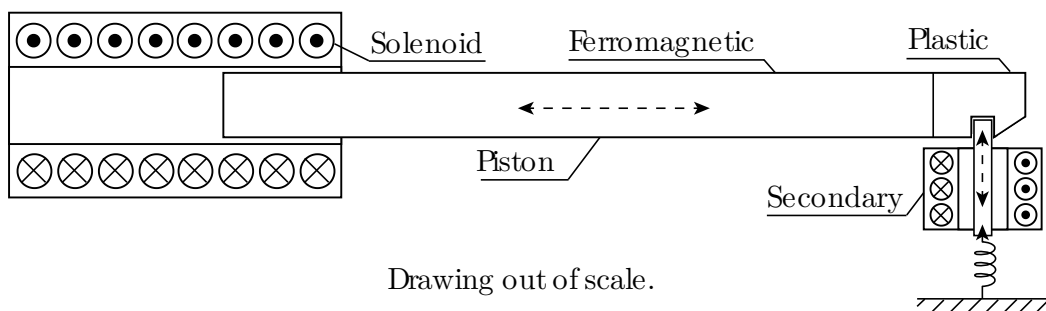


Figure 4.8: Solenoid piston simplified mechanical structure.

4.1.3 Electromagnetic locker

Following the same train of thought, the electromagnet physical characteristics of attraction and repulsion can also be a tool for this work's development. This option, (a) in figure 4.9, consists of the substitution of the mechanical locking, addressed in the solenoid piston subsection (4.1.2), for an electromagnet driven by a winding placed in series with the primary winding. Hence, the system will count with an adding extra force vector provided by the electromagnet's magnetic field, both in attraction and repulsion situations.

This option provides, at first, better safety properties once it has two magnetic fields continually attracting the locker. On the other hand, it demands constant current flow in both windings to counterbalance the spring's opposite force and ensure proper locking, which increase considerably the electric current demand.

An alternative to that would be (b), in figure 4.9, using the electromagnetic force to both push and pull the ferromagnetic piston towards the electromagnet. This option requires careful testing and it also has very high power consuming characteristics. Also, for both (a) and (b) options, the amount of ferromagnetic material needed to implement it pushes the expenses up.. Therefore, considering budget and expected implementation difficulties, the most viable solution would probably be the next topic, the stepper motor.

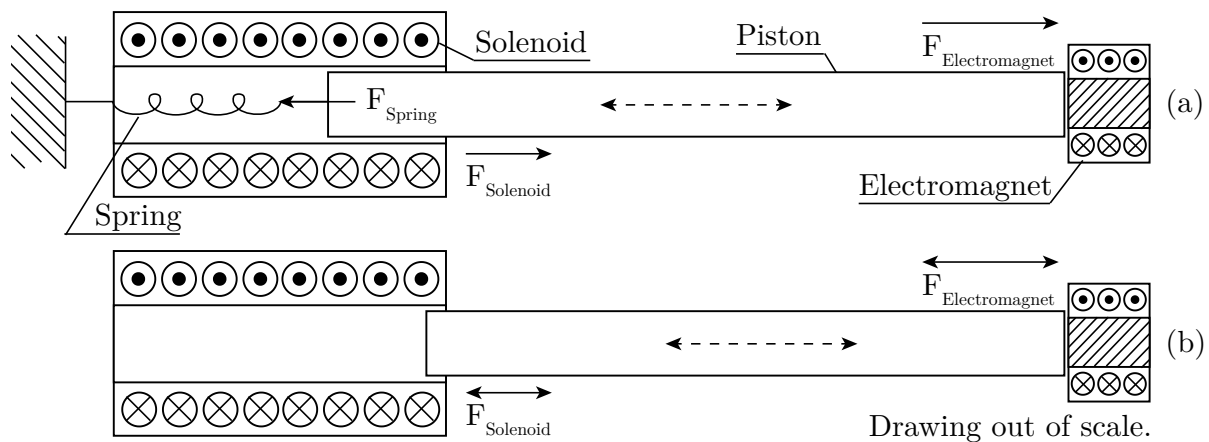


Figure 4.9: Electromagnet locking simplified mechanical structures.

4.1.4 Stepper motor

Lastly, an option is the placing of a stepper motor in each parking spot. The motor option is the safest and most easily predictable locking system, once its behavior depends only on the electronic driver's command. The stepper motor is a widespread technology because of its controllability, easiness of installation, and low-cost features relative to other electric motors categories. As this project considers negligible opposing force for the piston movement, possibly any motor can be implemented. Because of its price, dimensions, and available documentation, this work considers the NEMA 17 model for the mechanical designs. It demands 2.5 A of current and has a step angle of 1.8° , providing 4800 g.cm of torque [57].

However, the mentioned positive features of this choice come at a high price. The stepper motor is still notably expensive in a more comprehensive analysis. For the automation of a few bicycle spots, it is a great possibility. Nevertheless, considering scalability, it is a costly option since it is considerably more expensive than the solenoid options previously presented. Figure 4.10 illustrates the stepper motor.

This work will not consider this subsection in the subsequent chapter 5, Results, since the stepper motor employment is not the focus of this work and it is widespread through many electronic applications, which can be found in many projects' documentation. Still, the stepper motor is cited at the conclusions chapter as a viable and valid option for the deployment of smart parking systems.

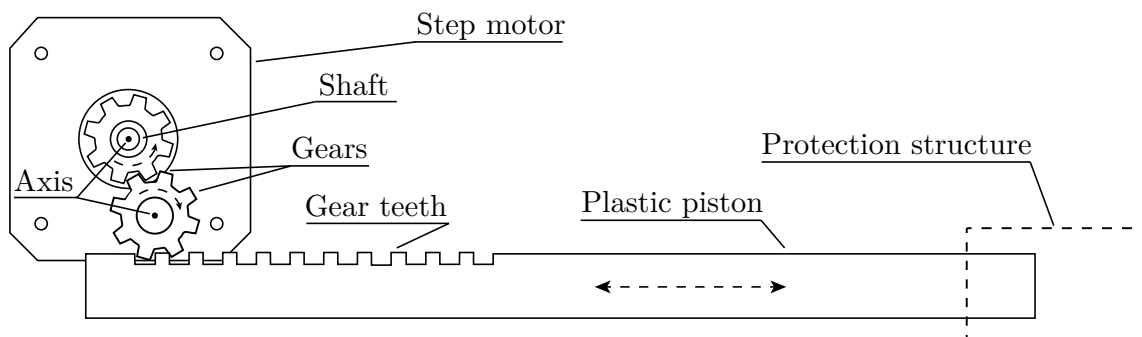


Figure 4.10: Simplified scheme of a stepper motor in a motorized smart parking spot.

4.1.5 Electronic drivers

The electronic drivers are a vital component of actuators. Hence, its proper implementation is crucial for the deployment of CPS such as this project. For the choice of an electronic topology, the outline should take some factors into account.

The project needs to fulfill parameters such as the required current boost from the MCU to the actuator, maintain the correct logic level for enough time, and offer befitting operation ranges to the solenoid. This work comprehends a set of different options for the actuators. Therefore, it extends a few different options for electronic drivers as well, illustrated by figure 4.11.

Darlington transistor

This electronic structure consists of two cascading transistors. A low-boost receives the MCU's PWM signal in its base. Next, the low and high-boost transistors collector pins are connected. In its turn, the emitter of the first connects to the base of the high-boost transistor. This construction characteristics confer the Darlington topology exceptional values of current boost (β). It is relatively more expensive than a conventional transistor but still viable for scalability. Its inherent operation characteristics determine that it works only with one current direction if fed by a single voltage source, which is the case.

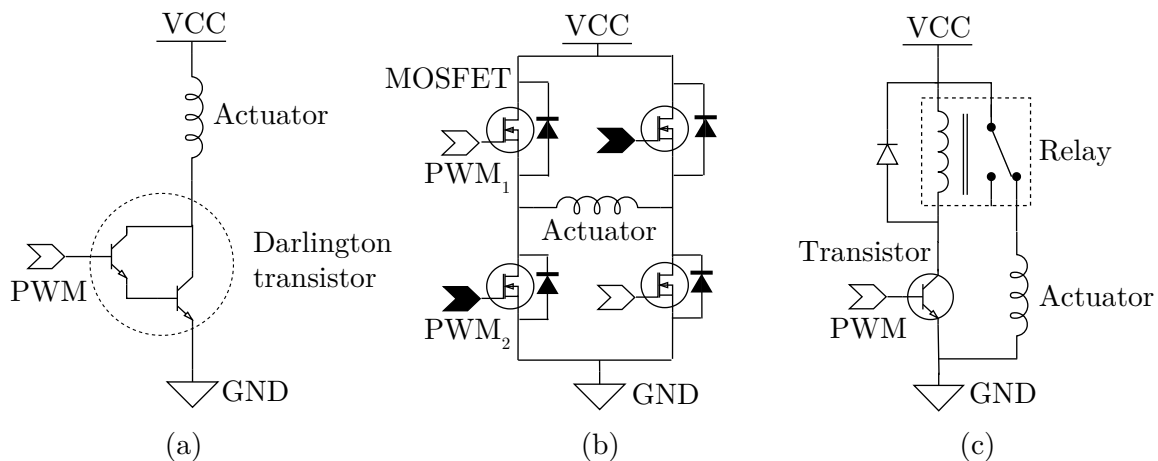


Figure 4.11: Three possible topologies for the implementation of the electronic drives. (a) Darlington transistor, (b) H bridge, (c) Relay.

The Darlington transistor is recommended for the following actuator options: Linear-to-rotational mechanism; double solenoid; and, the electromagnetic locker. The stepper motor requires an operation range this driver does not meet. This work considers 2N6052 model, which operates with 12 A, 100 V and is a PNP, it triggers at 2.8 V of base-emitter voltage [58]. Any Darlington transistor that meet the MCU and current requirements could be implemented here.

H bridge

The H bridge is a classic placement of four transistors in a way that two independent PWMs control the signal transferred to a load, the actuator in this case. It can operate by applying both positive and negative current flux in the actuator, which is an advantage. On the other hand, the H bridge needs more electronic components, which increases the overall price of the system. Summarizing, this option's operation range is great, but it requires more electronic parts and GPIOs from the MCU. These properties make this option feasible in some applications where its advantages outweigh the disadvantages.

The H bridge could be employed in any model to commute the winding's current when the piston reaches its middle point, so the solenoid apply force continuously to it. It is an option to boost the electromagnetic force. The chosen component to build the H bridge is FCB125N65S3, a power MOSFET, N-channel, which runs with 650 V and 24 A [59].

Relay

The relay works, different from transistors, switching pins according to a signal they receive from the MCU. Instead of a PWM, the microcontroller sends a control order to commute the switch when required. The relay would be a viable option in the electromagnet scenario, where the MCU would send a commute signal so that the locker secure the bicycles' wheels and another to release it. In the meantime, while the electromagnet does not receive any command, it would remain to retain the piston using its electromagnetic force fed by the voltage source. A generic relay was considered for the simulations.

4.2 Sensor operation

The system's sensor is a somewhat simple solution. There is a need for information on whether a bicycle is occupying a specific spot or not. This report serves as a flag for the system's routine, such as turning on, restart after fault, proper spot management, and confirmation of vacancy for the smart parking's virtual part. Its function is solely to alert the physical asset controller (PAC) when a bicycle has been placed or left its spot.

The found solution is a push-button connected to the electronic driver's PCB through wires. It is connected permanently to the ground and VCC with a pull-up resistor structure, as seen in figure 4.12. The exit of this system connects to a GPIO of the microcontroller coded as an input for an interruption. When the button happens to be pressed by the bicycle's wheel, the signal goes from VCC to ground. The controller detects this and executes the coded routine to save the information, alert the cloud computing device, and activate the actuator.

The pull-up arrangement is required for guaranteeing the high or low-voltage level signal entering the controller's input. Without it, a residual voltage may inadequately trigger the interruption cycle. Also, the PAC is programmed with a debounce code to filter the possible environmental natural noise and function properly, making sure it will not trigger actions before it should.

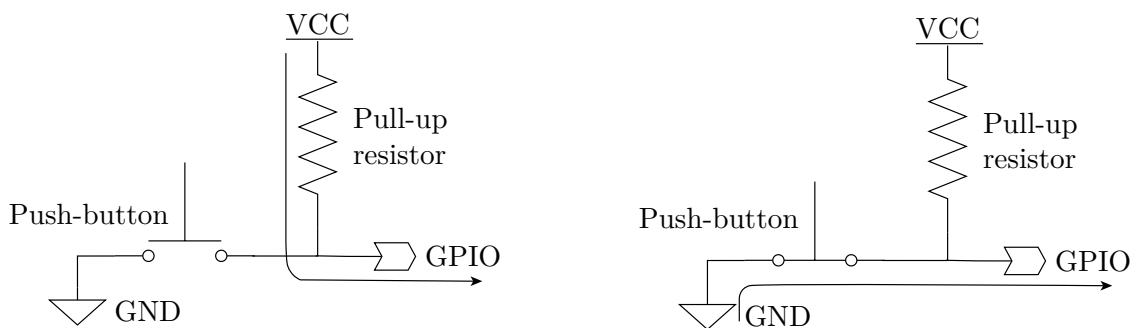


Figure 4.12: Pull-up resistor structure for the sensor.

4.3 Embedded power supply

Naturally, a bicycle parking spot's project condition is autonomy. It often stands far from any electricity supply spot and plans to automate it must consider independent energy supply. Hence, this project recommends the use of a small photovoltaic (PV) panel. So, the power supply system consists of a PV panel, an electronic driver for the charge and discharge management, and a battery to hold the charge in periods without sunlight.

4.3.1 Photovoltaic generation

With time, the photovoltaic generation transformed from an academic topic to a full commercial-ready reality. Now it is used from big solar energy farms to day-to-day devices. In this context, this work proposes the use of a small panel powering each bicycle's parking spot cluster.

The number of spots on each cluster varies, hence the need for electric power, ultimately altering the sizing of each panel. This document considers a six parking spots cluster, the most common in the IPB campus. In the scalable scenario, the author recommends a study on individual cases for the determination of each electric power demand intending to avoid oversizing.

Figure 4.13 shows the proposed schematic. As said, a PV panel uses the sunlight energy to provide electricity to the electronic battery management system (BMS). In its turn, the BMS manages the power flow between the PV panel and the charging battery, or the discharging battery and the physical elements of the smart parking system.

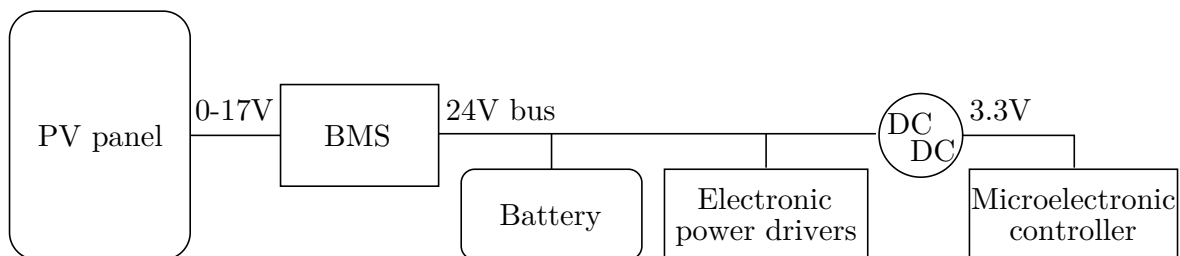


Figure 4.13: Proposed power flow.

4.3.2 Batteries and BMS

While the sizing of the PV panels regards the whole system’s energy demand, the batteries dimension depends on how long the parking spots will remain powered without sunlight. These calculations follow the PV panel sizing, as it relies on these results.

The batteries should maintain the system’s power supply during the night and possible weather events that undermine solar generation. Also, the microcontroller’s software has a security feature that monitors the battery’s output voltage so that, in the event of low charge, the MCUs enter a routine of battery saving by not allowing new users into the spot (the actuators of vacant spots are opened for use by physical lockers), it also sends a message to the broker so the user is notified. Consecutively, the BMS system choice is based on both the PV panels and the batteries previously selected. Its input must support the panel’s output voltage, and its output must be adjustable to the batteries’ and the system’s voltage level.

Figure 4.14 illustrates the current flow in the two likely situations: (a) PV panel and (b) battery feeding the system. In the first case, the current flows from the panel to the battery and system. Otherwise, when the battery takes over, the current flows from it to supply the system. The project places diodes between the battery’s nodes to avoid backward currents that can damage components such as the battery, BMS, and PV panel.

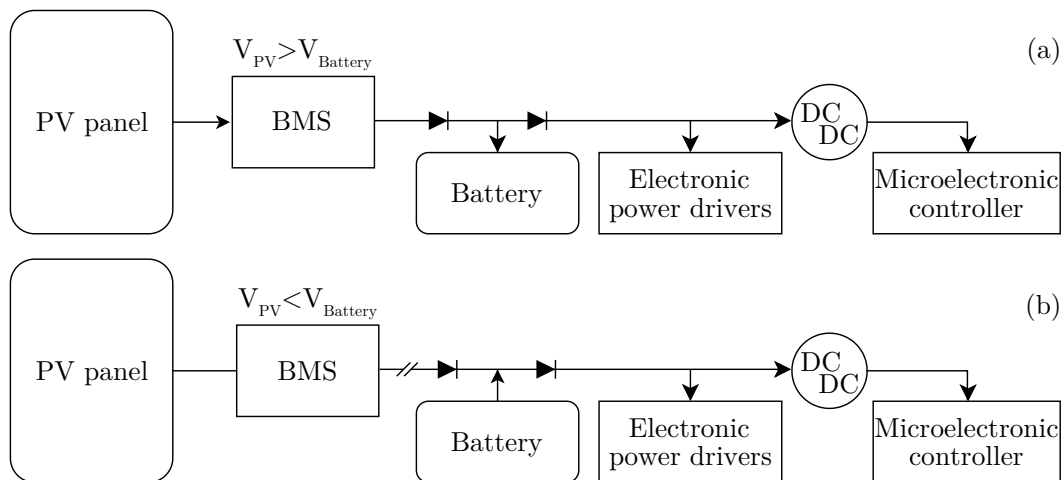


Figure 4.14: Current flow during expected two operation states.

4.4 Microcontroller

The microcontroller (MCU) is responsible for the control of all the physical features of the system. For this work, each MCU controls three parking spots at a time. That is because all clusters at IPB are a set of three parking spots, hence each of these clusters would receive a single microcontroller unit. In a scalable situation, each set of three parking spots receives a microcontroller accordingly. Figure 4.16 demonstrates this configuration.

The microcontroller communicates through an MQTT server running in the cloud, receives three interruption signals independently, and controls three actuators to lock the user's bicycles. The microcontroller ESP8266 NodeMCU Lua Wi-Fi meets all presented specifications and is reasonably simple to code. Therefore, it was chosen for this project. Figure 4.15 demonstrates the referred MCU's pinout schematic. It consists of 30 pins, in which there are sixteen GPIOs, 4 GNDs, three 3.3V port, and support for a variety of communication protocols, for example, I²C. This work uses the pins highlighted in bold. Notably, the pinout is underutilized, but the code will run MQTT communication through Wi-Fi, which loads the MCU's processing operation.

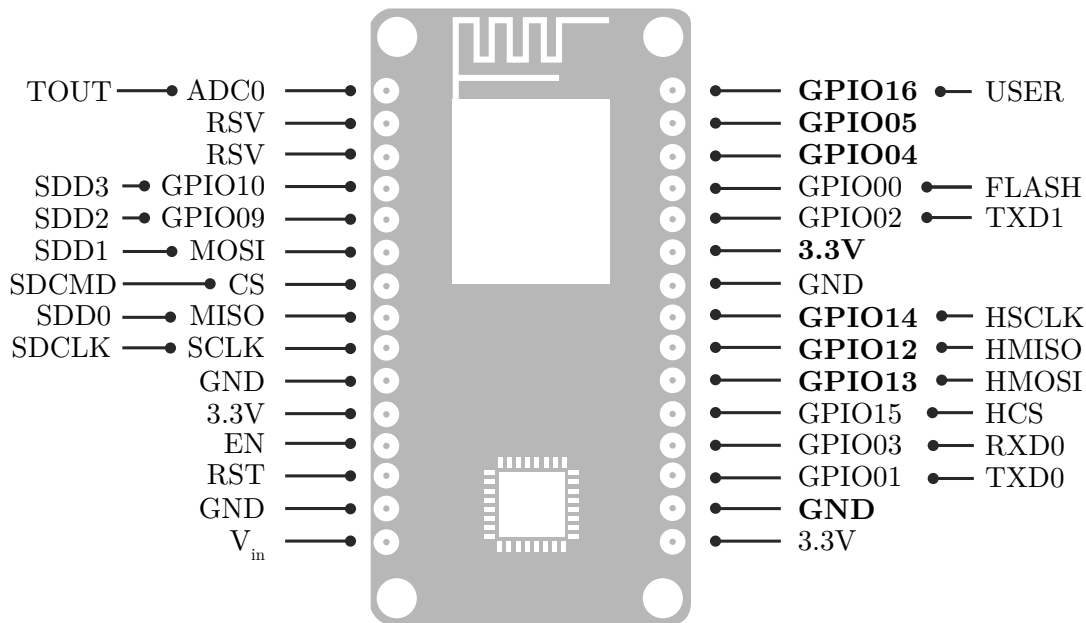


Figure 4.15: ESP8266 NodeMCU Lua Wi-Fi pinout layout.

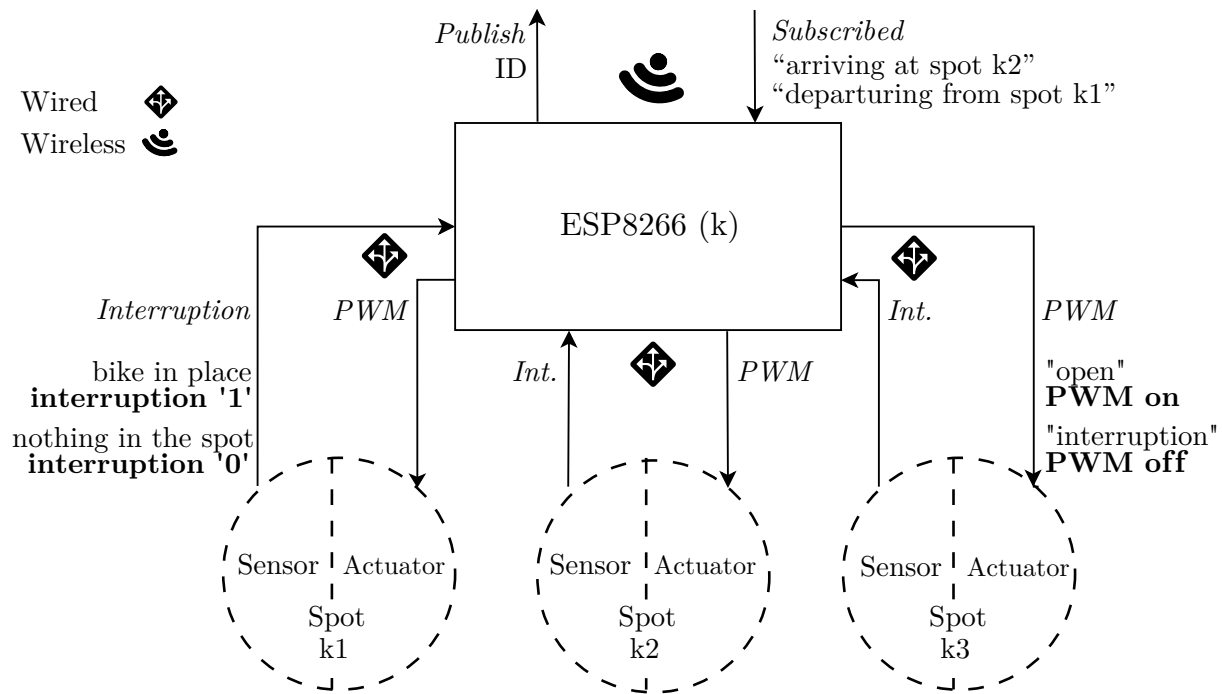


Figure 4.16: ESP8266 MCU proposed schematic.

Another possibility, not implemented here, is the use of the ESP8266-01 series module, which has fewer pins but, if implemented, would bypass the underutilization addressed before. Figure 4.17 illustrates this option’s pinout schematic. Notably, fewer pins allow the MCU to control fewer spots at once. Hence, the implementation architecture must comprise the single-spot control for each ESP, in this case.

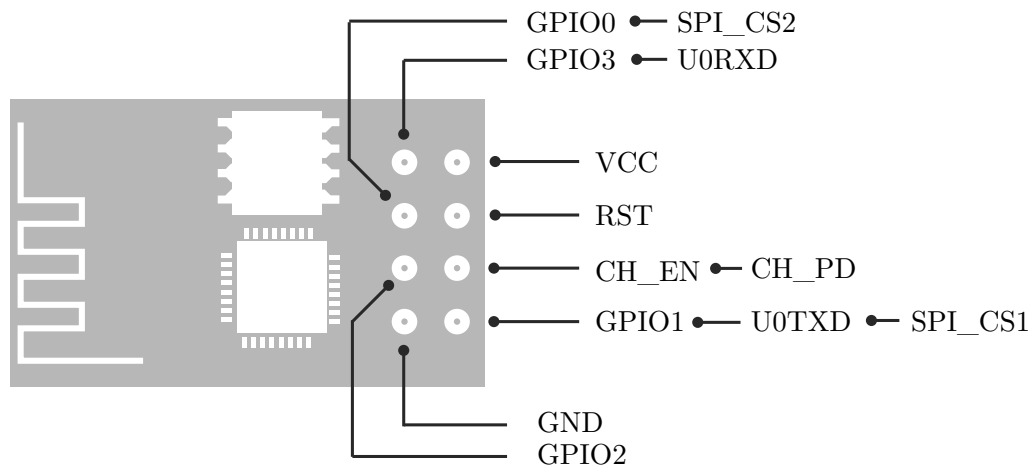


Figure 4.17: ESP8266-01 series module Wi-Fi pinout layout.

Chapter 5

Results

This section of the document explores results achieved during the work. It begins exploring the resources used through its development, moves to the mechanical designs and validations, finalizing with the magnetic distributions and electronic simulations. Each subsection describes the project's parameters taken for its development, followed by an explanation of the produced calculations, and concluding with graphical demonstrations.

5.1 Software resources

As a conceptualization project, this work is naturally dependant on a set of multidisciplinary software to validate actuator concepts, microcontroller routines, and communication parameters. SolidWorks 2019 and 2020, installed with the IPB's students license, were used to design all mechanical parts and simulate its movement. Next, the open-source FEMM software was employed in this work to simulate and analyze the existing magnetostatic behaviors and the relation between movement and magnetic force in the designed solenoid. The data processing of FEMM's results was developed in the free online software Google Sheets. Finally, the open-source LT Spice electronic circuits simulator was used to calculate the behavior of the electronic drive, while Autodesk Eagle student's version was used for the PCB's design.

5.2 Mechanical design

The design of the mechanical parts is crucial for the proper deployment of this project. It defines many variables for the following analysis and serves as the base to build the physical implementation of a smart parking system. This stage of the project runs alongside the magnetic design following in subsection 5.3 once any alteration on the mechanical design directly impacts the magnetodynamics simulated and treated data.

The first device consists of the implementation of the locker based on the linear-to-rotational mechanism. As seen in figure 5.1, it consists of a solenoid block that acts as the movement propellant pushing the mechanical parts. Next to that block, the mechanism is fixed on the same axis. The moving parts of the mechanism, in its turn, drive a second piston that transforms the linear upper movement to a side movement, locking the bicycle through a set of two identical gears arranged in a way they rotate together.

Finally, this device is shifted 45 degrees to match the IPB's spot design, once it does not fit horizontally to the available space. Which is the same reason why the initial idea of using the mechanism directly pointing sideways did not work. Due to the reduced available space, the set of gears and spare pistons were added to adapt this design. This design would fit better in a situation presenting more spaced parking spots, disposing the gears and added cylinder parts, hence considerably decreasing the mechanical complexity.

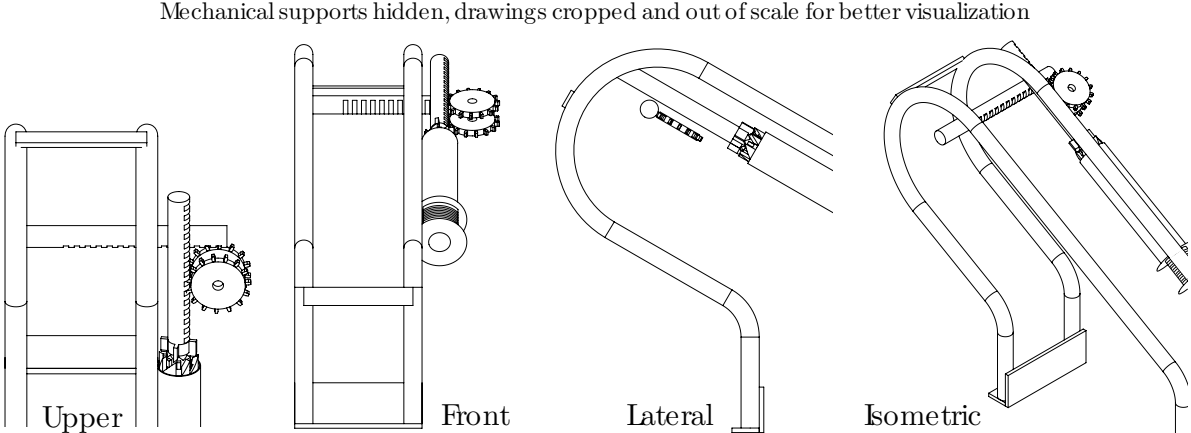


Figure 5.1: Mechanical project for the mechanism proposal.

The remarkable feature of this plan is the natural locking without the need for a device at the other end of the piston's path. Once the mechanism's RC reaches the top of the LMG, it locks. When the solenoid is triggered once more, in the same direction, the RC rotates and unlocks the backwards movement, pulling the locker with its components weight. Therefore eliminating the need for secondary devices or negative electric current.

In sequence, the second concept handled on the mechanical simulations software was the double solenoid proposal, shown in figure 5.2. It incorporates two solenoids, a spring, and a designed part to lock the primary piston's linear movement with the secondary. The secondary winding is triggered to unlock the system pulling its piston downwards.

In this idea, the electronic driver has two distinct operation cycles for locking and unlocking. In the first routine, it only supplies current to the primary winding to move the piston that locks the wheel, which is secured by the secondary piston, being pushed upwards by the spring, at the end of its path. Otherwise, to unlock, the driver triggers both windings, so the secondary releases the primary piston, which is being pulled back. These characteristics add complexity to the double solenoid, but it is mechanically more simple due to the fewer moving parts it requires in the implementation. Also, the electronic driver can push and pull it with only one direction of electric current, which simplifies the electronic circuit project as well.

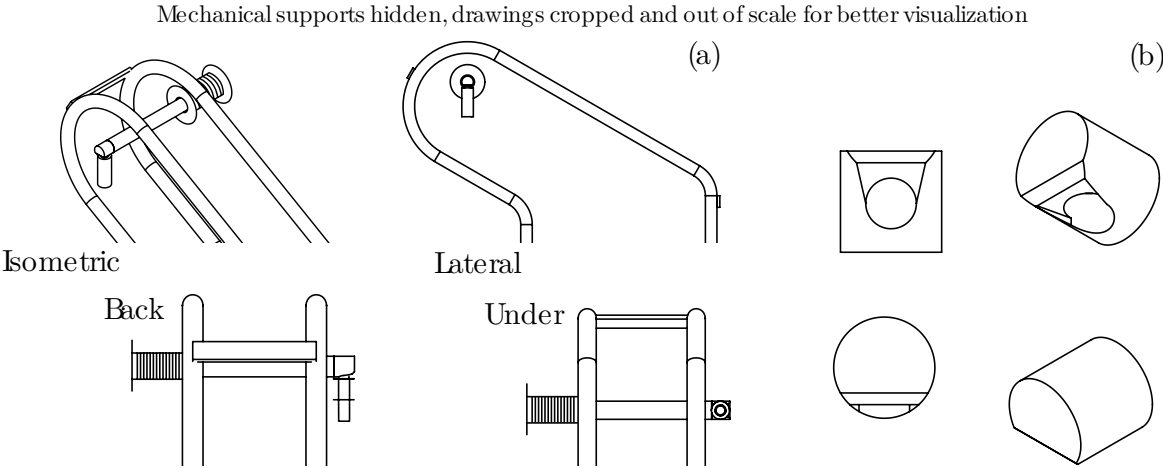


Figure 5.2: Mechanical project for the double solenoid proposal.

Lastly, figure 5.3 illustrates the (a) stepper motor and (b) electromagnet options. These two were not simulated, as justified in the previous chapter, but are also options for the actuator implementation. The stepper motor is an expensive choice, but it is expected to present fewer implementation difficulties since its deployment is widely documented. Also, it is the most simple mechanical project, which is convenient for installation and maintenance.

In its turn, the electromagnet is very similar to the double solenoid project. The difference consists of exchanging the perpendicular secondary system for an axis-aligned electromagnet that will be connected to the electronic driver aiming to add an extra force pushing and pulling the piston. As previously said, this option requires constant electric current flow, therefore it is expected to considerably increase the power demand.

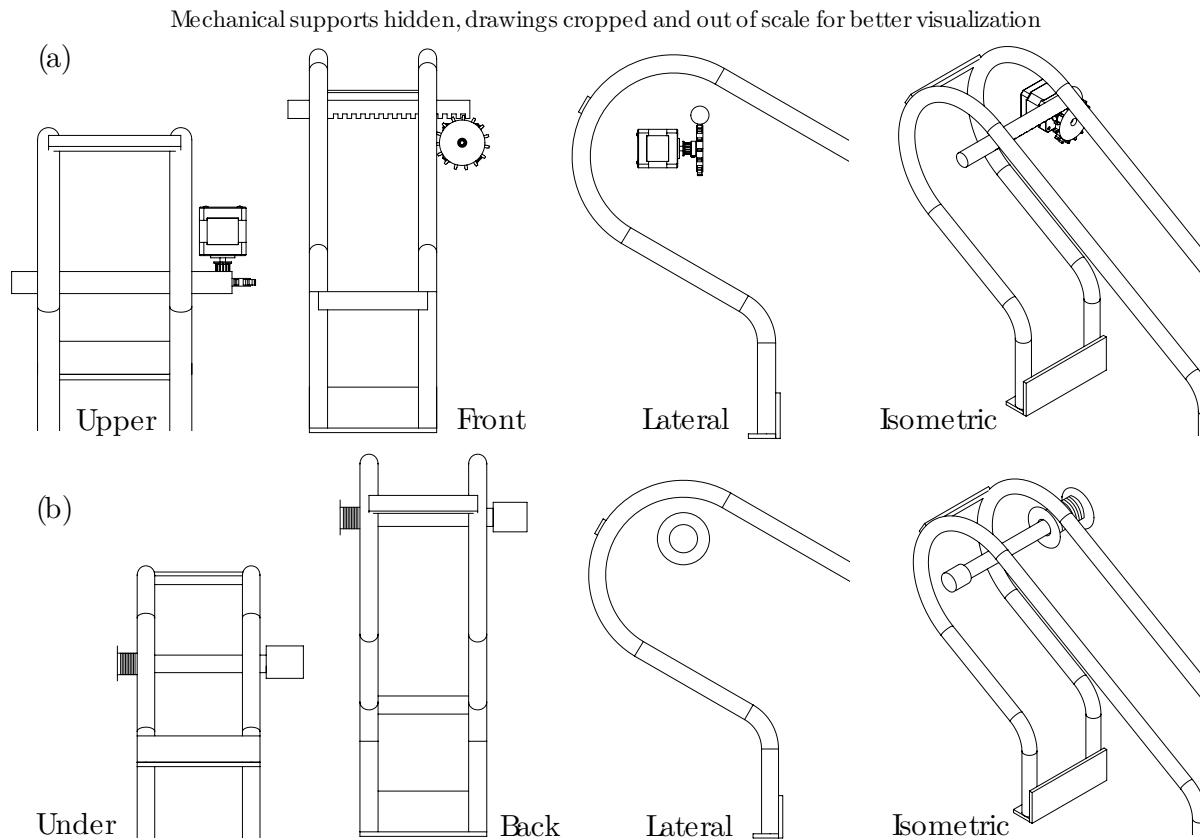


Figure 5.3: Mechanical projects for the electromagnet and stepper motor proposals.

Figure 5.4 better illustrates the double solenoid and linear-to-rotational mechanism proposed designs and intended operation. The left and right images show lateral views of each design path's starting and ending point, respectively.

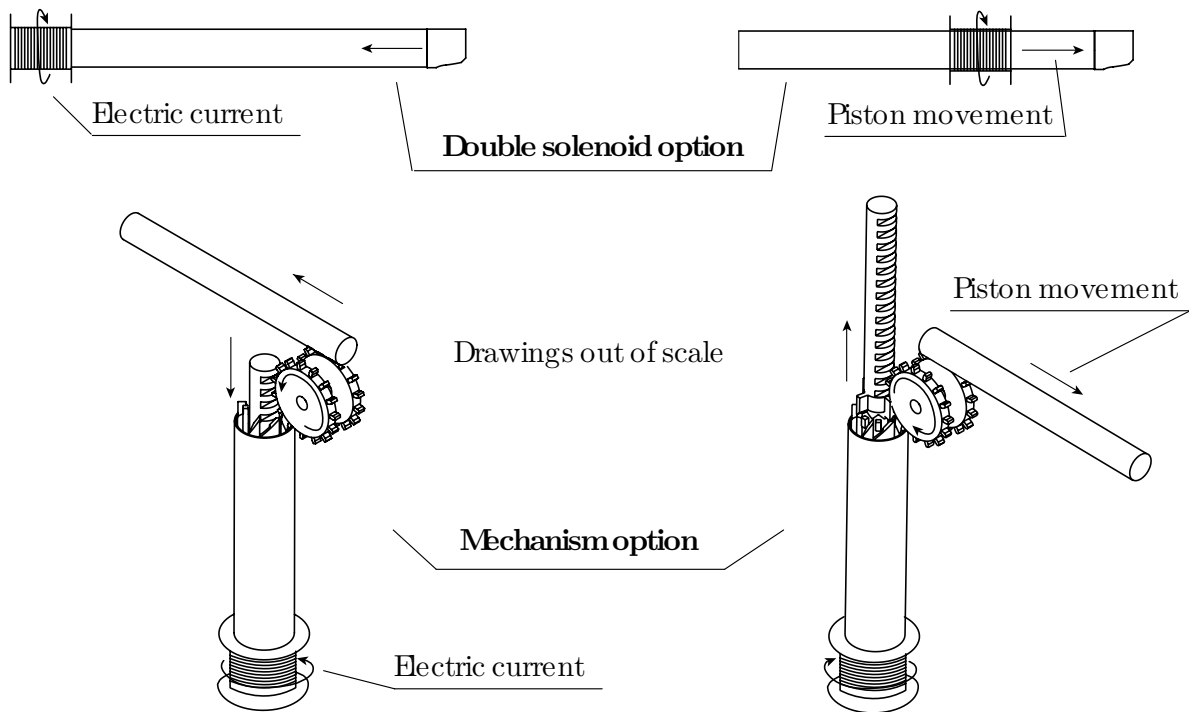


Figure 5.4: Mechanical designs scheme and proposed operation.

5.3 Magnetic distributions

Starting from the early design sketches, this step of the project was developed alongside the mechanical design process. The initial idea was to merge both winding and linear-to-rotational mechanism in a way that this scheme choice would save space in the parking spot. The magnetic simulations revealed this proposal did not provide enough force so that the device would correctly operate. A careful analysis clarified that these results were obtained because of the long gap between winding and piston that could not be redesigned maintaining its proposed operation.

Based on section 3.2.3 equations and expected mechanical behavior, a new action plan was drafted. It consisted of separating winding and mechanism in a way the magnetic field has more accurate low-reluctance paths and a very reduced gap. The following scheme had considerably better response in software simulations, confirming the theoretical expectations. Figure 5.5 illustrates the simulated cross-section. In this schematic all the non-ferromagnetic extra parts are hidden once it does not interfere in the simulations.

The selected ferromagnetic material for the simulation was ferromagnetic steel with a magnetic permeability of $\mu_r = 460 \cdot \mu_0$, the winding supports 10 amps and has a thousand turns of copper wire. These values are important because the electronic drivers addressed in subsection 5.4 have to supply the electric current the circuit demands for its proper operation.

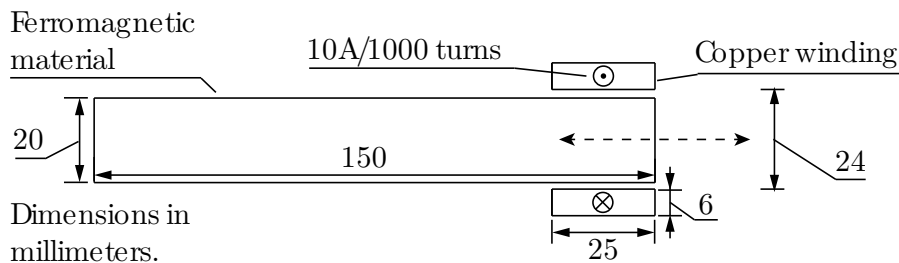


Figure 5.5: Drawing of the simulated solenoid.

This design was transferred to the finite elements magnetostatic analysis software. The software returned the results for the density plot and integral force calculations via weighted stress tensor. FEMM software provides the density plot in every desired points of the operation through simulation, along with the integral weighted stress tensor force results. The simulation was iteratively repeated to each point of interest defined. There is a point for each half-millimeter for the first thirteen millimeters of the piston's path. From the thirteenth to the 58th, there is a simulation point for each millimeter. And from then to the 158th, a simulation for each 10 millimeters. This resulted in 72 hand-drawn simulation points and, therefore, software outputs.

Table A.1, in Appendix 5, reports the information obtained from FEMM, both through simulated and treated data. The first column consists of the distance between the beginning of the solenoid to the piston's frontal face, and it was the software's input. The second column, F (N), is the electromagnetic force the solenoid applies to the piston, i.e., the simulation's output. Next, the accelerations were calculated from the given force and the part's weight for each case. For this, there is a need for an analysis of the force diagram for each actuator option, shown in figure 5.6.

The calculations used the materials parameters and FEMM's output data to determine, using linear accelerated movement equations, the expected physical behavior of each option. Below, the step-by-step process is more carefully addressed.

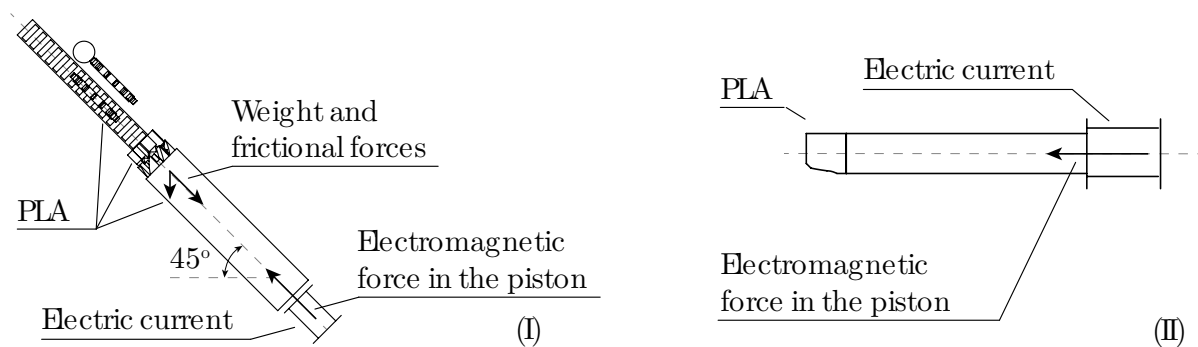


Figure 5.6: Forces diagram analysis to treat the data from FEMM.

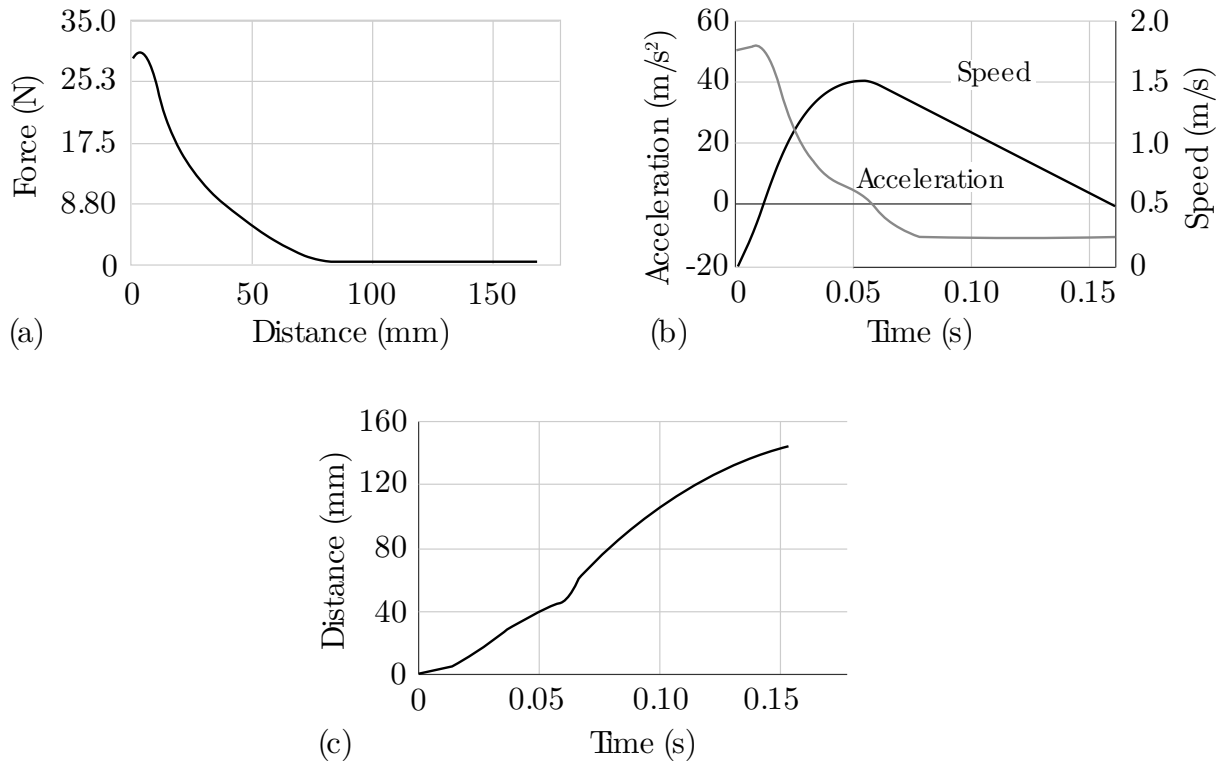


Figure 5.7: Physical behavior of the components from treated FEMM simulation data.
 (a) Force on the piston (FEMM) vs. distance between piston and solenoid;
 (b) Acceleration and speed of the piston (UARM eq. predicted) vs. time;
 (c) Traveled distance (UARM eq. predicted) vs. time.

Starting with (I) the mechanism option, the parts densities are 7.85 for steel and 1.24 g/cm³ for the 3D printed PLA. According to the volume of the mechanical parts, available from the SolidWorks design, the total mass the solenoid would push is approximately 0.5 kg. Gravity's constant opposing acceleration is considered 9.81 m/s². First, acceleration is obtained from $F = m \cdot a$. Next, the speed is determined based on the UARM equation for the speed: $s^2 = s_0^2 + 2 \cdot a \cdot \Delta x$. In sequence, the time for each fraction of acceleration comes from the equation $t = \Delta s/a$. Figure 5.7 shows the graphical plot of this data.

Note, in (a), the force has a behavior very close to the expected, as shown in figure 4.5, but with the solenoid turning halfway through the linear movement. The solenoid is turned off at the point where the electromagnetic force reaches zero, so it does not pull the components back. (b) plots the resultant acceleration and speed. (c) illustrates the movement of the piston, note that the curve tends to reach its peak at the very end of the

chart at the same time the speed curve is about to reach zero. If the speed turns negative in (b) and the curve (c) shows an inflection point before the 160 millimeters, the force is undersized and the project would need adaptations.

Here, we see this solenoid sizing is working on the limit. This work considers the worst-case-scenario, but if in the implementation more forces are discovered to play a role in the system, adaptations can be made on the solenoid winding characteristics, adding more electric current or winding turns to drive more force into the piston. It is noteworthy not to oversize it, since higher forces are provided at the expense of higher values of impedance. This changes would pull up the speed behavior seen in the very end of (b) as well as the overall behavior of (c) curve.

5.4 Electronic simulations

For this stage of the project, the drivers topologies, addressed in subsection 4.1.5, were transferred to the software LT Spice. The development of the simulations stands on the expected current and voltage levels, which are based on the previous magnetic simulations and the characteristics of typical commercially available batteries and PV panels.

The DC voltage supply is assumed to range from 0 to 17 V as a standard low-power PV panel. The power bus of the system, in its turn, was defined 24 V because of the conventional battery voltage rates. Also, the higher the available voltage, the less current is needed to drive strength from the solenoid's magnetic field. As figures 4.13 and 4.14 shows, the BMS handles this voltage difference internally, so the system is protected from short circuits due to the different voltage levels.

Table 5.1 lists the parameters and materials' categories defined for the deployment of this stage of the project. Also, the determined self-inductance, resistance, and minimum current that the driver needs to supply. The solenoid's physical dimensions are the same as the magnetic simulations.

Table 5.1: Solenoid's parameters to define the electronic drivers' specifications

Description	Type	Characteristics
<i>Previously defined physical characteristics</i>		
Wire	AWG 18	20.73 Ω / km
Number of turns	N	1000 esp
Diameter	D	24 mm
Length	l	25 mm
<i>Resultant values to be used in the simulations</i>		
Self-inductance	L	22.74 mH
Resistance	R	1.563 Ω
Minimum current	I_{min}	10 A

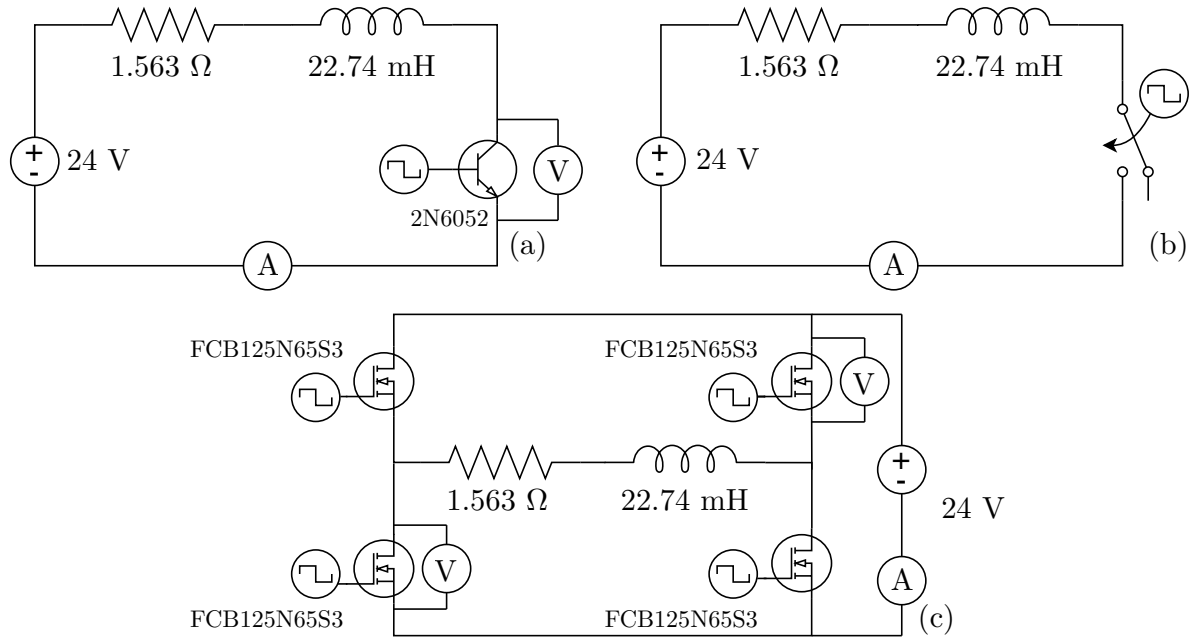


Figure 5.8: Electronic circuits as simulated in LT Spice.

The next step consists of inserting this information in the circuits simulation software, as illustrated by figure 5.8. (a) shows the Darlington transistor topology, which could be controlled with PWM or control command; in its turn, (b) represents the relay choice, controlled by a commute signal; finally, (c) shows the H bridge with internal freewheel diodes, an option when the system needs both positive and negative currents. All figures include the load, composed by the solenoid's resistive and inductive parameters.

The highly inductive load was identified as a challenge for the project's simulation. The physical characteristics that enhance the available electromagnetic force for the piston's driving also pull the inductance value up due to the electromagnetic formulations. Factors such as many turns and reduced length of the winding, the high current demanded, and the diameter of the piston end up making several options unfeasible.

The starting option of working with a PWM signal in the Darlington transistor and H bridge was discarded after the simulation. It pointed out that the electric current would increase at a slow pace due to the cited highly inductive behavior. The implications of the implementation of these options would be the current undersupply, and therefore lower electromagnetic force.

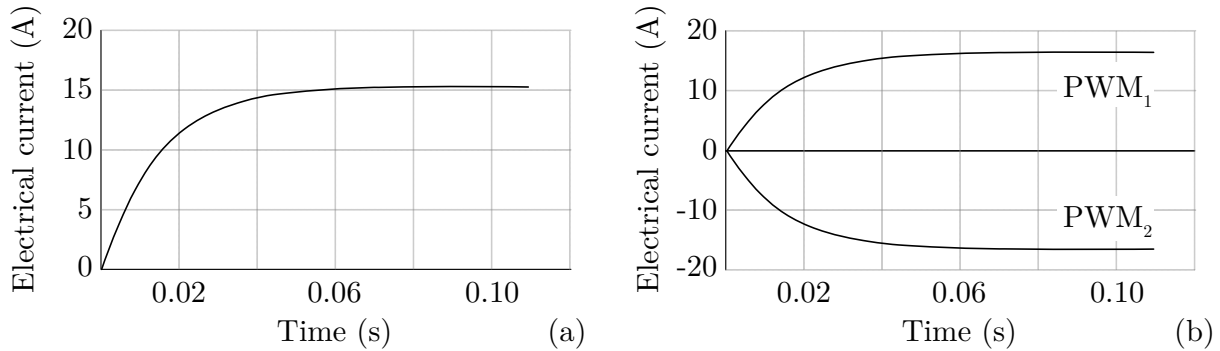


Figure 5.9: Simulated behavior of the (a) Darlington and relay, and (b) H bridge.

The discovered solution is to operate with the transistors in its active region, and not the cutoff, as previously stated. So, it is expected that it presents higher energy losses, which directly affect the PV panels and batteries sizing. That said, the following charts 5.9 exhibit the results for the operation with Darlington transistor, relay, and H bridge.

Notably, as the Darlington transistor is working in the active region, its performance should be almost identical to that observed on the relay operation (a). The choice between these two options occurs due to other factors, such as energy spends and components' price. (b) shows the H bridge, which is also operating in the active region, so its advantage consists of enabling both directions for the current flow, as said.

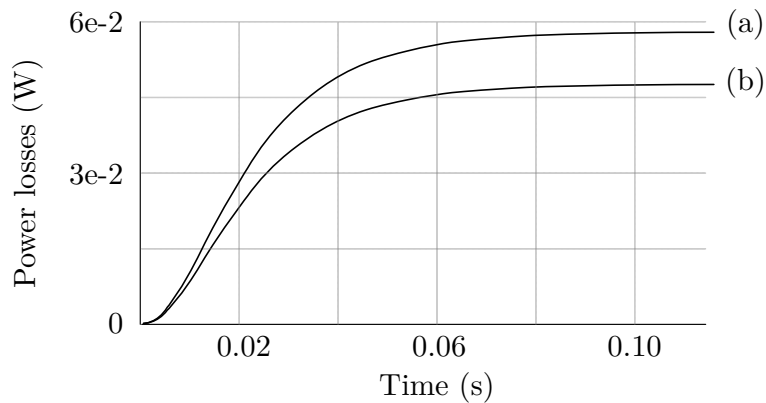


Figure 5.10: Simulated power losses of the (a) Darlington transistor and (b) H bridge.

Finally, figure 5.10 illustrates the simulated power losses for a generic Darlington transistor and two MOSFETs (present in the H bridge). Of course the values will change accordingly to the specifically selected components in the implementation steps, but the curve should remain with the same format and should not present much different values, remaining within the same order of magnitude. For the H bridge, it is estimated by the simulation a maximum power loss of $0.0166e-3$ Wh per drive, while the Darlington transistor value reaches approximately $0.0208e-3$ Wh per drive. The relay option should be proper physically tested in order to obtain an approximate value of its power losses.

5.5 Photovoltaic panels and batteries sizing

The first step towards the sizing of the project's power supply is to survey all the elements that demand electric power in the system. The central electricity need distinctly occurs on the actuator once it requires high currents and has an intrinsic resistance associated with its copper windings. As presented, the resistance is estimated in table 5.1, 1.563 Ω . With the value of the expected flowing current, 10 A, and the time it remains turned on for its operation, 0.06 seconds, it is possible to calculate the energy spending on each use of the actuator.

$$E_{actuator} = \frac{1}{3600} \cdot 1.563 \cdot 10^2 \cdot 0.06 = 2.605 \text{ mWh per use}$$

As discussed in section 5.4, above, the power losses in the electronic switches are approximately 0.0208e-3 Wh for each use. So, the following calculations consider an extreme scenario in which users request an activation per hour for twenty hours every day, aiming to calculate the required power for the electronic power drivers. It also counts a 20% safety factor. Of course, more detailed studies comprising the bicycles' users pattern of behavior would enrich this step of the project, avoiding batteries and PV panels oversizing.

$$E_{power} = 1.2 \cdot 20 \cdot [E_{actuator} + 0.0208 \cdot 10^{-3}]$$

$$E_{power} = 63.02 \text{ mWh per day}$$

According to the chosen MCU's documentation [60], the power consumption during typical operation ranges between 70 and 100 mA at 3.3 V. Therefore, to grant a 24h operation with a 20% of oversizing for security reasons:

$$E_{MCU} = 3.3 \cdot 0.1 \cdot 24 \cdot 1.2 = 9.500 \text{ Wh per day}$$

As said, each MCU controls a total of three actuators, each of them having its electronic driver. Hence, the estimation of the total power consumption per day is given below:

$$E_{total} = E_{MCU} + 3 \cdot E_{power} = 9.690 \text{ Wh per day}$$

$$E_{total} = 9.690 \text{ Wh per day} = 403.7 \text{ mAh per day}$$

The power consumption stated above is considerably low, and the PV panels' sizing offers values under the minimum commercial available. Therefore, oversizing is expected in this part of the project. Also, there is no need for a highly precise calculation of the PV panels due to the previously mentioned reasons. Hence, a renewable energy software [61] was used as a calculator for the PV panels and battery pack sizing.

The input data for the software were the electricity consumption, assumed 10 Wh per day; the days the system is supposed to run without sunlight, choice of 7 days; the effective capacity of the battery bank due to lower temperatures, choice of -1 °C because of the low temperatures in Bragança's winter; and it considers a 50% discharge to optimize the battery lifespan. For the PV panels, the information on the average of 8 sun-hours per day in Bragança [62] is added.

The software outputs are a 24V battery pack with 10 Ah charge capacity, and a PV panel electricity generation of 1.6 W. As all parking slots in the IPB campus comprises six spots, each three composed by one MCU and three electronic power drivers, one 5 W photovoltaic panel would be enough to feed the three batteries.

5.6 Microcontroller routine

As discussed in subsection 4.4, the project defined in its early stages that each microcontroller unit would act as the physical asset controller (PAC) of three parking spots in clusters called parking slots. The MCU coding consists of the following routines, pinout setup and physical assets' verifications, connection to the WiFi and MQTT server, main loop, and message received from the virtual controller. A general view of the MCU coding routines is illustrated by figure 5.11.

As the code starts running, the microcontroller defines three GPIOs as inputs and three more as outputs. The outputs are used for the control of the actuators and the inputs to enable interruptions from the sensors. Next, a function runs through the sensors' inputs

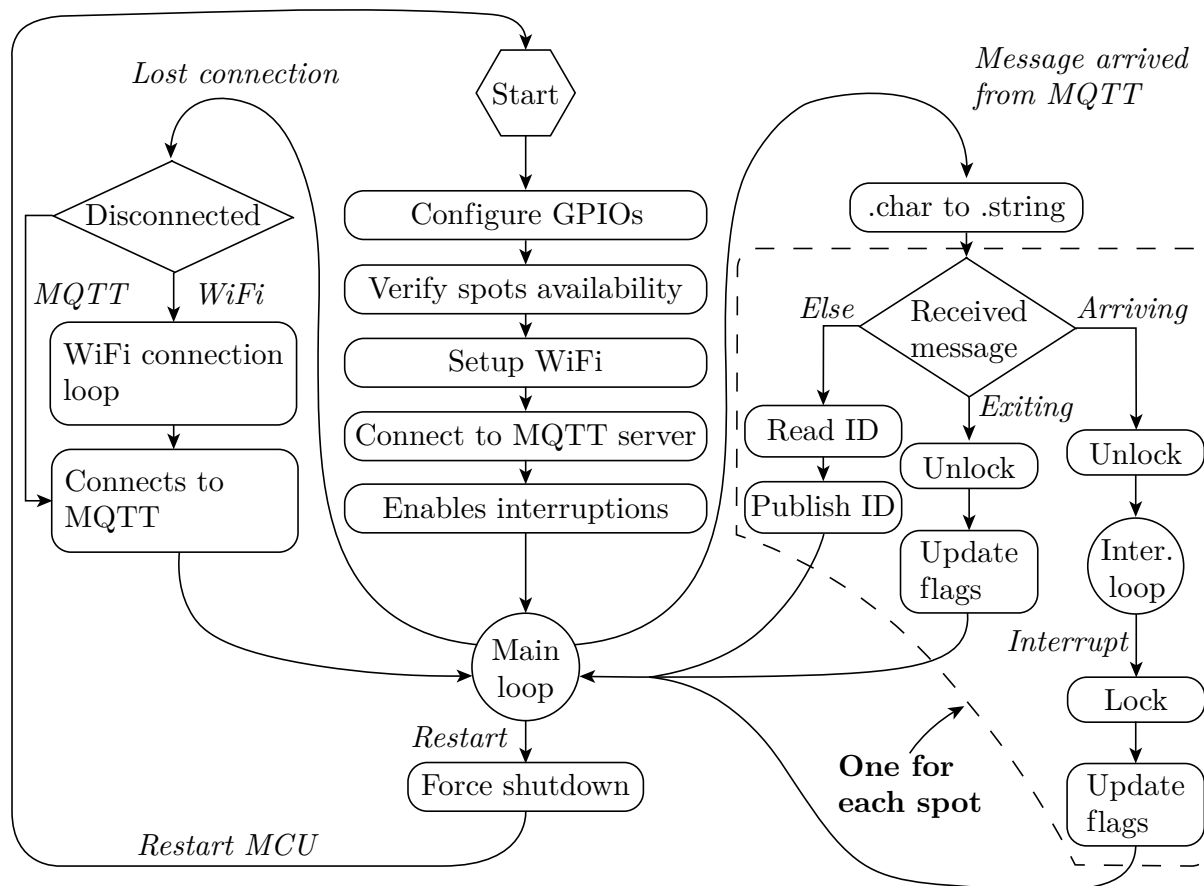


Figure 5.11: Flowchart of the MCU code.

to check its logic level aiming to verify if there is a parked bicycle in any of the spots, so the flags are updated. Right after that, the MCU connects to the local WiFi system and then to the MQTT server deployed in the same network. By the end of the setup stage, the code enables the debounced interruptions.

The WiFi and MQTT libraries already provide looping checks that run alongside the MCU's programs. Hence, the microcontroller can automatically reconnect in a disruption situation. If the MCU disconnects from the WiFi, it enters the reconnection to the internet routine and then the MQTT. Otherwise, if only the MQTT communication falls, it goes right to its connection routine.

Finally, the last block of this part of the system is the routine when a message arrives from the MQTT topic. When a message is received, the MCU immediately translates it from char to string. Right after, there is a series of checks the processor makes to review the message's content. It was standardized that the virtual controller sends three types of messages: departing, arriving, or "else" (which is anything else than the first two messages). For each of them, there is a coded action plan detailed in figure 5.11. There is a block of these for each spot, so the MCU subscribes to three MQTT topics to receive messages from the virtual controller.

As the demands for the physical asset controller in this work are not very high, many different commercial microcontrollers, with WiFi connection and enough memory, running on possibly any programming language would fit this application. ESP8266 was chosen due to its opportunity for simple coding in both language and software, pinout characteristics, and availability in the laboratories. The microcontroller was coded and tested in parallel with the development of the other parts of this project. Its routines were validated through the serial monitor provided by the software, and tests with oscilloscopes in the laboratory.

Appendix B shows selected parts of the code used to achieve the results seen below. The first expected outcome was to successfully connect to the WiFi and MQTT server at the initialization. The following tables are results shown in the serial monitor at the starting of the MCU operation. The IP address is hidden.

```
Device: [ESP1]

Connecting to [agents].....
» WiFi connected
» IP address [192.xxx.x.x13]

Attempting MQTT connection
» Connected and subscribed to [IPB/SmartParking/Sector1/Spot1;
                               IPB/SmartParking/Sector1/Spot2;
                               IPB/SmartParking/Sector1/Spot3]
```

When the MCU receives a message from the MQTT topic in which it is subscribed, the following information is obtained from the serial monitor. It contains all possibilities of messages and its actions. The locker routine, ID publishing, and MQTT message treatment are documented in Appendix B.

Received "arriving"

In this situation, the MCU received a message warning that a user is ready to lock a bicycle in a specific spot. The first situation illustrates the action of the controller when the spot is vacant. Next, the same message, but for an occupied spot. The second situation should not occur if the flags defined in the code are properly updated.

```
Message arrived from: [IPB/SmartParking/Sector1/Spot2]
» msg: "arriving"
bikeinplace2 == false
» Arriving in [k2].
» Initiate locker routine.
```

```
Message arrived from: [IPB/SmartParking/Sector1/Spot2]
» msg: "arriving"
bikeinplace2 == true
» Spot [k2] already occupied.
» Warn the virtual controller.
```

Received "departing"

The same situations addressed above, but now the microcontroller received a message warning that the user wishes to unlock its bicycle. Again, the second situation is not expected to occur if the flags are correctly defined.

```
Message arrived from:          [IPB/SmartParking/Sector1/Spot2]
» msg: "departing"
bikeinplace2 == true
» Departing from [k2].
» Initiate locker routine.
```

```
Message arrived from:          [IPB/SmartParking/Sector1/Spot2]
» msg: "departing"
bikeinplace2 == false
» Spot [k2] already empty.
» Warn the virtual controller.
```

Else

It was also standardized that the virtual controller, cloud-connected Raspberry Pi, publishes a message containing the user's ID after the bicycle locking. Then, the physical asset controller, ESP8266, reads it and publishes back for spots management use.

```
Message arrived from: [IPB/SmartParking/Sector1/Spot2]
» msg: ID
» ID: [msg].
» Publishing to:      [IPB/SmartParking/ID].
```

Chapter 6

Conclusions and future work

Smart parking systems are a proper solution for many traffic-related large cities problems such as air and noise pollution, fuel spendings and the resultant GHG emissions, and the considerable drivers time spent searching for a vacant parking spot. Considering this background scenario, Cyber-physical Systems emerge as a suitable solution for the implementation of such systems. This work uses CPS allying IoT techniques along with established embedded electronics materials to conceptualize the deployment of a fully automated bicycle parking spot, working from an already deployed multi-agent architecture.

To this end, this work first ensured proper communication using the implemented MQTT protocol through WiFi means. Then, several options for the actuator's project obtained from the literature review or designed from scratch were explored, focusing on its physical operation validation. These options then went through software tests such as magnetostatics and electronic circuit simulations for the magnetic circuits and electronic drivers, respectively. Previous works defined both the power supply as photovoltaic and microcontrollers (ESP8266 and Raspberry Pi). This project dimensioned the photovoltaic panels for electricity supply based on the expected electricity demand from the actuator, MCU, and electronic drivers project. Also, it implemented ESP's routine software.

The results show that a set of different actuators are possible validated solutions for the deployment of a bicycle's smart parking. Stepper motors are recommended once they provide safety and reliability but are considerably expensive, while crafted options are cheaper but offer many implementation uncertainties and greatly add to the project development's time. The mechanical, electronic and magnetic validations are illustrated in multiple figures and graphic plots obtained from SolidWorks, LT Spice, and FEMM simulations. ESP's routine seems to work as expected, reaching all the objectives defined for this project during laboratory tests.

Initially, this project's goal was to implement a complete automated spot for the deployment of a Smart Parking System. Due to uncontrollable difficulties, pieces of the project converted to validation, as stated in this document. From the agents to the parking spot itself, this work successfully implemented communication between a cloud-based Raspberry Pi and the physical asset controller ESP8266 using an MQTT broker as an intermediate. Also, all routines of the PAC were coded and run as expected, including sending and receiving messages, treating commands received from the cloud, and sending PWM signals to the electronic actuators. The actuator options were validated through 3D design and magnetic simulation software, as well as a 3D printed prototype for the linear-to-rotational mechanism, and the electronic circuitry was verified using circuits simulation software.

Future work should treat this document as a source of inspiration and simulation development. The future of this smart parking project would be a physical implementation, not done here due to COVID-19 setbacks, and a full integration between the low-level electronics and the cloud computing agent-based software. The first step is to choose between the actuator options named here and deploy it, then move on to testing the system and integrating it with the mobile application, ultimately obtaining a functional arrangement. The ultimate goal is to implement and validate this project both in the campus and society environments, so students from IPB or the general public would have access to a reliable, safe, effective, and time-saving system.

Bibliography

- [1] C. Savaglio, G. Fortino, M. Ganzha, M. Paprzycki, C. Bădică and M. Ivanović, “Agent-based Internet of Things: State-of-the-art and research challenges,” *Future Generation Computer Systems*, 2019. DOI: 10.1016/j.future.2019.09.016.
- [2] P. Patel and D. Cassou, “Enabling high-level application development for the Internet of Things,” *The Journal of Systems Software*, 2015. DOI: 10.1016/j.jss.2015.01.027.
- [3] T. W. Hnat, T. I. Sookoor, P. Hooimeijer, W. Weimer and K. Whitehouse, “MacroLab: A vector-based macroprogramming framework for Cyber-Physical Systems,” *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, 2008. DOI: 10.1145/1460412.1460435.
- [4] D. Legatiuk, M. Theiler, K. Dragos and K. Smarsly, “A categorical approach towards metamodeling Cyber-Physical Systems,” *The 11th International Workshop on Structural Health Monitoring (IWSHM)*, 2017. DOI: 10.12783/shm2017/13908.
- [5] Y. Geng and C. G. Cassandras, “A new Smart Parking System based on resource allocation and reservations,” *IEEE Transactions on Intelligent Transportation Systems*, 2013. DOI: 10.1109/TITS.2013.2252428.
- [6] United Nations (UN): Habitat, “Urbanization and development: Emerging futures,” Tech. Rep., 2016. [Online]. Available: <https://unhabitat.org/sites/default/files/download-manager-files/WCR-2016-WEB.pdf>.

- [7] G. Frame, A. Ardila-Gomez and Y. Chen, “The kingdom of the bicycle: what Wuhan can learn from Amsterdam,” *World Conference on Transport Research (WCTR)*, 2016. DOI: 10.1016/j.trpro.2017.05.203.
- [8] D. Roberts, “Cars dominate cities today. Barcelona has set out to change that,” *Vox*, Sep. 2019. [Online]. Available: <https://www.vox.com/energy-and-environment/2019/4/8/18273893/barcelona-spain-urban-planning-cars>.
- [9] A. Athira, S. Lekshmi, P. Vijayan and B. Kurian, “Smart Parking System based on optical character recognition,” *3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019. DOI: 10.1109/ICOEI.2019.8862517.
- [10] N. Promy and S. Islam, “A smart Android-based parking system to reduce the traffic congestion of Dhaka City,” *21st International Conference on Advanced Communication Technology (ICACT)*, 2019. DOI: 10.23919/ICACT.2019.8701935.
- [11] L. Sakurada, J. Barbosa and P. Leitão, “Deployment of industrial agents in a Smart Parking System,” *17th International Conference on Industrial Informatics (INDIN)*, 2019. DOI: 10.1109/INDIN41052.2019.8972190.
- [12] L. Sakurada, J. Barbosa, P. Leitão, G. Alves, A. P. Borges and P. Botelho, “Development of Agent-based CPS for Smart Parking Systems,” *45th Annual Conference of the IEEE Industrial Electronics Society*, 2019. DOI: 10.1109/IECON.2019.8926653.
- [13] D. Nunes, J. S. Silva and F. Boavida, *A practical introduction to human-in-the-loop Cyber-Physical Systems*, 1st. Wiley - IEEE Press, 2018, ch. 01, ISBN: 978-1-11-937779-5.
- [14] M. Kraeling and M. Brogioli, *Software Engineering for Embedded Systems*, 2nd. Newnes, 2019, ch. 07, ISBN: 978-0-12-809448-8.
- [15] R. I. Sokullu and A. Balcı, *Ambient Assisted Living and Enhanced Living Environments*, 1st. Butterworth-Heinemann, 2017, ch. 17, ISBN: 978-0-12-805195-5.

- [16] C. Antón-Haro and M. Dohler, *Machine-to-machine (M2M) Communications*, 1st. Woodhead Publishing, 2015, ch. 01, ISBN: 978-1-78242-102-3.
- [17] M. Törngren, F. Asplund, S. Bensalem, J. McDermid, R. Passerone, H. Pfeifer, A. Sangiovanni-Vincentelli and B. Schätz, *Cyber-physical systems*, 1st. Academic Press, 2017, ch. 01, ISBN: 978-0-12-803801-7.
- [18] F. Tao, M. Zhang and A.Y.C.Nee, *Digital Twin driven Smart Manufacturing*, 1st. Academic Press, 2019, ch. 12, ISBN: 978-0-12-817630-6.
- [19] R. Crowder, *Electric Drives and Electromechanical Systems*, 2nd. Butterworth-Heinemann, 2019, ch. 11, ISBN: 978-0-08-102884-1.
- [20] H. Boyes, B. Hallaq, J. Cunningham and T. Watson, “The Industrial Internet of Things (IIOT): An analysis framework,” *Computers in Industry*, 2018. DOI: 10.1016/j.compind.2018.04.015.
- [21] T. Lins and R. A. R. Oliveira, “Cyber-physical production systems retrofitting in context of Industry 4.0,” *Computers Industrial Engineering*, 2020. DOI: 10.1016/j.cie.2019.106193.
- [22] M. Hermann, T. Pentek and B. Otto, “Design principles for Industrie 4.0 scenarios,” *49th Hawaii International Conference on System Sciences*, 2016. DOI: 10.1109/hicss.2016.488.
- [23] German Federal Ministry of Education and Research, “Recommendations for implementing the strategic initiative Industrie 4.0,” Tech. Rep., Apr. 2013. [Online]. Available: <https://www.din.de/blob/76902/e8cac883f42bf28536e7-e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf>.
- [24] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld and M. Hoffmann, “Industry 4.0,” *Business Information Systems Engineering*, 2014. DOI: 10.1007/s12599-014-0334-4.

- [25] H. Song, R. Srinivasan, T. Sookoor and S. Jeschke, *Smart Cities: Foundations, Principles, and Applications*, 1st. Wiley Telecom, 2017, ch. 22 and 26, ISBN: 978-1-11-922644-4.
- [26] H. T. Mouftah, M. Erol-Kantarci and M. H. Rehmani, *Machine-to-Machine Communications in the Smart City — a Smart Grid Perspective*, 1st. Wiley Telecom, 2019, ch. 04, ISBN: 978-1-11-936012-4.
- [27] Q. F. Hassan, *Internet of Things A to Z: Technologies and Applications*, 1st. Wiley-IEEE Press, 2018, ch. 12, ISBN: 978-1-11-945673-5.
- [28] W. Imran, Z. H. Khan, T. A. Gulliver, K. S. Khattak and H. Nasir, “A macroscopic traffic model for heterogeneous flow,” *Chinese Journal of Physics*, 2019. DOI: 10.1016/j.cjph.2019.12.005.
- [29] S. P. Hoogendoorn and P. H. L. Bovy, “State-of-the-art of vehicular traffic flow modelling,” *Journal of Systems and Control Engineering*, 2001. DOI: 10.1177/095965180121500402.
- [30] Stacy C. Davis, and Robert G. Boundy, “Transportation energy data book,” Tech. Rep., Jan. 2020. DOI: 10.2172/1606919.
- [31] G. R. Timilsina and H. B. Dulal, “Urban road transportation externalities: Costs and choice of policy instruments,” *Oxford University Press for The World Bank Research Observer*, 2011. DOI: 10.1093/wbro/lkq005.
- [32] The World Bank, “World development indicators,” Tech. Rep., Jan. 2020. [Online]. Available: <http://datatopics.worldbank.org>.
- [33] H. Hao, Y. Geng and J. Sarkis, “Carbon footprint of global passenger cars: Scenarios through 2050,” *Energy*, 2015. DOI: 10.1016/j.energy.2016.01.089.
- [34] The Editorial Board, “Europe leads the world with its climate mission,” *Financial Times*, Dec. 2019. [Online]. Available: <https://www.ft.com/content/37db6aea-1c35-11ea-97df-cc63de1d73f4>.

- [35] European Environment Agency (EEA), “Greenhouse gas emission trend and target,” Tech. Rep., Dec. 2019. [Online]. Available: <https://www.eea.europa.eu/data-and-maps/figures/greenhouse-gas-emission-trend-projections>.
- [36] D. C. Shoup, “Cruising for parking,” *Transport Policy*, vol. 13, 2006. DOI: 10.1016/j.tranpol.2006.05.005.
- [37] —, *Parking*, 1st. Elsevier, 2020, ch. 01, ISBN: 978-0-12-815265-2.
- [38] INRIX Research, “Global traffic scorecard,” Tech. Rep., Mar. 2020. [Online]. Available: <https://inrix.com/scorecard/>.
- [39] G. Duranton and M. A. Turner, “The Fundamental Law of Road Congestion: Evidence from US cities,” *American Economic Review*, 2011. DOI: 10.1257/aer.101.6.2616.
- [40] New York City Department of Transportation, “Protected bicycle lanes in NYC,” Tech. Rep., Sep. 2014. [Online]. Available: <http://www.nyc.gov/html/dot/downloads/pdf/2014-09-03-bicycle-path-data-analysis.pdf>.
- [41] City of Seattle, “New protected bike lane triples bike riding on Second Avenue,” Tech. Rep., Sep. 2014. [Online]. Available: <http://www.seattle.gov/news/detail.asp?ID=14619&dept=19>.
- [42] H. Li, D. J. Graham and P. Liu, “Safety effects of the London cycle superhighways on cycle collisions,” *Accident Analysis and Prevention*, 2017. DOI: 10.1016/j.aap.2016.11.016.
- [43] M. A. Razzaque and S. Clarke, “Smart management of next generation bike sharing systems using Internet of Things,” *IEEE First International Smart Cities Conference (ISC2)*, 2015. DOI: 10.1109/isc2.2015.7366219.
- [44] L. F. Luque-Vega, D. A. Michel-Torres, E. Lopez-Neri, M. A. Carlos-Mancilla and L. E. González-Jiménez, “IoT Smart Parking system based on the visual-aided smart vehicle presence sensor: SPIN-V,” *MDPI Sensors*, 2020. DOI: 10.3390/s20051476.

- [45] D. N. C. Loong, S. Isaak and Y. Yusof, "Machine vision-based Smart Parking system using Internet of Things," *Telkomnika*, 2019. DOI: 10.12928/TELKOMNIKA.v17i4.12772.
- [46] D. Angulo-Esguerra, C. Villate-Barrera, W. Giral, H. C. Florez, A. T. Zona-Ortiz and F. Diaz-Sanche, "Parkurbike: An IoT-based system for bike parking occupation checking," *IEEE Colombian Conference on Communications and Computing (COLCOM)*, 2017. DOI: 10.1109/colcomcon.2017.8088201.
- [47] M. Karthi and P. Harris, "Smart Parking with reservation in Cloud-based environment," *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2016. DOI: 10.1109/ccem.2016.038.
- [48] H. Wang and W. He, "A reservation-based Smart Parking system," *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011. DOI: 10.1109/infcomw.2011.5928901.
- [49] K. L. Ainsworth and M. B. Beezer, "Bicycle security system," Tech. Rep., 1994. [Online]. Available: <https://patents.google.com/patent/US5278538A>.
- [50] G. A. Duncan, A. J. Feldman and A. J. Stone, "Wireless portable lock system," Tech. Rep., 2017. [Online]. Available: <https://patents.google.com/patent/US9679429B2/>.
- [51] T. Becker, "Lock for bicycle," Tech. Rep., 2011. [Online]. Available: <https://patents.google.com/patent/EP1760232B1>.
- [52] J.-C. Gagosz and E. Zeferino, "Automatic bicycle storage system," Tech. Rep., 2013. [Online]. Available: <https://patents.google.com/patent/EP1820722B2>.
- [53] P. Leitão and S. Karnouskos, *Industrial Agents: Emerging applications of software Agents in industry*, 1st. Elsevier, 2015, ch. 04, ISBN: 978-0-12-800341-1.
- [54] S. Lee, H. Kim, D.-k. Hong and H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," *The International Conference on Information Networking*, 2013. DOI: 10.1109/IC0IN.2013.6496715.

- [55] M. Yuan, “Getting to know MQTT,” *IBM Developer*, Jan. 2020. [Online]. Available: <https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/>.
- [56] A. E. Fitzgerald, C. Kingsley Jr., and S. D. Umans, *Electric Machinery*, 6th. McGraw Hill, 2003, ISBN: 0-07-366009-4.
- [57] Electrofun, “NEMA17: Motor de passo,” Tech. Rep., 2020. [Online]. Available: <https://www.electrofun.pt/motores-steppers/nema17-motor-de-passo>.
- [58] ON semiconductors, “2N6052-D – Darlington complementary silicon power transistors,” Tech. Rep., Sep. 2008. [Online]. Available: <https://www.onsemi.cn/PowerSolutions/document/2N6052-D.PDF>.
- [59] —, “FCB125N65S3 – Power, N-Channel, SUPERFET III, Easy Drive,” Tech. Rep., Feb. 2020. [Online]. Available: <https://www.onsemi.com/pub/Collateral/FCB125N65S3-D.PDF>.
- [60] Adafruit Learning System, “Adafruit Feather HUZZAH ESP8266,” Tech. Rep., Mar. 2020. [Online]. Available: <https://pt.mouser.com/datasheet/2/737/adafruit-feather-huzzah-esp8266-1396527.pdf>.
- [61] altE store, “Off-grid PV panels and battery pack calculator,” Tech. Rep., 2020. [Online]. Available: https://www.altestore.com/store/calculators/off_grid_calculator/.
- [62] Weather and Climate, “Climate in Bragança, Portugal,” Tech. Rep., 2019. [Online]. Available: <https://weather-and-climate.com/average-monthly-Rainfall-Temperature-Sunshine,Braganca,Portugal>.

Appendix A

FEMM data

Table A.1: Mechanism's magnetic simulation data from FEMM treated with UARM eq.

x (mm)	F (N)	$F_{res}(N)$	a (m/s ²)	s (m/s)	t (s)	$\Sigma t(s)$
0.00	30.06	25.16	50.310	0.000000	0.000000	0.000000
0.50	30.31	25.41	50.810	0.224299	0.004414	0.004414
1.00	30.50	25.60	51.190	0.317994	0.001830	0.006245
1.50	30.61	25.71	51.410	0.390269	0.001406	0.007651
2.00	30.79	25.89	51.770	0.451354	0.001180	0.008831
2.50	30.87	25.97	51.930	0.505460	0.001042	0.009872
3.00	30.85	25.95	51.890	0.554455	0.000944	0.010817
3.50	30.94	26.04	52.070	0.599425	0.000864	0.011680
4.00	30.82	25.92	51.830	0.641389	0.000810	0.012490
4.50	30.66	25.76	51.510	0.680595	0.000761	0.013251
5.00	30.52	25.62	51.230	0.717440	0.000719	0.013970
5.50	30.21	25.31	50.610	0.752296	0.000689	0.014659
6.00	29.90	25.00	49.990	0.785213	0.000658	0.015318
6.50	29.51	24.61	49.210	0.816425	0.000634	0.015952
7.00	29.13	24.23	48.450	0.846026	0.000611	0.016563

Table A.1 continued from previous page

7.50	28.62	23.72	47.430	0.874191	0.000594	0.017157
8.00	28.04	23.14	46.270	0.900911	0.000577	0.017734
9.00	26.71	21.81	43.610	0.950168	0.000549	0.018847
9.50	26.06	21.16	42.310	0.972846	0.000536	0.019383
10.00	25.39	20.49	40.970	0.994354	0.000525	0.019908
10.50	24.76	19.86	39.710	1.014746	0.000514	0.020421
11.00	24.09	19.19	38.370	1.034128	0.000505	0.020926
11.50	23.46	18.56	37.110	1.052516	0.000496	0.021422
12.00	22.85	17.95	35.890	1.070000	0.000487	0.021909
12.50	22.33	17.43	34.850	1.086642	0.000478	0.022387
13.00	21.72	16.82	33.630	1.102561	0.000473	0.022860
14.00	20.68	15.78	31.550	1.132652	0.000954	0.023814
15.00	19.70	14.80	29.590	1.160172	0.000930	0.024744
16.00	18.76	13.86	27.710	1.185403	0.000911	0.025654
17.00	17.98	13.08	26.150	1.208553	0.000885	0.026540
18.00	17.18	12.28	24.550	1.230000	0.000874	0.027413
19.00	16.46	11.56	23.110	1.249800	0.000857	0.028270
20.00	15.80	10.90	21.790	1.268156	0.000842	0.029112
21.00	15.15	10.25	20.490	1.285224	0.000833	0.029945
22.00	14.54	9.64	19.270	1.301069	0.000822	0.030768
23.00	13.96	9.06	18.110	1.315796	0.000813	0.031581
24.00	13.44	8.54	17.070	1.329489	0.000802	0.032383
25.00	12.91	8.01	16.010	1.342267	0.000798	0.033181
26.00	12.43	7.53	15.050	1.354142	0.000789	0.033970
27.00	11.98	7.08	14.150	1.365211	0.000782	0.034752
28.00	11.53	6.63	13.250	1.375536	0.000779	0.035532
33.00	9.56	4.66	9.310	1.422884	0.005086	0.040617

Table A.1 continued from previous page

38.00	7.90	3.00	5.990	1.455232	0.005400	0.046018
43.00	6.51	1.61	3.210	1.475669	0.006367	0.052384
44.00	6.27	1.37	2.730	1.477843	0.000796	0.053181
45.00	6.01	1.11	2.210	1.479689	0.000835	0.054016
46.00	5.76	0.86	1.710	1.481182	0.000873	0.054889
47.00	5.55	0.65	1.290	1.482336	0.000895	0.055784
48.00	5.24	0.34	0.670	1.483206	0.001298	0.057082
49.00	5.02	0.11	0.230	1.483658	0.001964	0.059046
50.00	4.80	-0.11	-0.210	1.483813	0.000738	0.059784
51.00	4.53	-0.38	-0.750	1.483671	0.000189	0.059973
52.00	4.30	-0.61	-1.210	1.483166	0.000418	0.060390
53.00	4.07	-0.84	-1.670	1.482349	0.000489	0.060879
54.00	3.84	-1.07	-2.130	1.481222	0.000529	0.061408
55.00	3.63	-1.28	-2.550	1.479784	0.000564	0.061972
56.00	3.40	-1.51	-3.010	1.478060	0.000573	0.062545
57.00	3.20	-1.71	-3.410	1.476022	0.000598	0.063143
58.00	2.98	-1.93	-3.850	1.473710	0.000601	0.063743
68.00	0.93	-3.98	-7.950	1.447349	0.003316	0.067059
78.00	0.00	-4.91	-9.810	1.391337	0.005710	0.072769
88.00	0.00	-4.91	-9.810	1.318947	0.007379	0.080148
98.00	0.00	-4.91	-9.810	1.242345	0.007809	0.087957
108.00	0.00	-4.91	-9.810	1.160698	0.008323	0.096279
118.00	0.00	-4.91	-9.810	1.072856	0.008954	0.105234
128.00	0.00	-4.91	-9.810	0.977149	0.009756	0.114990
138.00	0.00	-4.91	-9.810	0.870988	0.010822	0.125812
148.00	0.00	-4.91	-9.810	0.749947	0.012339	0.138150
158.00	0.00	-4.91	-9.810	0.605161	0.014759	0.152909

Appendix B

ESP8266 code

MCU receives message from MQTT topic

Only the code for a single spot is shown here since the others are copies replacing, for example, `topic_sub1` for `topic_sub2`. The code and indentation are simplified to save space.

```
void callback(char* topic, byte* payload, unsigned int length) {
    String msg = toString(payload, length); // Translate the message to String

    if (topic == topic_sub1) {
        if (msg == arriving_k) {
            if (state1 == true && bikeinplace1 == false) {
                Serial.println(">> Arriving in [k1].\n");
                bikeinplace1 = true;
            }
        }
        else if (state1 == false && bikeinplace1 == false) {
            locker(1);
            state1 = true;
            Serial.println(">> Arriving in [k1].\n");
            bikeinplace1 = true;
        }
    }
}
```

```

    }
    else {
        Serial.println(">> Spot [k1] already occupied.\n");
    }
}
else if (msg == departing_k) {
    if (state1 == false && bikeinplace1 == true) {
        locker(1);
        state1 = true;
        bikeinplace1 = false;
        Serial.println(">> Departuring from [k1].\n");
    }
    else if (state1 == true && bikeinplace1 == true) {
        Serial.println(">> Spot [k1] already opened.\n");
    }
    else {
        Serial.println(">> Spot [k1] already empty.\n");
    }
}
else {
    Serial.print(">> ID: [");
    Serial.print(msg);
    Serial.println("] ");
    String ID_k1 = msg;
    String topic_publish = "IPB/SmartParking/" + msg;
    client.publish((char*) topic_publish.c_str(), (char*) msg.c_str());
}
}
}

```

Functions `locker()` and `verify_spots()`

Functions to respectively feed the solenoid's winding for a time and verify the spots availability. The variable `timetotap` is an int defined in milliseconds and consists of the time demanded for the locker's proper mechanical operation. The redundant code is simplified to save space.

```
void locker(int k) {
  switch (k) {
    case 1:
      analogWrite(D1, duty);
      delay(timetotap);
      break;
    case 2:
      ...
    case 3:
      ...
    default:
      Serial.println("Unknown error in switch case (locker function).");
      break;
  }
}
```

```
void verify_spots() {
  if (pushbutton = true) {
    if (digitalRead(D5) == HIGH) {
      bikeinplace1 = true;
      state1 = false;
      Serial.println("There is a parked bike in spot [k1]");
    }
  }
}
```

```

    if (digitalRead(D6) == HIGH) {
        ...
    }
    if (digitalRead(D7) == HIGH) {
        ...
    }
}
else {
    if (digitalRead(D5) == LOW) {
        bikeinplace1 = true;
        state1 = false;
        Serial.println("There is no parked bike in spot [k1]");
    }
    if (digitalRead(D6) == LOW) {
        ...
    }
    if (digitalRead(D7) == LOW) {
        ...
    }
}
}
}

```