Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Radio Engineering

Mobile-Robot and Platform for VLC Indoor Navigation

Master's thesis

Bc. Štěpán Bosák

Master's program: Electronics and Communications
Specialisation: Radio Communications and Systems
Supervisor CTU: doc. Ing. Stanislav Vítek, Ph.D.
Supervisor NTUST: Professor, Jenq-Shiou Leu, Ph.D.

Prague, January 2023

Thesis Supervisors:

doc. Ing. Stanislav Vítek, Ph.D.
Department of Radio Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
160 00 Prague 6
Czech Republic

Professor, Jenq-Shiou Leu, Ph.D.
MIT Laboratory
Department of Electronic and Computer Engineering
National Taiwan University of Science and Technology
No.43, Keelung Rd., Sec.4, Da'an Dist.
Taipei City 106335
Taiwan (R.O.C.)

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Bosák Štěpán**	Personal ID number:	**474230**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Electromagnetic Field**		
Study program:	**Electronics and Communications**		
Specialisation:	**Radio Communications and Systems**		

II. Master’s thesis details

Master’s thesis title in English:

Mobile-Robot and Platform for VLC Indoor Navigation

Master’s thesis title in Czech:

Mobilní robot a platforma pro indoor VLC navigaci

Guidelines:

Design and implement tools to verify VLC indoor navigation algorithms. Follow the following guidelines to develop your work:
1 Design and implement the HW for the VLC beacon,
2. Design and implement a semi-automated mobile robotic platform.
3. Conduct a research on VLC indoor localization algorithms. Based on the research, design and implement a suitable VLC receiver.
4. Propose methods to preprocess the data from the VLC receiver.

Bibliography / sources:

[1] GUAN, Weipeng, et al. Indoor Localization System of ROS mobile robot based on Visible Light Communication. arXiv preprint arXiv:2001.01888, 2020.
[2] GUAN, Weipeng, et al. High-precision indoor positioning algorithm based on visible light communication using complementary metal–oxide–semiconductor image sensor. Optical Engineering, 2019, 58.2: 024101.

Name and workplace of master’s thesis supervisor:

doc. Ing. Stanislav Vítek, Ph.D.   Department of Radioelectronics FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **07.02.2022**   Deadline for master's thesis submission: **10.01.2023**

Assignment valid until: **30.09.2023**

doc. Ing. Stanislav Vítek, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt	Student’s signature

Acknowledgements

I would like to express my deep gratitude to Associate Professor Stanislav Vítek, my research supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. I would also like to thank Professor Jenq-Shiou Leu, for his advice and assistance. My grateful thanks are also extended to my colleague Martin Suda for his help and cooperation on the project. Special thanks should be given to Mr. Benjamin Jackson for his professional language guidance and support. Finally, I wish to thank my family and friends for their support and encouragement throughout my study.

Declaration

I declare that I have written submitted thesis by myself and that I have listed all information sources in accordance with Methodical Guideline on Compliance with Ethical Principles.

Prague, January 2023

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, Leden 2023

....

Bc. Štěpán Bosák

Abstract

The inherent advantages of Visible Light Communication (VLC), such as high throughput and security, have heightened the interest among researchers in studying it, especially in terms of Visible Light Positioning (VLP). The research has resulted in excellent positioning systems and algorithms with a centimetre localization accuracy. However, most researchers focus on presenting their newly designed algorithms instead of describing and analyzing the utilized workspace and testing environments. Moreover, most papers build use-case specific testbeds. The system implementation and hardware features affect the VLP system's performance and cost, and determine its applicability in the commercial sphere. This thesis involves a VLC transmitter node, including electrical simulations and electrical and thermal measurements. The study shows the realization of a node and the testing environment.

Meanwhile, categorizing the node and theoretically evaluating all recent and available positioning algorithms makes the node easily modifiable and adaptable.

The second objective of the research is to select a proper VLC receiver and construct a robotic platform. This paper proposes a robotic device based on an ESP32 microcontroller. The robot integrates a light sensor, enabling VLP and navigation algorithms testing. The designed firmware collects the sensor's data and controls the robot wirelessly.

Lastly, all gathered data and the robot's controls are accessible through a web browser application. The research in collaboration with Suda, M. processes the gathered image data. The web application utilizes the architecture designed by Suda, M.

Keywords: Visible Light Communication (VLC), Visible Light Positioning (VLP), Navigation, Indoor, Transmit/ Transmitting/ Transmitter (TX), Receive/ Receiving/ Receiver (RX), Localization, Hardware design, Light-Emitting Diode (LED), Simulations, Measurements, Robot, Firmware, Control application, Wireless communications, Wireless Fidelity (Wi-Fi).

Abstrakt

Primární výhody komunikace ve viditelném světle, jako je vysoká propustnost a její bezpečnost, způsobily nárůst zájmu vědecké komunity, a to především v oblasti lokalizace. Dnes máme k dispozici vynikající lokalizační systémy a algoritmy s centimetrovou přesností. Většina výzkumů se však zaměřuje na zdokonalování navigačních, respektive lokalizačních algoritmů a nezohledňuje popis použitého testovacího prostředí. Navíc, často výzkumné týmy konstruují jednoúčelová testovací prostředí, která nejsou použitelná v komerční oblasti a nezohledňují adaptivnost systému. Tato diplomová práce ukazuje návrh vysílače pro komunikace ve viditelném světle včetně simulací, měření a realizace.

Sestavený vysílač je dále kategorizován podle nejnovějších průzkumů a jsou zhodnoceny jeho limity a schopnosti.

Druhá část práce se věnuje příjímací straně. Na základě předchozího zhodnocení je vybrán vhodný přijímač a zkonstruován dálkově ovládatelný robot. Robot využívá mikrokontrolér ESP32 s Complementary Metal-Oxide-Semiconductor (CMOS) optickým senzorem. Součástí robota je také program, který sbírá data ze senzorů robota a umožňuje jeho dálkové ovládání.

V závěru práce je připojeno rozšíření do aplikace, kterou navrhl Suda M. v jeho diplomové práci. Toto rozšíření respektuje architekturu aplikace a dále doplňuje intuitivní ovládání robota a zobrazení nasbíraných dat.

Klíčová slova: Komunikace ve viditelném světle, navigace, lokalizace, hardware design, firmware design, simulace, měření, robot, kontrolní aplikace, bezdrátové komunikace.

Contents

List of Figures

List of Tables

List of source code examples

List of Algorithms

List of Acronyms

AC/DC	Alternating Current to Direct Current	5, 13, 17 f.
ADC	Analog-to-Digital Converter	30
AoA	Angle of Arrival	23 ff., 27
AP	Access Point	1, 30, 43, 48, 53
BER	Bit Error Rate	25
BFSK	Binary Frequency Shift Keying	25
BLE	Bluetooth Low Energy	1, 30
BPSK	Binary-Phase Shift Keying	25
BT	Bluetooth	1, 22, 30
CCD	Charge Coupled Devices	28
CCW	Counterclockwise	34
CMMR	Common-Mode Rejection Ratio	7
CMOS	Complementary Metal-Oxide-Semiconductor	vi, 10, 24, 28 ff., 43, 59
CPU	Central Processing Unit	24, 30
CW	Clockwise	34
DAC	Digital-to-Analogue Converter	5 f., 28, 30
DC	Direct Current	13, 15, 33 ff., 59
DC/DC	Direct Current to Direct Current	13 f., 35, 38, 41
DMIPS	Dhrystone Million Instructions per Second	30
DSP	Digital Signal Processor	30
EKF	Extended Kalman Filter	27
EMC	Electromagnetic compatibility	35
ER	Extinction Ratio	28 f.
ESD	Electrostatic Discharge	13 ff.
FoV	Field of View	22
FPGA	Field Programmable Gate Arrays	33
GO	Geometric Optics	22
GPS	Global Positioning System	1, 25
HTTP	HyperText Transfer Protocol	3, 44 f., 48, 50, 59
IC	Integrated Circuit	14, 34
IoT	Internet of Things	1
IPS	Indoor Positioning System	1
JSON	JavaScript Object Notation	xii, 46, 49 f., 52, 56 f.
LAN	Local Area Network	53
LBS	Location-Based Service	25
LDMOS	laterally-diffused metal-oxide-semiconductor	34
LED	Light-Emitting Diode	v, 1, 5, 7, 9 f., 13, 17 ff., 21–26, 28, 38, 41, 43–49, 53, 55 ff.
LMS	Least Mean Square	25
LoRa	Long Range	3, 5 f., 13
LoRaWAN	Long Range Wide Area Network	3, 5 f.
LP	Low-Pass filter	6, 14
NMOS	N-type Metal-Oxide-Semiconductor	6 ff., 16, 18 f., 59
OFDM	Orthogonal Frequency Division Multiplexing	25
OOC	Optical Orthogonal Codes	25

OPA	Operational Amplifier	6 f., 10, 13 ff., 59
OTA	Over-The-Air	43, 46, 53, 59
OWL	Optical Wireless Location	26
PCB	Printed Circuit Board	18 f., 35
PSD	Power Spectral Density	23
PSRAM	Pseaudostatic Random Access Memory	30
PWM	Pulse Width Modulation	6, 10, 12, 30, 33 f., 45, 47
RF	Radio Frequency	1
RFID	Radio Frequency Identification	1, 22
RSS	Received Signal Strength	1, 21–27, 36, 38, 59
RSSI	Received Signal Strength Indication	1
RSSR	Received Signal Strength Ratio	25
RTOS	Real-Time Operating System	30
RX	Receive/Receiving/Receiver	v, 22, 25 f.
SD	Secure Digital	30
SoC	System on a Chip	30
SPIFFS	Serial Peripheral Interface Flash File System	43 f., 51 f., 59
SPST	Single Pole Single Throw	38
SRAM	Static Random Access Memory	30
STA	Station Mode	30
SVM	Support Vector Machines	22
TCP/IP	Transmission Control Protocol/Internet Protocol	3
TDoA	Time Difference of Arrival	21, 24 f.
THT	Through-Hole Technology	13
ToA	Time of Arrival	23 ff.
TTL	Time To Live	31
TX	Transmit/Transmitting/Transmitter	v, 5, 22, 25
UHF	Ultra-High Frequency	1
UI	User Interface	55 ff., 59
URL	Uniform Resource Locator	46, 48 f., 51, 56
VLC	Visible Light Communication	v, 1, 5 f., 18, 21 ff., 27 ff., 59
VLP	Visible Light Positioning	v, 1 f., 21, 24, 27, 33, 59
Wi-Fi	Wireless Fidelity	v, xii, 1, 30, 35 f., 43, 46, 48 f., 51 ff., 59

Chapter 1

Introduction

Positioning services are the 21st-century hot research topic. The development of the Internet of Things (IoT) and its integration into the commercial sphere enabled positioning technologies in various fields. The Global Positioning System (GPS) secures outdoor real-time positioning and navigation. The applications range from personal portable devices to large devices such as aircraft and vehicles. However, in indoor environments, GPS positioning fails due to signal obstructions. [1] Therefore, adopting alternative technologies that overcome the issues of GPS is necessary.

Since wireless signals are critical for positioning, researchers utilize different technologies for Indoor Positioning Systems (IPSs), such as Wi-Fi, Bluetooth (BT), or Radio Frequency Identification (RFID). In [2], Ranimol et al. propose a Wi-Fi-based fingerprinting positioning system that achieves an accuracy of 93% using Received Signal Strength (RSS) from each Access Point (AP). In recent years, Bluetooth Low Energy (BLE) emerged, and devices integrating BLE have become affordable. Research [3] from 2021 applies four ESP32 BLE transmitters and one receiver. The receiver measures the Received Signal Strength Indication (RSSI) and transfers it into the distance. The target applies trilateration on the measured distances and gets positioned in a co-ordinate system. Paper by A. Vena, I. Illanes, L. Alidieres, B. Sorli and F. Perea [4] implemented an Ultra-High Frequency (UHF) RFID tracking and positioning system. The researchers propose a visitor tracking service in a real museum in Agde, France. The tracking results provide information to the museum staff about the visitor's interests.

Visible Light Positioning (VLP) systems emerged as an alternative solution to Radio Frequency (RF) localization in recent years. Visible light positioning utilizes Visible Light Communication (VLC) to provide localization services. VLC connects wireless devices equipped with a visible light receiver similarly to radio-wave utilizing methods with the visible light as the carrying medium. Apart from the communication, VLC provides the deployment environment with lighting conditions. Figure 1.1 shows good use of VLC in a living room. [5]

LEDs showed promising features regarding VLC. These features mainly are:

1. broad bandwidth (100's of MHz) and simple reception using a photodetector;
2. signal security - light does not penetrate solid opaque objects, and therefore rooms within indoor environments do not interfere with each other;
3. LEDs are energy efficient even for high data-rate transmissions. (88.3% higher efficiency than a halogen lightbulb [6]);
4. LED components neither get disturbed by nor produce radio or electromagnetic interference, which is crucial for applications in electromagnetic sensitive places such as hospitals;
5. cost-effectiveness and up to 10 years of reliable lifetime. [7]

Moreover, visible light signals possess extraordinary features, which are worth mentioning. These features are:

1. shorter wavelength than RF signals. The short wavelengths give light signals better immunity to multipath effects and increase the signal's predictability.
2. Light signals can provide both communication and lighting functions.

Precedent works of the research group focused on building a foundation for a visible light positioning testbed, including both the hardware [8] and software layer [9]. However, the proposed implementations suffered from multiple unwanted effects, which needed fixing in the practical realization. Most researches regarding VLC and VLP rarely discuss the implementation of the testing

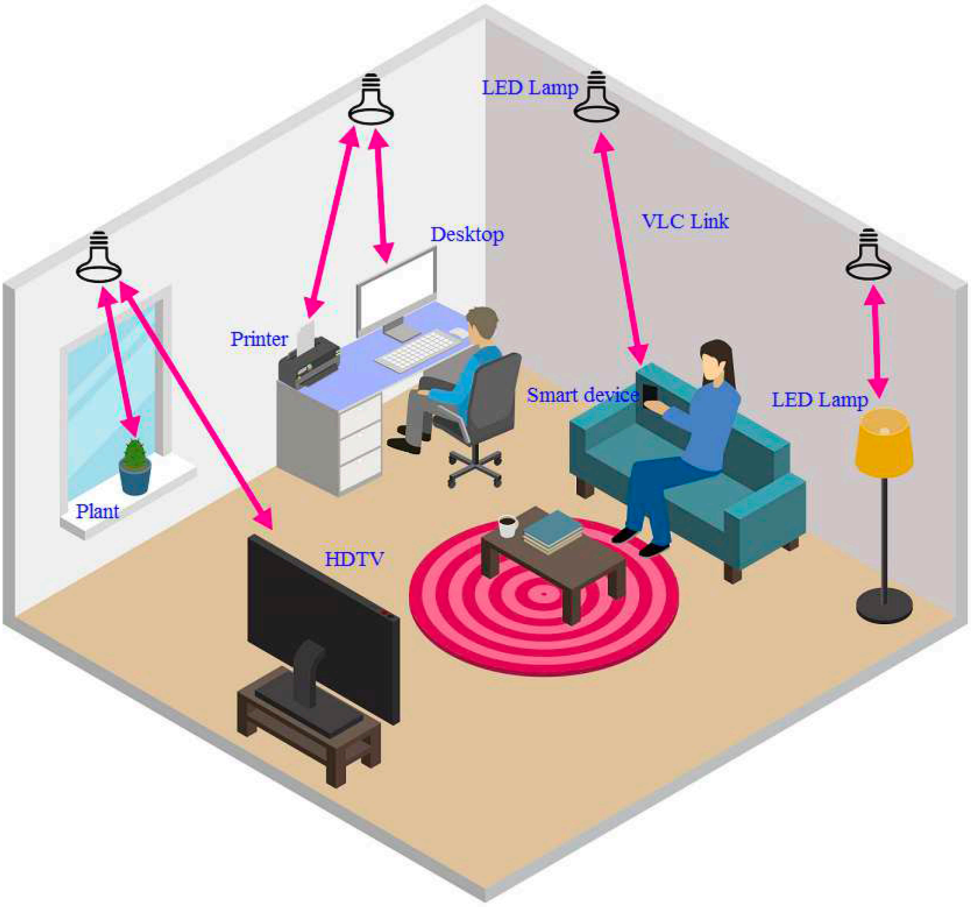


Figure 1.1: Diagram explaining the Visible Light Communication (VLC) as a communication medium. Adopted from [5].

environments and seldom present the actual hardware designs and documented code. This closed source approach makes testing new ideas in indoor positioning services tedious since every research group needs to build a testing environment from scratch. In paper [10], French scholars present a realization of such a testbed. However, the paper only shows the general idea of the realization without much detail.

The ultimate contribution of this work is to create a partially open-source realization of the platform that allows testing of various positioning algorithms and can be easily modified if the need arises.

This thesis makes the following contributions and arranges as follows:

1. Chapter 2 shows the hardware realization of a visible light transmitter. The chapter compares the transmitter from the previous research [8] and details the improvements in the design. Apart from the design and realization, it also depicts simulations of the module and electrical and thermal measurements.

2. Chapter 3 categorizes the proposed transmitter and digests the available VLP algorithms for the module based on the categorization. The localization algorithms divide into various groups according to the exploited positioning mechanisms. Lastly, the chapter depicts the selected receiving module for this research.

3. Chapter 4 documents the hardware design of a mobile robot, which serves as a visible light receiver. The part shows all applied electrical components and their interconnections, and describes their functions. All robots compose of moving parts. Therefore, sections of the chapter report the mechanical construction.

4. Chapter 5 delineates the software design of the robot's firmware. The platform uses *ESP32-CAM* [11] development module. The firmware uses the *Arduino* open-source framework, making the development process more straightforward.

5. Chapter 6 briefly documents a high-level application that allows the robot's remote control over Transmission Control Protocol/ Internet Protocol (TCP/ IP) and HyperText Transfer Protocol (HTTP). The robot's controls are only a tiny part of the application since the same application controls the testbed using Long Range (LoRa) as the physical layer and Long Range Wide Area Network (LoRaWAN) as the networking protocol.

6. Finally, Chapter 7 summarizes all contributions of the paper in detail and goes over all achieved results. Furthermore, the conclusion suggests the next steps of the research.

7. Appendix B stores additional documents about the project's realization.

Chapter 2

Hardware of the transmitting node

The transmitter stations are essential components of any VLC navigation system. Such stations provide the indoor space with comfortable lighting conditions and, more importantly, create a network of reference points for any VLC navigation algorithm. The two main parts of a VLC node are a microcontroller and a LED driver with a high power LED. [6]

Block diagram 2.1 shows a generic architecture of a VLC TX node. The microcontroller is a generator of analogue signals, either amplified or converted by the LED driver to a suitable form. The LED driver then modulates the desired LED with such signals. The microcontroller needs a Digital-to-Analogue Converter (DAC) to generate analogue signals. Otherwise, the microcontroller only provides digital signals to the driver, which might be sufficient for some applications.

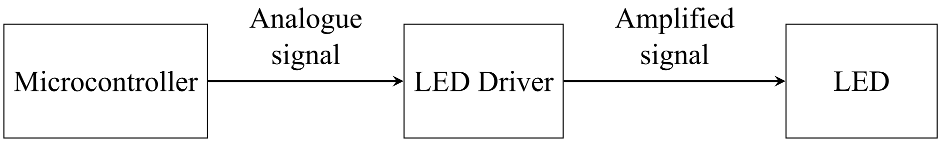


Figure 2.1: Block diagram of a generic Visible Light Communication Transmitter.

The previous research [8] proposed a VLC navigation node based on *NUCLEO-F446RE*. The node also encapsulated a LoRaWAN communication board and a LED driver. However, the designed system suffered from a low slew rate. Moreover, the LED driver produced significant heat and required sizeable passive heat sinks to prevent damage. The conclusion of the referenced thesis indicated these facts.

Long-term use of the device revealed a stability problem. The issue arose by reprogramming the *NUCLEO* device. The stability problem mainly concerned the Voltage-to-Current converter depicted in Fig. 2.2 adopted from [8]. The reprogramming of the *NUCLEO* board changes voltage levels randomly at active GPIO pins. Such rapid changes caused oscillations of the Voltage-to-Current converter based on a feedback loop.

I-NUCLEO-LRWAN1, LoRaWAN communication extension board, showed undesirable flash memory behaviour. The VLC node application requires low-latency communication with uplink and downlink communication. Alternating Current to Direct Current (AC/DC) power source powers the device. Therefore, energy consumption is not an essential parameter. Device class C, defined by LoRa communication protocol, possess these attributes. [12]. AT commands (collection of preloaded instructions) control the *I-NUCLEO-LRWAN1*. [13] However, the preloaded instructions do not allow the implementation of Class C at this time. Hence, to use this device efficiently, new firmware with a unique set of AT commands would have to be implemented. The flash memory of the board strongly limits its capabilities. The volume of the memory is 64 Kbytes. The memory volume prevents the uploading of programs developed for a different microcontroller and makes firmware development difficult. Suda, M. [14] describes the issues in detail.

This chapter further presents adjustments to the VLC node proposed in [8], solving the issues described above. A set of critical measurements and simulations support the applied techniques.

2.1 Microcontroller and LoRa communication

The previous research proposes using two microcontroller devices. The first device, *NUCLEO-F446RE*, worked as a waveform generator based on the downlink LoRa communication. *I-NUCLEO-LRWAN1* communicated with the LoRaWAN gateway and transferred the decoded messages to the waveform generator. *B-L072Z-LRWAN1* [15] replaces the two development boards for the reasons described above the system's unpractical shape. [8]

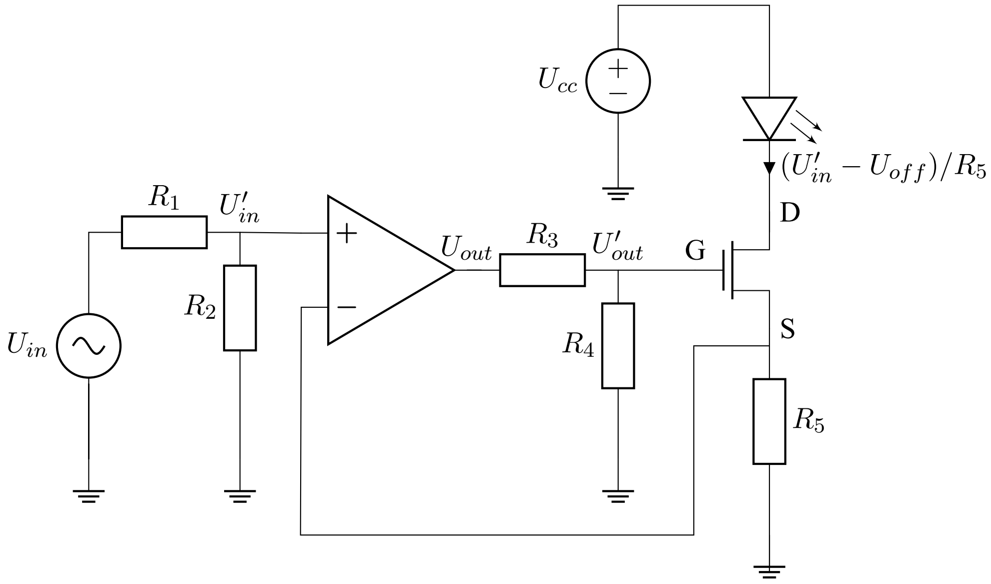


Figure 2.2: Schematic of the Voltage-to-Current converter. Adopted from [8].

Table 2.1 summarizes the critical features of the *NUCLEO* board. Firstly, the board needs to tolerate a 12 V power supply (see Section 2.3). Components such as *ADG819* [16] require an external power supply (the details are in [8]) from the microcontroller. The system communicates using the LoRa and LoRaWAN protocols. Therefore, the *NUCLEO* boards must support all LoRa modulations and be LoRaWAN class C compliant (see [14]). Lastly, for efficient VLC communication, the board needs timers capable of generating Pulse Width Modulation (PWM) signals and DAC for different waveforms. Table 2.1 shows that *B-L072Z-LRWAN1* satisfies all conditions of the system.

	B-L072Z-LRWAN1
Power supply	7-12 V DC power supply
External Sources	5 V external supply
LoRa	FSK, GFSK, MSK, GMSK and OOK modulations
LoRaWAN	class A and C compliant
Timers	6-bit (General purpose, Basic, Low Power)
DAC	2, GPIO PA4 (12-bit with 1 channel each)

Table 2.1: Essential *B-L072Z-LRWAN1* evaluation board features. [15]

2.2 Voltage-to-Current converter

Figure 2.3 depicts the newly proposed modification of the LED driver. The main difference between the new driver and the driver from Figure 2.2 stands in components C_1 and R_5. These components form a compensation network, which secures the stability of the circuit and dampens oscillations. However, the compensation network behaves similarly to a Low-Pass filter (LP) and limits the system's frequency response.

Schematic 2.3 also shows red and brown nodes and the standard black junction nodes. The colourful nodes indicate the measurement points. Section 2.2.4 explains the measurement techniques in detail.

Moreover, the proposed Voltage-to-Current converter introduces changes to multiple components. The previous research described the drawbacks of the applied Operational Amplifier (OPA) *UA741CD* [17]. Furthermore, sections of the research focus on heat issues of the system, especially of the N-type Metal-Oxide-Semiconductor (NMOS) transistor *IRF710* [18].

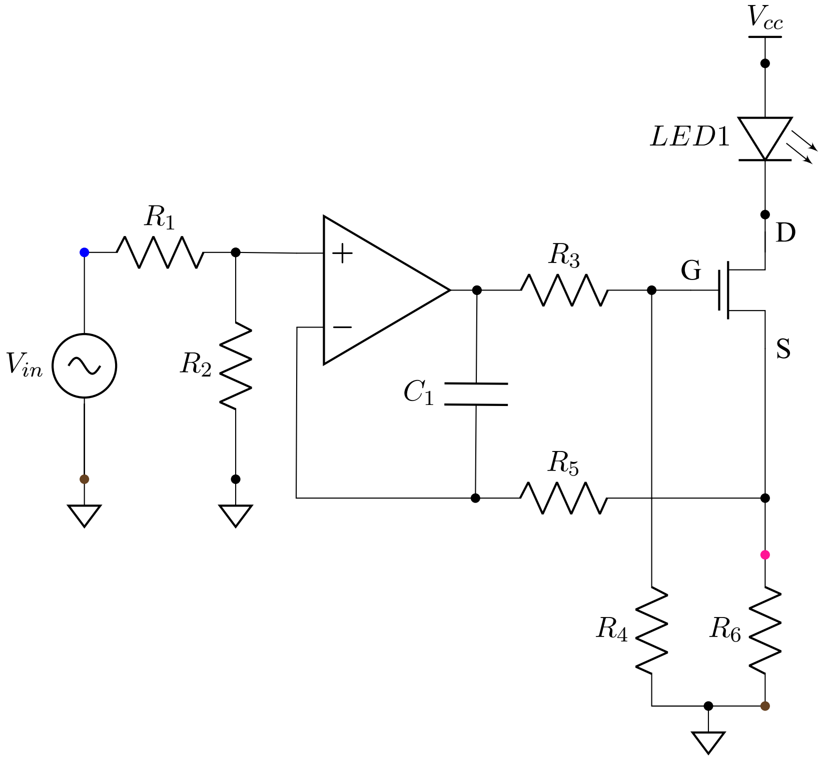


Figure 2.3: Schematic of the Voltage-to-Current converter.

2.2.1 Variations of electrical components

The schematic of the LED driver 2.3 shows three vital semiconductor components. The first one is the OPA connected in a non-inverting topology. The second essential element is the NMOS transistor, and the last crucial component is the LED. However, the LED requires no modifications, and therefore the paper suggests using the same component.

AD817 [19] replaces *UA741CD* [17]. The different OPA brings many improvements to the design. Table 2.2 summarizes all essential parameters of the operational amplifiers. The paramount factor is the slew rate of the OPA. *AD817* increases the slew rate from 0.5 V/μs to 350 V/μs. Therefore, the new OPA satisfies a slew rate condition, described in the bachelor's thesis, for 9.3 V/μs [8]. Moreover, the Common-Mode Rejection Ratio (CMMR) of *AD817* improves from 90 dB to 120 dB, and input offset voltage decreases to 0.5 V, making the system more precise and robust.

Parameter	Conditions [AD817]	AD817	Conditions [UA741CD]	UA741CD
Unity Gain Bandwidth	-	50 MHz	-	-
Slew Rate	Gain = 1, $R_L = 1$ kΩ	350 V/μs	$V_I = 10$ V, $R_L = 2$ kΩ, $C_L = 100$ pF, Gain = 1	0.5 V/μs
Input offset voltage	-40 °C to +85 °C	0.5 mV	$V_{out} = 0$ V	1 mV
Input offset current	-40 °C to +85 °C	25 nA	$V_{out} = 0$ V	20 nA
CMMR	$V_{CM} = \pm 12$ V	120 dB	$V_{CM} = \pm 12$ V	90 dB
Supply voltage	-	±18 V	-	±18 V

Table 2.2: Typical values comparison of *AD817* [19] and *UA741CD* [17]. The values are true for ambient temperature $T_a = 25$ °C and the supply voltage $V_{cc} = \pm 15$ V unless stated otherwise.

Transistor *IRF540NPbF* [20] replaces transistor *IRF710* [18]. A significant part of the previous research discussed the solution to the thermal issues of the power NMOS transistor [8]. However, the paper does not discuss reducing the Static Drain-to-Source On-Resistance $R_{DS_{(on)}}$ [Ω]. Switching the transistors decreases the $R_{DS_{(on)}}$ to 44 mΩ. Equation 2.1 obtains power loss P_d [W] as

$$P_d = \max\{Id\}^2 \cdot R_{DS_{(on)}}, \tag{2.1}$$

where the $\max\{Id\}$ [A] stands for the maximal drain-source current. The power loss equals 99 mW for maximum drain current 1.5 A and $R_{DS_{(on)}} = 44$ mΩ, which is a significant enhancement from the 8.1 W of the previous research.

However, swapping the transistors reduces the system's response speed. Table 2.3 indicates that the overall dynamic response times of transistor *IRF540NPbF* increase compared to the *IRF710*. Although the component's dynamic parameters worsened, the system does not suffer from low response speed.

Parameters	Conditions [IRF540]	IRF540NPbF	Conditions [IRF710]	IRF710
$R_{DS_{(on)}}$	$V_{GS} = 10$ V, $I_D = 16$ A	44 mΩ	$V_{GS} = 10$ V, $I_D = 1.2$ A	3.6 Ω
$V_{GS_{(th)}}$	$V_{DS} = V_{GS}$, $I_D = 250$ μA	4.0 V	$V_{DS} = V_{GS}$, $I_D = 250$ μA	4.0 V
Rise Time	$V_{DD} = 50$ V,	35 ns	$V_{DD} = 200$ V,	9.9 ns
Fall time	$I_D = 16$ A,	35 ns	$I_D = 2.0$ A,	11 ns
Turn-off delay time	$R_G = 5.1$ Ω,	39 ns	$R_g = 24$ Ω,	21 ns
Turn-on delay time	$V_{GS} = 10$ V	11 ns	$R_D = 95$ Ω	8 ns

Table 2.3: Typical values comparison of *IRF540NPbF* [20] and *IRF710* [18]. The parameters are valid for junction temperature $T_j = 25$ °C unless noted otherwise.

2.2.2 Stability of the Voltage-to-Current converter

The components C_1 and R_5 form the compensation network. The compensation secures the correct behaviour and prevents the NMOS transistor from *Ringing*. *Ringing* is an oscillation effect caused by the transistor's parasitic capacitance and is typical for power MOSFET devices. [21]

Open-loop gain and phase are crucial factors for finding the stability of a feedback loop system. The feedback loop must be broken and replaced with an excitation source to transfer the circuit from a closed-loop circuit to an open-loop circuit. Fig. 2.4 depicts an open-loop simulation circuit of the Voltage-to-Current converter.

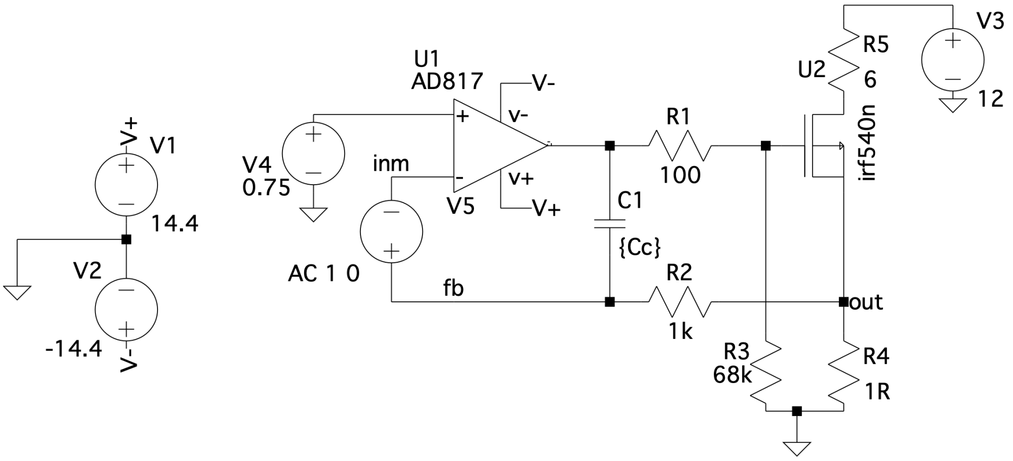


Figure 2.4: Open-loop stability simulation circuit of the Voltage-to-Current converter. Parameter C_c signals variable capacitance value of capacitor C_1.

In schematic 2.4, the voltage source V_5 breaks the feedback loop and excites the system with a Dirac delta function. Voltage source V_4 provides the actual bias voltage to the circuit. The bias is 0.75 V due to the output current range being 0 A to 1.5 A. Company Analog Devices provides the know-how for designing an open-loop simulation circuit and simulation techniques in an application note [22].

Resistor R_5 replaces the LED. The resistance value is 6 Ω. The resistor simulates the typical behaviour of the LED that the previous research [8] measured. LED's datasheet [23] describes its parameters.

One of the techniques to evaluate the stability of the circuit is assessing the phase margin [°] of the system's frequency response. The ratio of voltages in nodes inm and fb (see Fig. 2.4) provides the appropriate transfer function. Computing the ratio $V(fb)/V(inm)$ results in a loop gain magnitude [dB] and phase [°]. The phase margin is a value of the phase plot at frequency, where the loop gain magnitude reaches 0 dB. [24] [22]

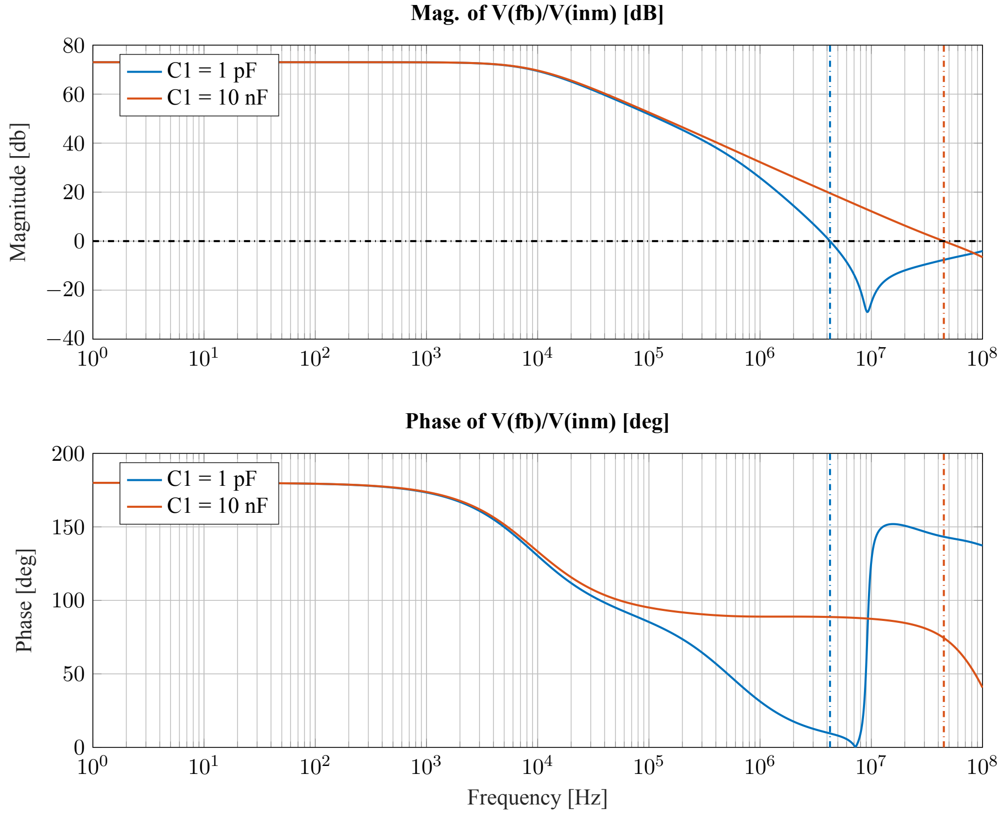


Figure 2.5: Open-loop stability simulation. The graphs show magnitude [dB] and phase [°] of $V(fb)/V(inm)$ ratio. The blue waveform indicates system behaviour for capacitance value C_1 = 1 pF. The orange curve offers the converter's behaviour for C_1 = 10 nF. The dotted-dashed lines mark the phase margin value. The colours of the marking lines correlate to the colours of the waveforms. Capacitance C_1 = 1 pF achieves phase margin ≈ 9°. C_1 = 10 nF returns phase margin ≈ 75°.

Figure 2.5 offers the stability simulation results. The first panel shows the loop gain magnitude, and the second depicts the phase plot. The charts display two waveforms. The blue curve represents the circuit's behaviour with a capacitor C_1 = 1 pF. The orange line exemplifies the response with a C_1 = 10 nF. The vertical dashed lines mark the phase margins for both functions. The colours of the lines correlate to the functions. The simulations deliver that adding the capacitor and resistor to the feedback loop flattens the loop gain magnitude and causes the phase to roll off later. Changing the capacitance values increased the phase margin from ≈ 9° to ≈ 75°, above the recommended threshold of 45°. [24]

2.2.3 Closed-loop simulation of Voltage-to-Current converter

The final simulation of the Voltage-to-Current converter is the closed-loop model simulation. The compensation network with resistor R_1 forms the feedback loop and necessary attenuation in this scenario . The simulation schematic is available in Fig. 2.6.

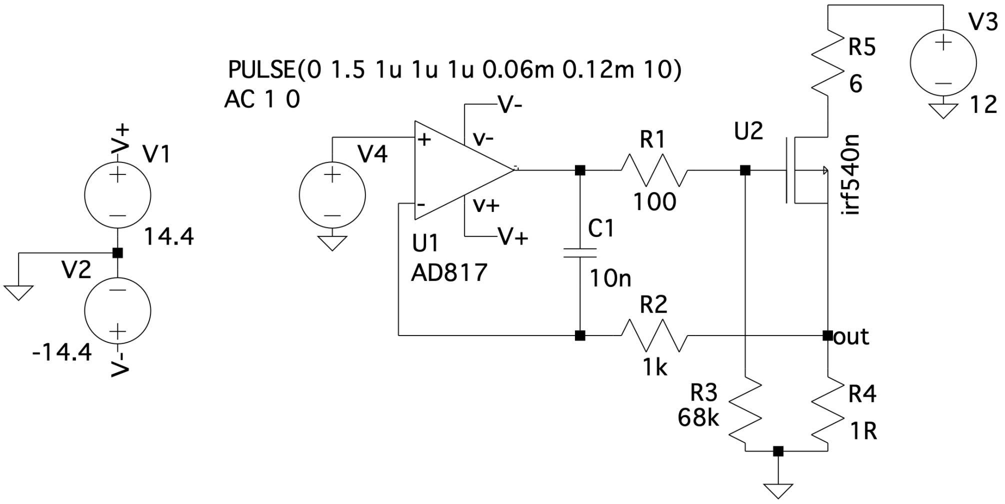


Figure 2.6: Closed-loop simulation circuit of the Voltage-to-Current converter. The simulation circuit uses two excitation sources (V_4): an Alternating Current source and approx. 8.5 kHz Pulse Width Modulation source.

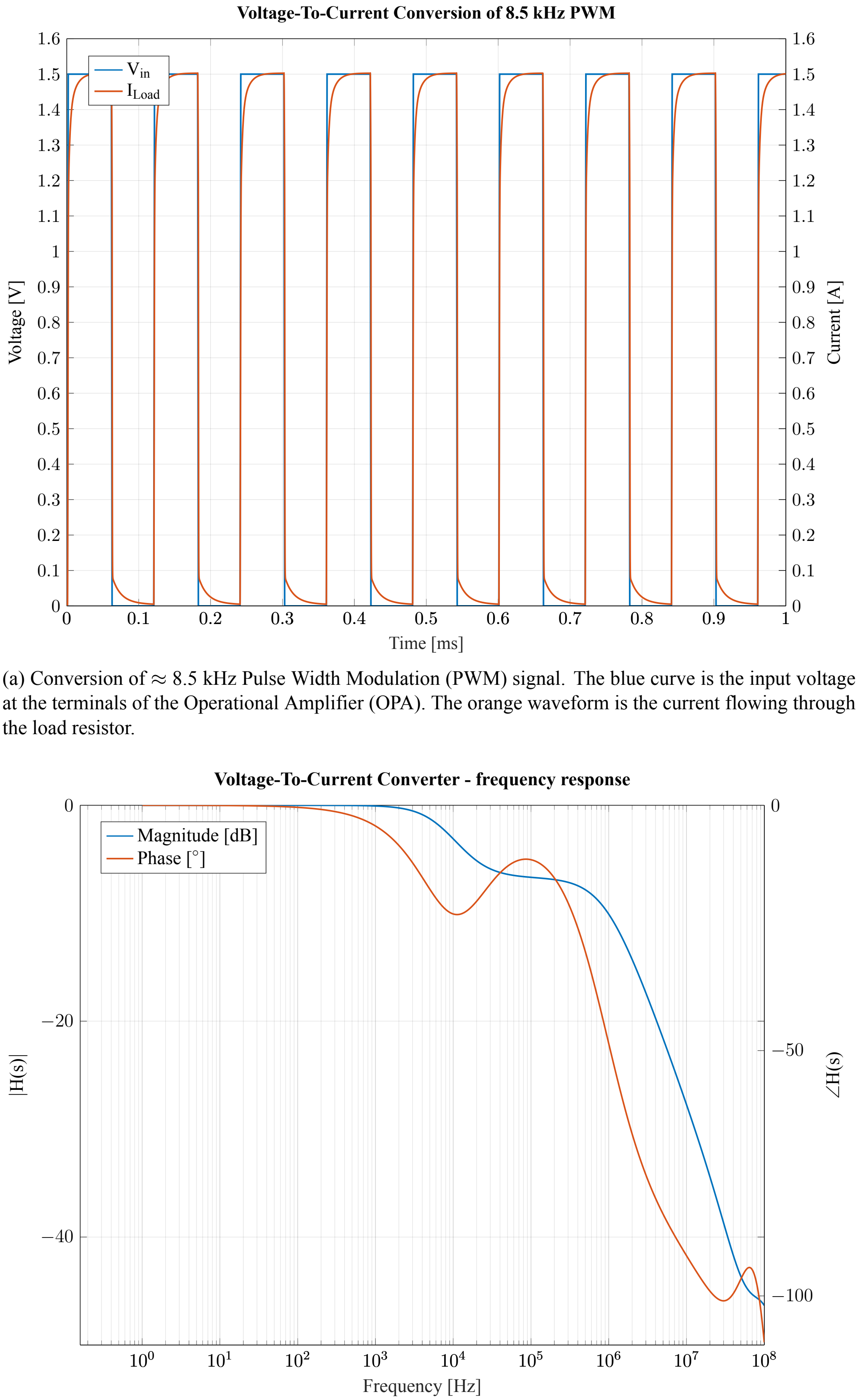
Figure 2.6 shows the excitation of the system. The voltage source V_4 carries SPICE directives for time-domain simulation and frequency-domain simulation. Approximately 8.5 kHz PWM function with ten cycles and a rise/fall time of 1.5 μs excites the computer simulation. Fig. 2.7 are the calculation results.

Figure, 2.7a, offers the time-domain assessment. The blue curve is the input voltage at the terminals of the OPA. The orange waveform is the current flowing through the load resistor. The simulation uses nominal temperature T_{nom} = 27 °C.

The waveforms illustrate that the current curve deviates at rising and falling edges from the input voltage signal. The frequency of the PWM is above sufficient for the algorithm testing platform. Moreover, studies using CMOS cameras suggest that frequencies above 7 kHz might be inefficient, and the LED recognition rate decreases [25].

The frequency-domain voltage source excites the system with a Dirac function of magnitude 1 V.

Lastly, Figure 2.7b delivers the frequency response of the system. The blue waveform is the magnitude of the transfer function. The orange curve stands for the phase of the transfer function. The graph indicates that the system suppresses frequencies higher than approximately 3 kHz. The attenuation at 10 kHz is 3 dB. It also shows that the trend of the phase curve is highly non-linear. The phase curve deviates from the null phase shift at approximately 200 Hz. However, the maximum phase deviation from 1 Hz to 10 kHz is approximately 22°.



(a) Conversion of ≈ 8.5 kHz Pulse Width Modulation (PWM) signal. The blue curve is the input voltage at the terminals of the Operational Amplifier (OPA). The orange waveform is the current flowing through the load resistor.

(b) The transfer function of the Voltage-to-Current converter. Phase deviation in a band from 1 Hz to 10 kHz is 22°. The magnitude at 10 kHz reaches an attenuation of 3 dB.

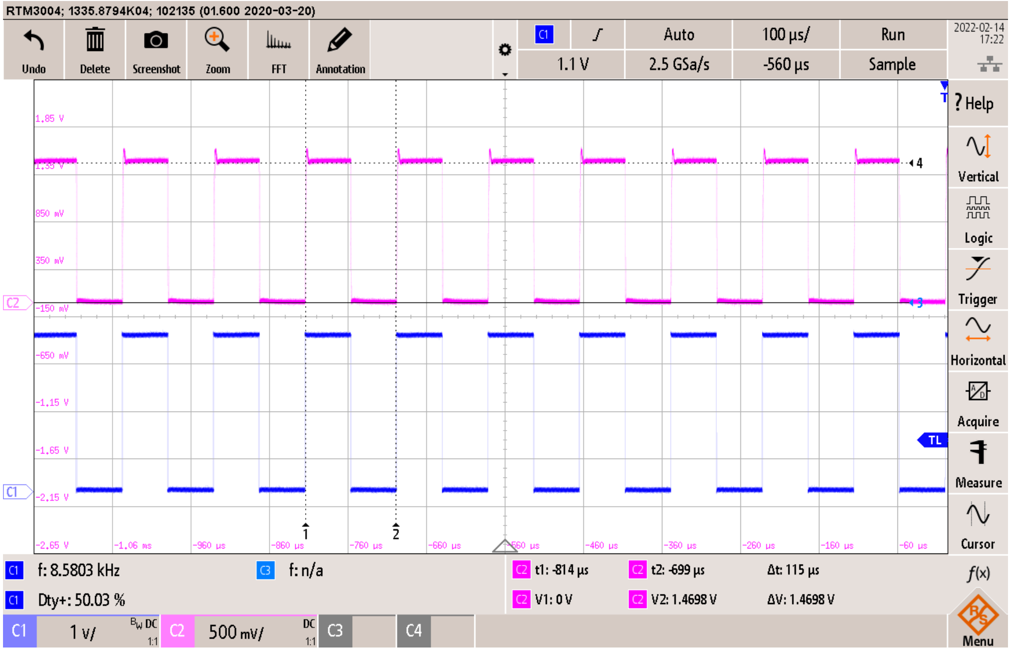
Figure 2.7: Closed-loop transient and transfer simulations. The nominal temperature T_{nom} is 27 °C.

2.2.4 Measurement of the Voltage-to-Current converter

Measurements of the electronics are a critical part of the development process. Measurements are the best comparison between simulated and practical results.

A time-domain signal evaluation is the most straightforward assessment of the Voltage-to-Current converter. Specifically, the presented measurement used the *Rohde & Schwarz RTM3004 + RTM-B2410* oscilloscope [26]. Fig. 2.3 shows the analysis points. The blue and bright pink dots mark the positions of the oscilloscope probes. The brown dots are the grounding points.

Fig. 2.8 offers a measurement result of a PWM conversion. The blue curve represents the output signal of the microcontroller *B-L072Z-LRWAN1*. The bright pink signal is the voltage signal of the grounded resistor R_6, where R_6 = 1 Ω. Therefore, the voltage directly correlates to the current signal driving the load.



RTM3004; 1335.8794K04; 102135 (01.600 2020-03-20)

C1 1.1 V Auto 2.5 GSa/s 100 µs/ -560 µs Run Sample

2022-02-14 17:22

C1 f: 8.5803 kHz

C1 Dty+: 50.03 %

C3 f: n/a

C2 t1: -814 µs

C2 t2: -699 µs

Δt: 115 µs

C2 V1: 0 V

C2 V2: 1.4698 V

ΔV: 1.4698 V

C1 1 V/ BW DC 1:1

C2 500 mV/ DC 1:1

C3

C4

Figure 2.8: Measured Pulse Width Modulation signals. The blue signal is the output signal of the *B-L072Z-LRWAN1* microcontroller. The bright pink signal delineates the converted signal. The assessed signal is a voltage signal measured on a 1 Ω grounded resistor R_6 (see Fig. 2.3). Thus, the voltage signal is equivalent to the load's current signal.

The comparison of the simulation results in Fig. 2.7a and the practical results in Fig. 2.8 accentuates the differences. Firstly, the practically acquired data do not possess slow transitions from the rising and falling edges. Contradictorily, the measured voltage signal shows an overshoot in the rising edge transitions. Moreover, the assessed data of the converted signal show a slight amplitude drop compared to the simulation. The amplitude behaviour is most likely caused either by the resistivity tolerance of the applied resistors or by the compensation network. The newly selected transistor causes both the slow transitions in the simulation and the overshoots in the measurement. Transistor's *IRF540NPbF* [20] slower dynamic response introduces the undesirable transitions. The inaccuracies of the simulation models consequence in the difference between the signal's nature in measurement and the simulation. The horizontal cursors point out the peak-to-peak amplitude. The vertical cursors read the signal period.

However, the imperfections of the conversion are slight deviations from the ideal state. Such anomalies present minor issues to the system and are negligible.

2.3 Power supply

The original design suggests using a 24 V power supply with a maximum output current of 2.5 A. The proposed AC/ DC converter provides power for the entire system, including the LED driver, microcontroller and a LoRa extension board. [8]

Two separate sources power the OPA. The first source is a positive voltage regulator, providing electrical power to the *NUCLEO* board. The regulator adjusts the 24 V AC/ DC adapter to 12 V. However; this system is highly vulnerable to overheating due to the significant electrical power flowing through the regulator. Using the well-known formula obtains dissipated power as

$$P_d = V \cdot I, \tag{2.2}$$

where V [V] stands for the voltage drop across the device and I [A] is a current flowing through the device. Considering the maximum current draw of the *NUCLEO* board, 250 mA, and the voltage drop of 24–12 $=$ 12 V, the total power dissipation is 3 W. The *LM340T-12/NOPB*'s datasheet does not provide data regarding the internal power dissipation. However, neglecting this parameter is valid because the current into the load is considerably higher than the internal currents. [27]

Table 2.4 summarizes the essential parameters of the voltage regulator.

LM340T-12/NOPB

Dissipated power [W]	Package	Ambient temperature [°C]	Junction-to-ambient thermal resistance [°C/W]
3	TO-220	25	23.9

Table 2.4: *LM340T-12/NOPB*'s relevant features. Data acquired from the component's datasheet [28].

Equation from [27] determines the junction temperature T_j as

$$T_j = T_A + \theta_{JA} \cdot P, \tag{2.3}$$

where T_A [°C] is the ambient temperature, θ_{JA} [°C/W] is the junction-to-ambient thermal resistance, and lastly, P [W] is the component's power dissipation. For the parameters listed in Tab. 2.4, the junction temperature is 96.7 °C. This temperature is within the applicable range of 0 °C $\leq T_j \leq$ 125 °C. However, if the ambient temperature rises, the system might overheat and be damaged.

The second power source is a high current charge pump Direct Current to Direct Current (DC/DC) converter *TC962EPA*, inverting the positive voltage of 12 V. This sub-system presents no temperature issues. However, the components are spatially exhausting due to the Through-Hole Technology (THT) montage.

Moreover, this design provides no anti-measures against possible ripple in the Direct Current (DC) voltage and Electrostatic Discharge (ESD) protection circuits. [8]

2.3.1 The Alternating Current to Direct Current adapter

Changes in the Voltage-to-Current converter explained in section 2.2 enabled a change in power supply and lowered the system's overall power consumption. From the previous 24 V, the power voltage lowers to 12 V. The lower voltage allows direct powering of the microcontroller and removal of the voltage regulator.

RPS-30-12 replaces the previously used power supply *RAC60-24S/OF*. It is an AC/DC converter with variable input from 80 VAC to 264 VAC and an output of 12 V for frequencies between 47 Hz to 63 Hz. It can provide up to 2.5 A output current. Additional information about the component is under [29].

2.3.2 The OPA's power supply

Redesigning the power supply was necessary to solve ESD and robustness related issues. The direct connection between the AC/DC adapter and the OPA presents a potential problem due to the adapter's

100 mV peak-to-peak ripple voltage [29]. Furthermore, a short connection between the elements can lead to possible destruction of the OPA in a case of a voltage surge. The proposed solution is in Fig. 2.9. The circuit consists of two main parts: DC/DC converter and a voltage smoothing circuit with ESD protection.

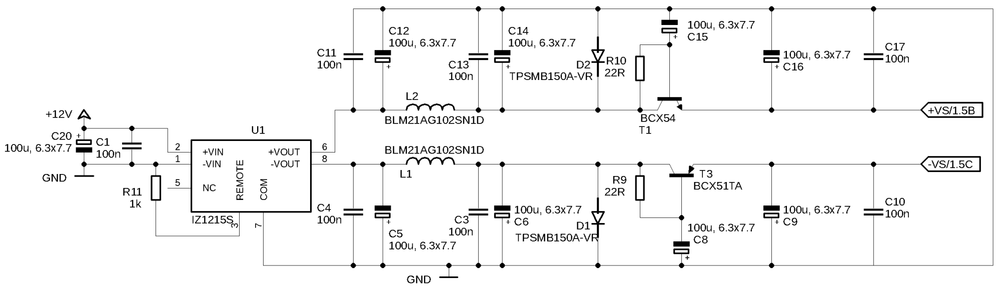


Figure 2.9: Power supply of the Operational Amplifier.

IZ1215S serves as the DC/DC converter. It is an isolated module that adjusts input voltage of 9 V to 18 V to ±15 V, and it provides up to ±100 mA output current. However, it might produce a 75 mV peak-to-peak ripple voltage at the output. This component provides the necessary means to power the OPA symmetrically [30]. The input capacitors ensure the stability of the converter and filter noise at the input terminals by shorting it directly to the ground (pg. 8, [31]).

Resistor $R11$ connects the negative input voltage (the ground) to the **REMOTE** pin of the Integrated Circuit (IC). This grounding resistor ensures that the converter is always in the *On* state [30]. The resistor prevents damage to the IC from ESD or unexpected shorts.

The output circuits serve as filtration of the output voltage of the converter. The branches combine ceramic capacitors, electrolytic capacitors, ferrite beads, ESD protection diodes and capacitance multipliers.

The first output capacitors ($C11$ and $C12$ for the positive output and $C4$ and $C5$ for the negative output) fulfil the same filtering function as the input capacitors.

Components $L1$ and $L2$ are the ferrite beads. These components act as a series LR circuit. Connecting this component into a series with a capacitor creates an LP filter similar to an RC filter. The selected component is the *BLM21AG102SN1D*. [32]

Reeve B. in [33] explains the function of a capacitance multiplier. The capacitance multiplier combines a bipolar transistor, resistor and capacitor, where the bipolar transistor acts as an emitter follower. Fig. 2.10 shows a capacitance multiplier connected as a simple RC filter.

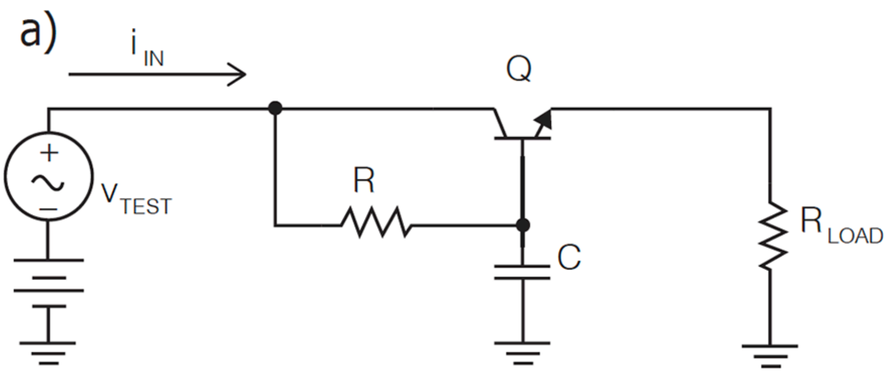


Figure 2.10: Capacitance multiplier used as RC filter. Adopted from [33].

The following equation determines the transfer function as

$$\frac{Vout}{Vin} \cong \frac{1}{RC} \cdot \frac{1}{s + \frac{1}{(R \| \beta R_L)C}}. \tag{2.4}$$

R [Ω] is the resistance value of the resistor R, C [F] is the capacitance value of capacitor C. s is the Laplace operator, and R_L [Ω] stands for the resistance of the load. Lastly, β [-] represents the current gain of the transistor. Equation 2.4 shows that the transistor enhances the pole of the transfer function. Improving the pole results in enhanced ripple rejection and lower low-frequency insertion loss, resulting in lower power dissipation and lower drop out voltage. Moreover, the transfer function shows that the capacitance C is effectively multiplied by β, allowing the selection of a small-scale capacitor, with a lesser capacitance value. [33].

However, the transistor requires a correct bias voltage to work correctly. Therefore, the output voltage is less than the base voltage of the transistor by V_{BE}, which is the base-emitter voltage. V_{BE} is generally around 0.65 V. [34]

The designed capacitance multipliers use transistors *BCX54-16*, a typical NPN bipolar transistor [35] and *BCX51TA*, a standard PNP bipolar transistor [36].

The remaining capacitors ($C16$ and $C17$ for the positive output, $C9$ and $C10$ for the negative output) boost the filtering effect and stabilize the output voltage.

Diodes $D1$ and $D2$ are ESD protective diodes designed for lightning or other transient effects. Diode *TPSMB150A-VR* provides fundamental ESD protection against high voltage surges. [37]

2.3.3 Simulation of the Operational Amplifier's power supply

Lastly, this subsection delivers simulations of the OPA power block. Fig. 2.11 depicts the time-domain simulation schematic. The process uses SPICE models provided by the manufacturers, and all models correlate to the circuit schematic shown in Fig. 2.9. Resistor $RL = 4.6$ MΩ replaces the load. The value is a rough estimation of the impedance value for 15 V supply voltage and 3.3 μA current, a typical input bias current of the *AD817* OPA [19]. The solver uses a nominal temperature of 27 °C.

The time-domain analysis uses two separate voltage sources to create a noisy switch function. The two voltage source technique simulates the initial state of the device. Sinusoidal function with amplitude of 0.2 V and 60 Hz frequency composes the noise. The pulse is a Heaviside like function with an initial voltage equal to 0 V, 15 V amplitude, a rise time of 1 μs, and a delay of 1.5 s. Fig. 2.11 shows the voltage sources and their SPICE directives. The period of the transient analysis is 0-600 ms.

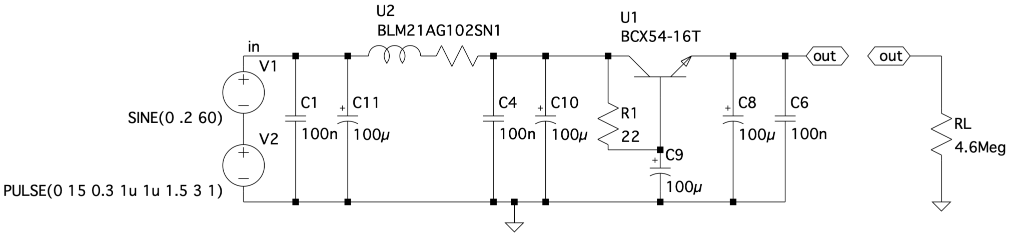


Figure 2.11: LTspice time-domain simulation circuit of the Operational Amplifier's supply block.

Fig. 2.12 illustrates the results. The blue curve is the input voltage signal, and the orange line is the output voltage on the resistor RL. The results show that the filtering circuit eliminates the noise. Furthermore, the output voltage settles approximately 0.5 V below the 15 V amplitude of the pulse. A DC voltage measurement reads that the actual output voltage is 14.41 V.

The frequency-domain analysis uses a setup from Fig. 2.13. The difference between the time-domain analysis stands in the circuit's excitation. The frequency circuit is excited by an approximation of the Dirac delta function. The simulation runs in 1 Hz to 1 GHz intervals and calculates 100 points per octave.

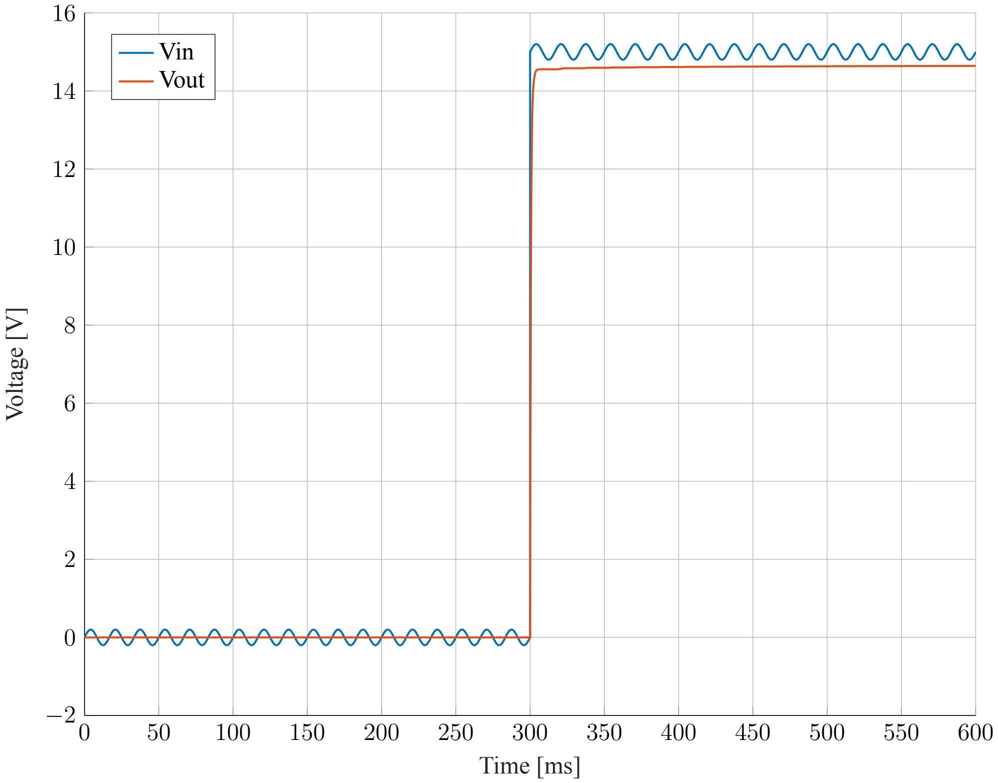


Figure 2.12: Transient simulation of the Operational Amplifier's power supply circuit.

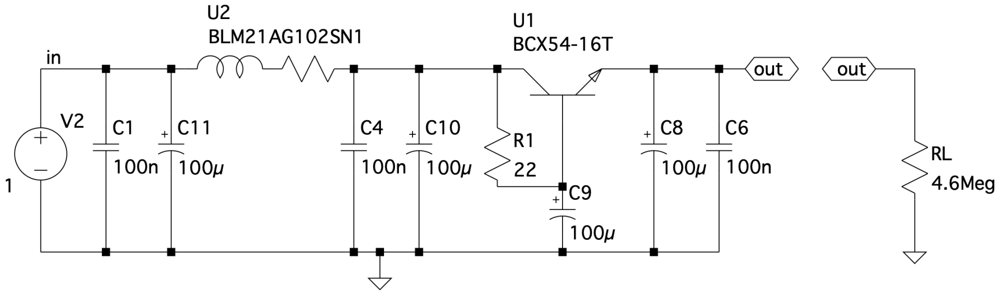


Figure 2.13: LTspice frequency-domain simulation circuit of the Operational Amplifier's supply block.

Fig. 2.14 represents the result of the simulation. The analysis shows that the filtering circuits strongly suppress frequencies higher than 0 Hz. The graph shows a magnitude [dB] of the output voltage on resistor RL.

2.4 The thermal management

Subsection 2.2.1 explains modifications of the components of the Voltage-to-Current converter. Replacing the NMOS transistor decreased the dissipated power in the transistor and reduced the component's dissipated heat. The previous research discussed the effects of the components' power flow and dealt with the thermal issues. [8]

Previously, the research established that the heat sink's thermal resistance is equal to

$$R_{th} = \frac{T_{max} - T_a}{P_d} - R_{CS} - R_{JC}, \tag{2.5}$$

where T_{max} [°C] is the maximally allowed temperature of the device and T_a [°C] is the ambient

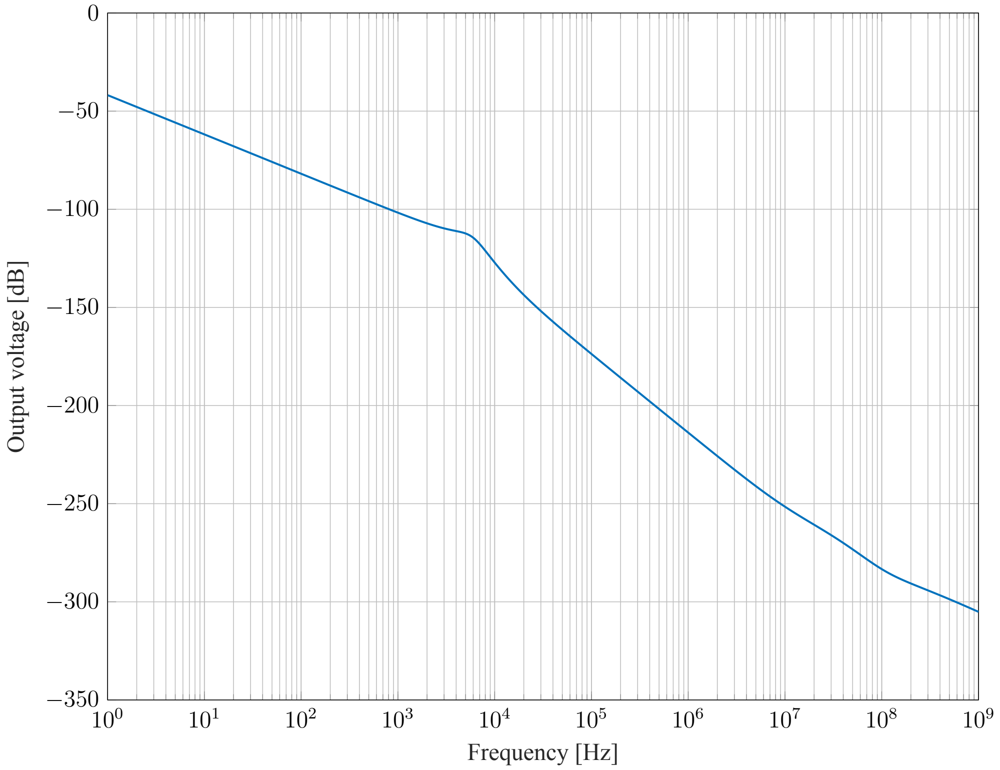


Figure 2.14: Transfer function of the Operational Amplifier's power sub-system.

temperature. P_d [W] stands for the dissipated power in the device, and lastly, R_{CS} [°C/W] quantifies the case-to-sink thermal resistance, and R_{JC} [°C/W] specifies the junction-to-case thermal resistance. [8]

Table 2.5 compiles all necessary data about transistor *IRF540NPbF* to estimate the requisite heat sink's thermal resistance. Using Equation 2.5 and data from Tab. 2.5, the R_{th} is equal to

$$R_{th} = \frac{75 - 25}{44 \times 10^{-3}} - 0.5 - 1.15 \approx 1135\text{°C/W}.$$

The result shows that the transistor does not require a heat sink since the junction-to-ambient thermal resistance is 62 °C/W. The LED installation uses the same calculation approach indicated in the previous research [8].

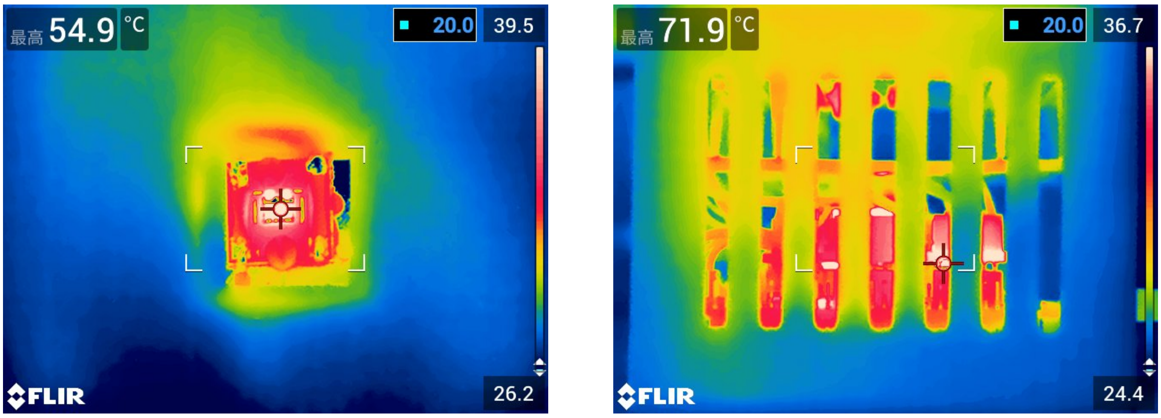
IRF540NPbF

T_{max}	T_a	P_d	R_{CS}	R_{JC}
75 °C	25 °C	44 mW	0.5 °C/W	1.15 °C/W

Table 2.5: Thermal features of transistor *IRF540NPbF*. P_d is accurate for current equal to 1.5 A. T_{max} is the required maximum temperature of the device. T_a is an estimate of the ambient temperature.

Lastly, Fig. 2.15 shows an evaluation of the system's temperature. The LED's temperature reaches approximately 55 °C. The side view of the system reveals that the extension board with the Voltage-to-Current converter is the system's most searing part. The heat source targeted by the thermal camera *FLIR E75* [38] is the 1 Ω resistor of the Voltage-to-Current converter. The resistor's power dissipation is $P_d = 1.5^2 \cdot 1 = 2.25$ W, which makes the resistor the second most electrically stressed component. However, a temperature of 72 °C is still below the temperature threshold of 75 °C.

Fig. 2.15 show the system's protective case. The case also integrates a holder for active cooling. Fan *DC0401012V2B-2T0* [39] works as a safety measure against the system overheating. The 12 V AC/DC supply powers the fan and the rest of the system.



(a) The temperature measurement of the Light-Emitting Diode.

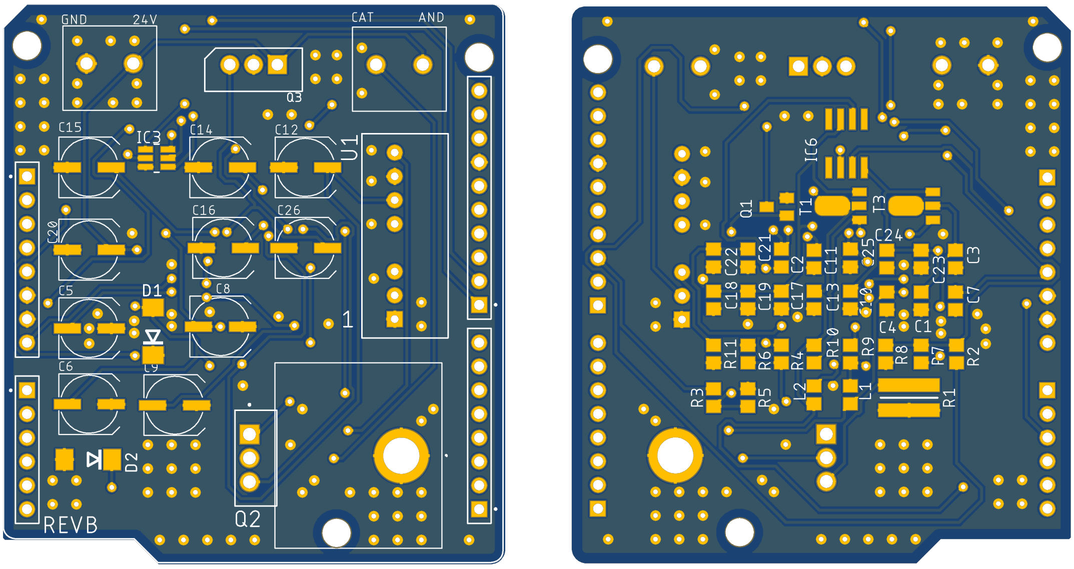
(b) The temperature measurement of the node installation —side view.

Figure 2.15: The thermal assessments of the Light-Emitting Diode and the node installation. Images come from a *FLIR E75* thermal camera. [38]

2.5 Realization

The previous chapters detail the VLC node's hardware. The last remaining part is the system's realization.

A Printed Circuit Board (PCB) accommodates the Voltage-to-Current converter, the power supply's sub-systems and the signal multiplexing (see [8]). The PCB forms an extension board that allows connecting the LED, the AC/DC power supply, and the NMOS transistor. The NMOS transistor shares the same heat sink with the LED. Fig. 2.16 depict the front and back view of the PCB board.



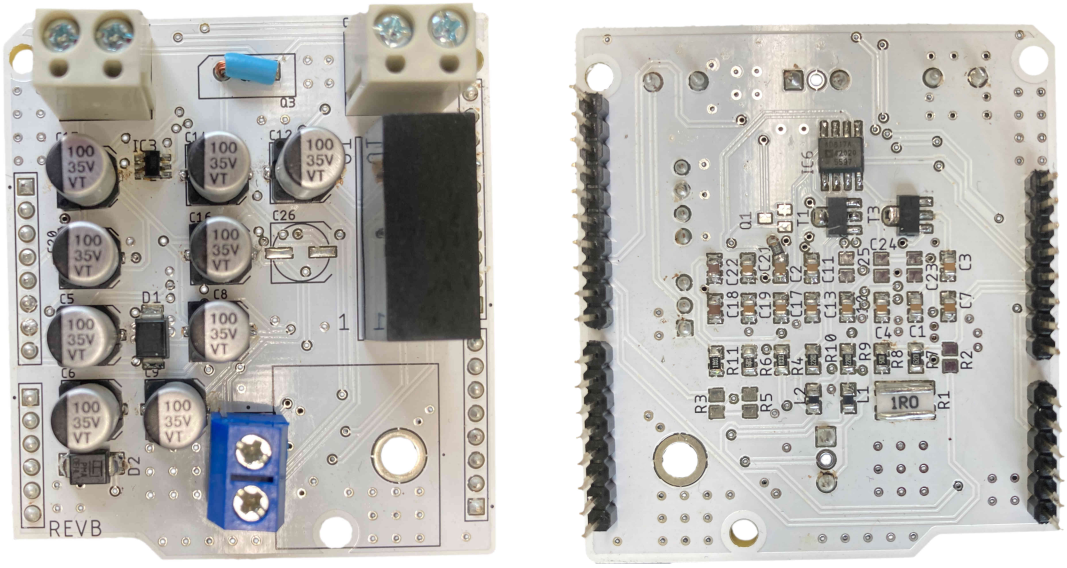
(a) The front view of the PCB.

(b) The back view of the PCB.

Figure 2.16: The extension board realization for a Visible Light Positioning node-exported CAD views.

The PCB underwent some modifications. Firstly, the initial supply voltage is 24 V. The PCB accounted for the voltage regulator supplying the *NUCLEO* board (component $Q3$). Lastly, the board supports the connection of the NMOS transistor and its heat sink (component $Q2$). The PCB also implements an On/ Off branch for the Voltage-to-Current converter, which showed undesirable behaviour in testing.

Figures 2.17 show the PCB's front and back view with mounted components. The images depict replacing $Q3$ with a bridging wire, switching NMOS transistor $Q2$ with terminal blocks and lastly, the back view shows modifications to the PCB regarding capacitor $C2$. The backside's augmentations refer to changes in the On/Off branch and the Voltage-to-Current converter tuning.



(a) The front view of the PCB.

(b) The back view of the PCB.

Figure 2.17: The extension board of the Visible Light Positioning node-realization photos.

The power NMOS transistor shares the same heat sink as the LED. Figures B.1, B.2, and B.3 offer the final installation of the system and are a part of the Attachments B. The images depict the positioning of the sub-systems within a 3D printed custom case. Images with the LED and without the LED illustrate the node's composition. The last image of the triplet frames the transistor mounting onto the heat sink and its connection to the PCB board.

Chapter 3

Theory and the visible light receivers

The previous chapter discusses the technology of a transmitter station. However, every complete VLP system consists of a receiving station and software backend. The backend provides the positioning algorithms. Both hardware and software set apart the different VLC positioning systems. A system can be categorised using the taxonomy from a VLP survey [40]. The categorisation is a crucial step for understanding the possibilities and the limits of the system and selecting an excellent VLC receiver. Figure 3.1 illustrates the classification of VLP systems.

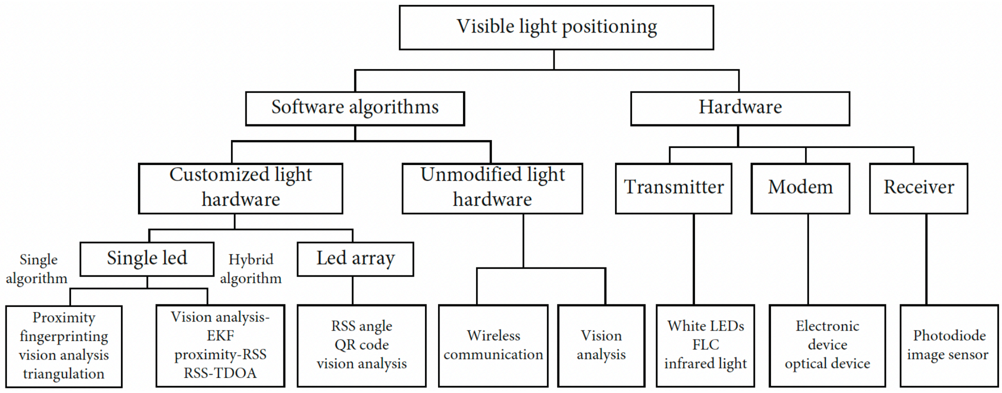

Visible light positioning
Software algorithms
Hardware
Customized light hardware
Unmodified light hardware
Transmitter
Modem
Receiver
Single algorithm
Single led
Hybrid algorithm
Led array
Proximity fingerprinting vision analysis triangulation
Vision analysis-EKF proximity-RSS RSS-TDOA
RSS angle QR code vision analysis
Wireless communication
Vision analysis
White LEDs FLC infrared light
Electronic device optical device
Photodiode image sensor

Figure 3.1: Taxonomy of VLP systems. Adopted from [40].

The proposed hardware is a custom setup using a specifically selected microcontroller device and electronic circuit for modulating the LED. Furthermore, the transmitter utilizes a single LED for illuminating the space and transmitting the required data. According to the taxonomy tree, the most utilized approaches are the single and the hybrid algorithms. The single algorithm refers to proximity, fingerprinting, or triangulation. The previous research discussed these methods in the theoretical introduction [8]. The hybrid algorithms combine the single algorithms to achieve higher accuracy and robustness, such as proximity-RSS, or RSS-Time Difference of Arrival (TDoA). [40]

Figure 3.2 depicts the general hardware structure of a VLP system. Choosing the proper hardware simplifies the positioning process, increases the system's robustness, and reduces the device's cost. The transmitting end consists of a modulator and a transmitting device. The previous chapter describes the transmitter in detail. The transmitter emits the modulated optical signal into the free space. The receiving end perceives these signals and demodulates the critical data from the signals. Lastly, the selected VLP algorithm retrieves the location information from the signal and the demodulated data. Three main blocks classify the entire structure: modem (modulator and demodulator, transmitter and receiver). [40]

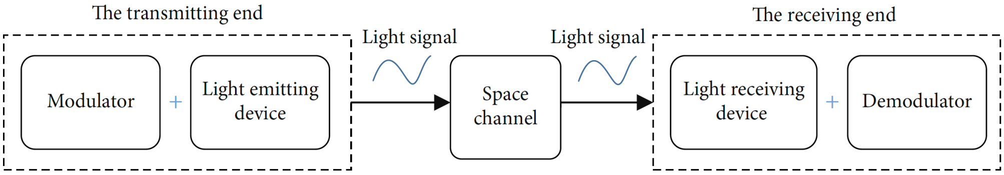


Figure 3.2: Hardware structure of a Visible Light Positioning system. Adopted from [40].

This chapter discusses the available positioning algorithms for the designed hardware and state-of-the-art studies that successfully implement the positioning systems. Furthermore, the chapter suggests diverse VLC receivers.

3.1 Software positioning algorithms

The previous part of the chapter mentioned the categorisation of the proposed device. Since the system uses a microcontroller and modulating circuit, it belongs to the custom hardware class. The unmodified lighting hardware refers to systems that do not modify the lighting hardware and thus inherit the system's properties. Such a setup reduces the system's deployment cost and increases its versatility. However, the shortcomings of such a setup are undeniable: the optical parameters are difficult to extract and analyse [40]

The difference between a single LED and an LED array is that with a single LED, each LED represents the positioning device. In contrast, the array is a device composed of multiple lighting devices, and each of them can serve as the positioning unit. The most common positioning principles of a single LED and customised hardware are proximity, fingerprinting, vision analysis, triangulation, and hybrid algorithms. [40]

3.1.1 Proximity

Researchers have widely explored proximity algorithms with communication technologies such as BT or RFID. These methods provide relative location information and are the most straightforward positioning techniques. The principle of the algorithm is very trivial. Usually, when the receiving station perceives the incoming signal, the transmitter is close to the receiver. Simply comparing the signal strength from different transmitters returns the information about the closest TX station. The transmitter's location dictates the positioning of the receiving end - the RX's position is approximately the position of the nearest transmitter. The target is between the transmitters if the RSS is the same for multiple signals. [7]

The simplicity of the approach makes this technique very attractive. Nakajima, Madoka and, Haruyama, Shinichiro focused their research [41] on helping the visually impaired. Their device uses a geomagnetic sensor and a VLC receiver to determine user's position. The system also guides the user via voice navigation and provides information about the movement direction. The system achieves an accuracy of 1 to 2 meters.

In [42], the authors focus on assessing the Field of View (FoV). The calculations utilize the 3D rotation matrix, a cone function, and Support Vector Machines (SVM). The FoV is an area around the receiver where the device can perceive incoming signals. The SVM is a supervised machine learning approach for object classification. The classifier determines the class of an item. The classifier is a model trained using training data and training algorithms [40]. The paper mathematically defines possible azimuth and tilt angulations. Finally, the results show that the approach reduced the computation redundancy during the Geometric Optics (GO) calculations by more than 80% in all evaluation cases.

Lastly, in [43], the researchers combine VLC and an ad-hoc network for localization. The research utilizes Zigbee mesh wireless network and produces two-hybrid positioning systems. The first design is the noncarrier VLC-based transceiver scheme, which showed better characteristics in long-range localization and low data rate communication. The second scheme applies a 4 MHz carrier to the VLC-based transceiver scheme. The second approach proved suitable for mid to short-range localization and high optical data rates. The main advantage of the 4 MHz carrier scheme is its robustness against noise and wide visible light reception.

3.1.2 Fingerprinting

Fingerprinting or sometimes called scene analysis, is an approach that compares measurement data related to the receiver's position with data measured online. The fingerprints (data measured in the first stage) may vary due to irregular base station distribution, obstacle uncertainty, and the receiver's location changes. However, the method relies on the differences since it estimates the position based on the variations in fingerprints. [40] [7]

There are two types of fingerprinting techniques. The first method is *radio-map-based* fingerprinting positioning. This approach executes two stages - the offline and online stages. The system measures the indoor environment's specific information in the offline stage, such as the signal's Angle of Arrival (AoA), RSS or Time of Arrival (ToA). A database collects this data and processes the necessary information. The gathered data forms the radio map. In the online stage, the system matches the data measured online with the data in the database and retrieves the relative location of the target. The second approach, called the *map-free* fingerprinting positioning, reduces the complexity of mapping maintenance. [40] [7]

A common fingerprinting technique is the RSS based positioning. The map consists of a grid, where each window contains a fingerprint location and the RSS value of the observed LED. The grid size is a hyperparameter and directly influences the maintenance cost of the map. Welch's method assesses the Power Spectral Density (PSD) and returns the information about the signal's amplitude. A filtering method can separate and distinguish each LED lamp if every LED emits a modulated signal with a different carrier frequency. A VLC receiver records RSS from each LED lamp in the online stage. A metric, typically the Euler distance, evaluates the difference between the online measured signal strength and the RSS contained in the radio map. The Euler distance for fingerprints $i = 1, ..., N$ and modulating frequencies $j = 1, ..., V$ of LEDs is:

$$D(\mu, i) = \sum_{j=1}^{V} (\mu_{o,j} - \mu_{i,j})^2. \tag{3.1}$$

$\mu_{o,j}$ is the observed RSS value for frequency j and $\mu_{i,j}$ stands for the i^{th} fingerprint in the collection with frequency j. A fingerprint that yields the lowest Euler distance, points out the target's position. [7]

In [44], the authors propose a wireless sensing positioning system that utilizes LED ceiling lights. Each LED has a unique address. The predetermined address and the transmitted data help evaluate the target's position. The correlation between the address and the data extracts the target's position. The system tests use a rectangular environment of 1 m × 1.2 m × 1 m. The maximum distance error is 12.46 cm, and the error mean value is 4.38 cm.

3.1.3 Vision analysis

The vision analysis is an approach that utilizes a geometrical relationship between a camera's 3D position within the natural world and a two-dimensional position of a projected item by the image sensor. The analysis uses a pinhole camera model. Fig. 3.3 illustrates the idea of the model.

Figure 3.3 shows three coordinate frames. The first frame with the origin O_c is a three-dimensional frame representing a figure's coordinates. The second projection with centre O_i represents a two-dimensional image coordinate frame. Finally, the frame with origin O_w captures the real-world position coordinates. Further, the frames' name and their origin name are the same. Frame O_w describes the position of an item and the camera in the real-world coordinates. Frame O_c defines the position of the image in the camera. Lastly, projection O_i captures the location of a pixel in an image. [40]

For an object M with real-world coordinates $M(X_w, Y_w, Z_w)$ and the projected coordinates $M_i(X_i, Y_i)$. The camera coordinates of the item M are $M_c(X_c, Y_c, Z_c)$. Equation 3.2 describes the conversion relationship between frame O_c and the image frame O_i as:

$$\begin{aligned} Z_C[X_i\, Y_i\, 1]^T &= R[X_c\, Y_c\, Z_c\, 1]^T, \\ R &= \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \end{aligned} \tag{3.2}$$

where R is the camera correction matrix, and f represents the distance between point O_c and the image projection. Lastly, the conversion relation between the coordinates of the O_i frame and the O_w's

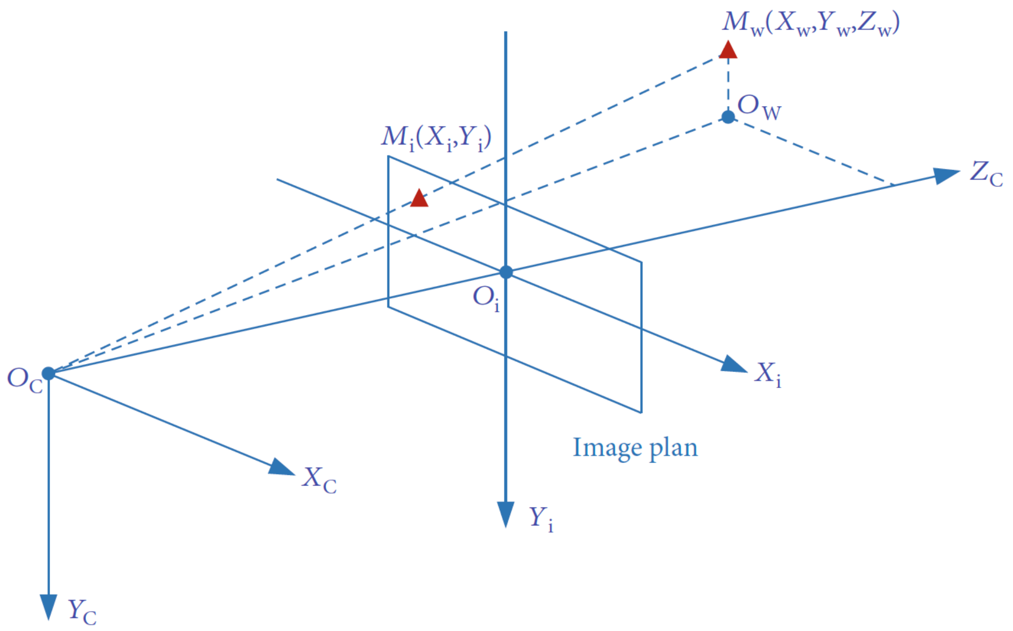


Figure 3.3: Pinhole camera model. Adopted from [40].

coordinates is:

$$[X_c\ Y_c\ Z_c\ 1]^T = \begin{bmatrix} N & t \\ 0 & 1 \end{bmatrix} [X_w\ Y_w\ Z_w\ 1]^T, \tag{3.3}$$

where N is a 3 × 3 orthogonal matrix and t represents 3 × 1 translation vector. [40]

In [45], the authors explore a VLP system based on an unmodified smartphone camera. Slightly modified LED lamps transmit their identifiers and location using On-Off Keying modulation. The smartphone exploits the rolling shutter effect of the CMOS camera embedded in the device to read the emitted data and determine its location. Furthermore, the paper presents an approach for using the camera as an AoA sensor and suggests optimization techniques for maximizing the positioning accuracy. The authors achieved decimeter-level accuracy and 3°orientation error.

In [46], Yang et al. designed a polarization-based modulation, which solves the flickering problem of the intensity-based systems. The technique reduces the required resources of the receiving end and allows the application of devices such as smart glasses. Moreover, the scheme applies to any light source and therefore eliminates the need for a LED. The proposed method called the *PIXEL* incorporates a colour-based modulation scheme, an adaptive downsampling algorithm to prevent issues related to the uneven sampling of cheap cameras and finally, an optimization method for positioning algorithms. Experiments show a 300 mm positioning error in 90% of test cases for the receiver's Central Processing Unit (CPU) frequency as low as 300 MHz and camera resolution of 60 × 80 pixels.

3.1.4 Triangulation

The triangulation algorithms exploit the triangle's geometric properties to determine the target's location. The triangulation techniques fork into two classes: angulation and lateration. As the name suggests, the angulation methods use measured angles corresponding to the transmitting stations to locate the target. AoA is the typical and widely explored method of angulation. Lateration approaches utilize the distances between the target and the transmitter to find the position. The commonly used techniques are RSS, ToA and TDoA. [40]

Received Signal Strength (RSS) methods obtain the signal's strength to determine the target's position. Perceiving a signal's strength is easier compared to other characteristics. Only a single photodetector can read the signal's intensity. The signal's amplitude behaves according to the channel model. Therefore, the intensity reduces as the distance between the mobile device and the TX station increases. [7]

In [47], the authors combine RSS with Optical Orthogonal Codes (OOC). The OOCs encrypt the transmitters' identification so that the receiver can distinguish between different light sources. The system utilizes the trilateration RSS to calculate the position of the receiver. The mean positioning error is 8 cm in 12 m × 35 m.

In [48], Hu et al. designed the Pharos system, which employs beacon lights modulated with Binary Frequency Shift Keying (BFSK) modulation scheme. An algorithm converts the captured signal strength with a mobile device to a distance. Moreover, the algorithm accounts for potential lamp dimming. The positioning uses multilateration principles. The system exploits channel hopping to avoid collision between different illuminating sources. Lastly, the positioning algorithm assumes that the receiver faces squarely upwards towards the ceiling and the LEDs orient downwards towards the receiver. Moreover, the framework utilizes the Least Mean Square (LMS) method to determine the target's location. The practice reaches submeter accuracy for rooms of sizes 5 m × 8 m and 3.5 m × 6.5 m.

In [49], the same authors improve the system's robustness and positioning accuracy. The approach accounts for the imperfect orientation of the receiver or a lack of transmitting beacons. The experiments illustrated an accuracy of 0.4 m in 20 m × 20 m × 3 m.

Received Signal Strength Ratio (RSSR) utilizes the ratio of distance converted from the received power from the beacons. The advantage of RSSR over RSS is in higher accuracy due to the method cancelling the non-zero irradiance angle of light. However, this method is only applicable if the receiver's surface is parallel to the LED lamp, which is unrealistic in most applications. Therefore, this paper does not discuss these methods further. [40]

Time of Arrival (ToA) became famous as the GPS positioning scheme. The distance between the transmitter and the receiver depends on the time of travel of the signal. Multiplying the travel time by the signal's propagation velocity returns the distance between the two points of interest. However, this method requires perfect time synchronization between the transmitter and receiver. Therefore, the realization of a ToA system is problematic and disadvantageous. [40] [7]

In [50], the authors present a simulation of ToA indoor positioning. The arrangement uses Orthogonal Frequency Division Multiplexing (OFDM) to separate the LEDs. The experiment assumes perfect synchronization between the transmitter and the receiver. The calculation results show accuracy ranging from 6 cm to 2 cm.

Time Difference of Arrival (TDoA) offers a feasible alternative to ToA. Researchers often prefer TDoA over ToA since it does not require time synchronization of the receiver and the transmitters. However, the synchronization of transmitting stations remains. [7]

Light beacons synchronously transmit signals to the receiver. The various distances between TX and RX induce various signal arrival times. TDoA always subtracts two different arrival times to estimate its positioning between two transmitting stations. The mobile device lies in equidistant positions between the two emitters, hence on a hyperbola. The system requires three transmitting stations to locate the target in a 2D plane. [40]

In [51], Choi et al. propose a Location-Based Service (LBS), which exploits the TDoA algorithm. The system employs four LED lamps and a Binary-Phase Shift Keying (BPSK) modulation scheme, surpassing other modulation methods regarding Bit Error Rate (BER). The nonlinear least square algorithm estimates the target's location. The simulations show an average location error of 0.14 m.

Paper [52] proposes a TDoA positioning system using a single photodetector. The base stations transmit pilot signals that do not carry transmitter identifications. The pilot signals are rectangular pulses that are sent periodically from each station. The receiver uses cross-correlation to distinguish each transmitting station and retrieves the signal's arrival time. The paper simulated the positioning service in a room of 5 m × 5 m× 3 m and showed a promising positioning error of 3.9 cm.

Figure 3.4 illustrates the idea of the **Angle of Arrival (AoA)**. The mechanism exploits the signal's angles of arrival to estimate the mobile station's location. The process requires only two stations to estimate the target's position in a 2D plane and three stations for 3D positioning. Furthermore, the technique does not obligate any time synchronization between the nodes, similarly to RSS.

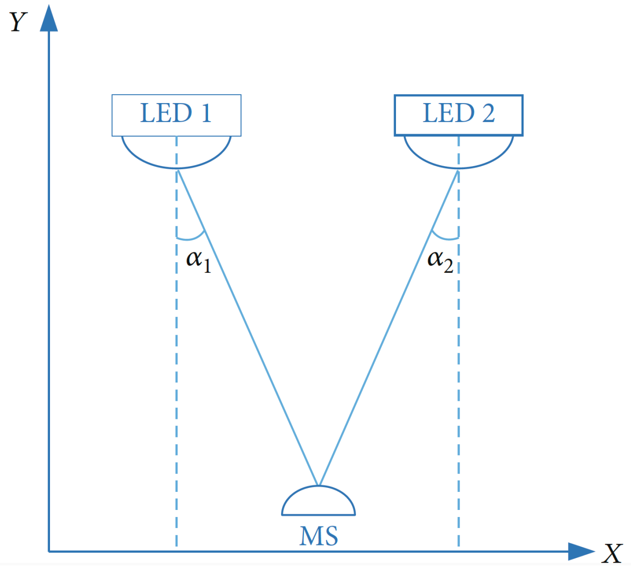


Figure 3.4: Localization based on Angle of Arrival. Adopted from [40].

All LED light signals follow the *cosine emission law*, sometimes called *Lambert's emission law*. The law states that the luminous intensity from an ideal diffuse radiator is directly proportional to the cosine of the angle between the surface normal and the direction of the emitted light. Equation 3.4 mathematically describes the law as:

$$I = I_o \cdot \cos(\theta), \tag{3.4}$$

where the angle θ [rad] is the angle between the emitter and the surface normal, I_o [cd] stands for the radiant intensity, and the emitter, and I [cd] refers to the perceived luminous intensity.

In [53], the authors demonstrate an Optical Wireless Location (OWL) system. They propose a corner-cube photodetector that measures differential currents. The experiments suggest an ideal value of incident intensities, which should stay above 0.2 μW/cm^2. Moreover, the paper shows that the photodetector should be placed at least 0.5 m from the reflective surfaces to attenuate the multipath effects. Finally, the research finds an ideal position for the receiver to minimize the positioning error.

In conclusion, the article mentions that increasing the number of detectors increases the system's precision. The method achieved a four times lower positioning error than a tested RSS method. The absolute positioning error is 5 cm.

Yasir M. et al. [54] introduce an indoor positioning system, which utilizes a photodiode and an accelerometer. The receiver measures the received signal intensity and corresponding orientation of the mobile station. The proposed algorithm then computes the distance between the beacons and the RX. Time-division multiplexing secures an undisturbed transmission of the signals. The paper presents simulations and experiments in an office environment to prove the system's functionality. The achieved mean positioning error is below 0.25 m, and the authors mention that the orientation of the receiver plays a vital role in the positioning accuracy.

A succeeding paper [55] improves the system's capabilities proposed in [54]. The receiver utilizes multiple optical receivers and an accelerometer to determine the distance and orientation. The paper shows five added features to the previous design:

1. The receiver can be mobile.
2. Experiments show a positioning speed of 2.5 ms.
3. The beacons' height must not match.

4. No requirement for the receiver's height.

5. Transmitter's and receiver's normals do not need to align.

The average localization error is below is 6 cm, even for a target moving with a velocity of 1.3 m/s.

3.1.5 Hybrid algorithms

The hybrid algorithms combine the above-described methods to improve the positioning performance or decrease the system's requirements.

Article [56] suggests a two-phase hybrid algorithm. The algorithm combines AoA and RSS to determine signal strength, azimuth, and elevation. The first RSS stage coarsely locates the target. The second phase estimates the angle between the transmitter and the receiver based on the received and theoretical power ratio. Finally, the quadratic programming approach finds the delicate location of the target using the previously measured data. The two-phase approach solves the problem of the triangulation failure. When the triangulation fails, the algorithm's output is a coarse location rather than a localization failure. The simulations illustrate the median coarse phase accuracy of 34.88 cm and the precise accuracy of 13.95 cm.

In [57], Japanese researchers utilized a *LEGO* robot equipped with a compass, visible light receiver and gyro sensor to test a positioning algorithm based on visible light and dead reckoning. Specifically, the algorithm detects the absolute position using the visible light in the first phase. The second stage retrieves an output of the geomagnetic sensor, gyroscope, and Kalman filter for the dead reckoning. The experiments exploited a moving robot that sent the gathered data into a personal computer. The results report an error of 46.62° of the geomagnetic sensor, gyro sensor's error of 14.17° and finally, 5.95° error of the Kalman filter. The achieved localization error is 10.5 cm.

Nguyen et al. [58] utilize a photodiode array and a hybrid algorithm of AoA and RSS to address the positioning of mobile robots. The system's simulations reveal that the positioning error depends on the number of photodiodes in the array. Moreover, the accuracy is also heavily influenced by the path loss of the environment. The paper mainly aims to improve the positioning accuracy by employing the Extended Kalman Filter (EKF). EKF improved the system's performance and alleviated its accumulative errors.

3.2 Visible light receiver

The visible light is a sensor converting a light signal to an electrical signal. Some popular sensors are phototransistor, photodiode, and image sensors. Figure 3.5a illustrates the principle of photoelectric detection. When light hits the depletion layer, its energy forces the low-energy state electrons to move to a high-energy state. This process forms photogenic electron-hole pairs. The internal electrical field enforces a particle flow making the electrons move to the N region and the holes to the P region, thus causing the conversion of light signal to electrical. [40]

The preferred sensors for VLP are a photodiode and an image sensor. The photodiode senses light signals and can determine the user's location. Therefore, the photodiode systems are passive positioning systems. The image sensor is an active positioning system since it captures a series of images and determines the user's position. [40]

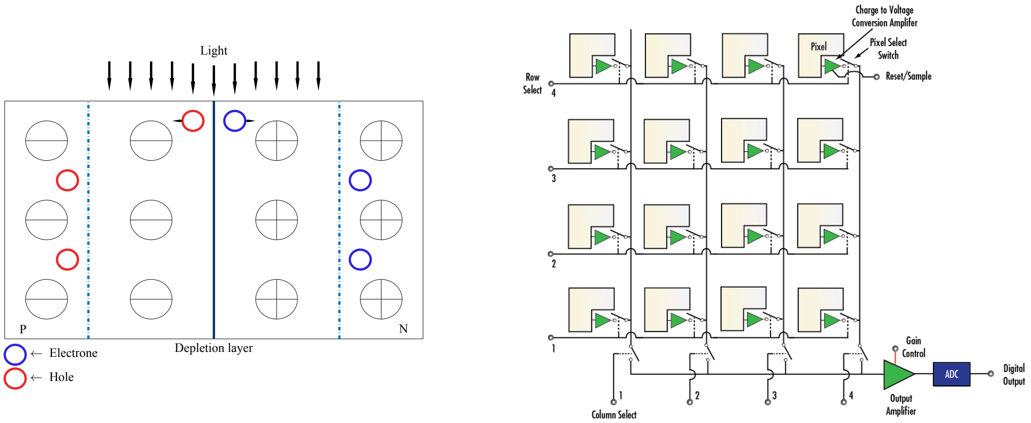
3.2.1 Photodiode

A photodiode is a cheap, accurate, low noise component highly adapted for indoor positioning. It can detect the signal's intensity and arrival time due to high response behaviour and separate signals of different frequencies. Its wide response frequency bandwidth is a significant advantage over the image sensor. Researches studying proximity, fingerprinting, or triangulation mainly exploit photodiodes as VLC receivers. [40]

3.2.2 Complementary Metal-Oxide-Semiconductor architecture

Figure 3.5b shows the block diagram of a CMOS. In each sensor cell, a photosensitive pixel converts its charge to voltage. Rows and columns multiplex the signal from the sensor to multiple DACs. Therefore, CMOS is a digital device. Each pixel is a photodiode with three transistors that reset or activate the pixel, amplify and convert the electric charge and multiplex the signal. [59]

CMOS sensor most often utilizes an electronic rolling shutter for multiplexing the signals. However, additional transistors at the pixel's site can facilitate the global shutter multiplexing. Subsection 3.2.3 describes the differences in the multiplexing strategies. [59]



(a) Principle of photoelectric detection. Image inspired by [40].

(b) Block diagram of CMOS. Adopted from [59].

Figure 3.5: Photoelectric detection and CMOS diagram.

3.2.3 Image Sensors and signal multiplexing

The image sensor captures a series of images, and a processing algorithm determines the presence or absence of the light beacons by analyzing the series. The images make separating the light signals from the external light easier than the photodiode's electrical signals. However, the image sensor's frame rate is limiting. Its receiving frequency band is much narrower than that of the photodiode. The image sensors fork into two categories: CMOS sensors and Charge Coupled Devices (CCD). [40] [25] [60]

Figure 3.6 shows the principle of image acquisition by CMOS and CCD sensors. In the case of the CCD device, the entire surface of the sensor is sensitive and reads data simultaneously. This technique is the global shutter mechanism of the CCD sensor. The shortcoming of the global shutter mechanism stands in its low data reception. According to *Nyquist sampling law*, the data transmission must be lower than the sampling rate of the camera. The CMOS sensor uses the rolling shutter mechanism. Each row of pixels is active in its time frame, and the rows follow an activation sequence, meaning that the exposure and data reading comply with the same sequential behaviour. The frame comprises all acquired rows. The integration begins by resetting the first row and ends by reading out the last row. When a LED switches on and off during this integration period, the sensor's output is bright and dark stripes on the image. These stripes can represent bits of data that a VLC algorithm processes. Suppose the shutter time is lower or approximately equal to an inverse of the switching frequency of the LED, causing the CMOS sensor to obtain only the average light intensity, which makes the LED appear as half turned on. The data rate of the CMOS sensor is only kbit/s of data. However, the data rate presents no obstruction for an indoor positioning system since sending identification or positioning information is still viable. The blooming effect, the read-out time gap or Extinction Ratio (ER) present a more significant challenge than the data rate limitation. [60] [25]

The overlapping behaviour of the CMOS shutter mechanism reduces the effective number of bits in each image frame. The read-out time gap or the frame-to-frame processing time gap appears whenever

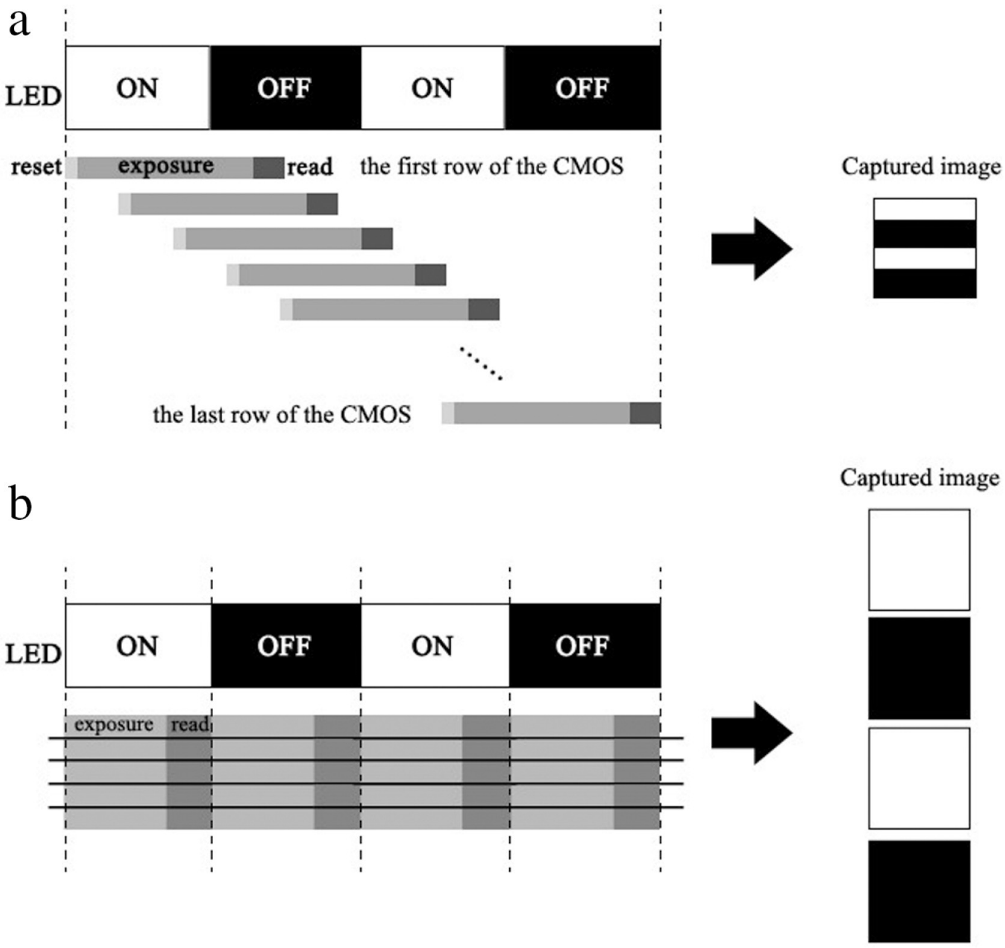


Figure 3.6: Image acquisition in the image sensor. a) The rolling shutter of the CMOS sensor. b) The global shutter of the CCD sensor. Caption and image acquired from [60].

the rows merge into a frame. Since the sensor is inactive during this period, data loss arises. Designing a correct frame structure and ensuring that each frame contains the complete data packet solves this issue. [60]

Blooming is a saturation of the image sensor. When the perceived light signal's intensity is too high, particles from the saturated pixels flow into the neighbouring pixels. This effect might cause the dark stripes on the image to completely disappear and generally decreases the contract ratio between the dark and bright stripes. This effect causes the data transmission to fail. [60]

Lastly, uneven light exposure of the sensor surface causes high data fluctuation and ER variation. The ER is a ratio of greyscale high and low value. When the ER is too low, demodulation becomes impossible. Therefore, the noise filtering and thresholding scheme increases transmission distance and attenuates environmental disturbances. [60]

3.3 ESP32-CAM as the receiver

This research proposes deploying an image sensor for the Visible Light Communication (VLC). According to research [61], the photodiode is sensitive to the light beam's direction, limiting the receiving end's mobility. Moreover, the localization accuracy depends on the angular and signal strength measurements, and the positioning requires solving quadratic equations leading to a large percentage of error. On the other hand, image sensors are immune to reflected light. Lastly, today's mobile phones come with embedded image sensors, and therefore the same methods applied to a CMOS in the proposed environment can be used in the commercial domain.

The proposed mobile platform uses a microcontroller *Ai-Thinker ESP32-CAM* [11]. The evaluation board hosts a camera module slot, Secure Digital (SD) card slot and a Wi-Fi connection interface. Table 3.1 digests the essential features of the development board. *ESP32-CAM* is one of the most miniature Systems on Chip (SoCs) modules that grant 802.11b,g,n Wi-Fi and BT/BLE wireless communication protocols. Furthermore, the microcontroller has a 32-bit CPU with clock speed up to 240 MHz and computing power up to 600 Dhrystone Million Instructions per Second (DMIPS). The SoC has built-in 520 kB Static Random Access Memory (SRAM), and the development board accommodates 8 MB of Pseaudostatic Random Access Memory (PSRAM). The device can communicate via UART, SPI, and I2C communication protocols and provides PWM, Analog-to-Digital Converter (ADC), and DAC input/output signal interfaces. Moreover, the Wi-Fi connection supports Station Mode (STA)/AP/STA+AP modes. The board enables technologies such as Smart Config and AirKiss. The firmware can take advantage of *FreeRTOS* [62] (a Real-Time Operating System (RTOS) for microcontrollers) and uploads either through a serial port or remotely.

ESP32-CAM
802.11b/g/n Wi-Fi BT/BLE SoC module
Low power 32-bit CPU
Up to 240 MHz clock speed, summary computing power up to 600 DMIPS
Built-in 520 kB SRAM, external 8 MB PSRAM
UART, SPI, I2C, PWM, ADC, DAC
OV2640 [63] and OV7670 [64] cameras, built-in flash lamp
Supports image Wi-Fi upload
Supports TF card
Embedded LwIP and FreeRTOS
STA/AP/STA+AP operation mode
Smart Config/AirKiss technologies
Serial port local and remote firmware upgrades (FOTA)

Table 3.1: Features of *ESP32-CAM*. [11]

Finally, the SoC supports two camera modules, *OV2640* [63] and *OV7670* [64]. Moreover, it has a built-in flash lamp, which extends the device's capabilities. Both of these devices are CMOS type cameras with a rolling shutter mechanism. Apart from the sensors, the devices merge an analogue signal processor, ADC converter, Digital Signal Processor (DSP) and other digital and analogue peripherals. The integrated structures allow a straightforward application of both sensors. Table 3.2 summarizes the critical features of the camera chips.

The cost-effectiveness of the sensors makes them a popular choice for development boards and algorithm testing. Camera *OV2640* is more suitable for the desired application. The most significant advantage of the chip is its resolution and frame rate. The last parts of the chapter described the importance of frame rate for data transfer. Moreover, camera *OV2640* fuses an on-chip JPEG encoder, making image processing more straightforward and reducing memory requirements for the microcontroller's internal RAM.

The board has an integrated antenna for wireless communications, and it introduces an *IPEX* connector for an external antenna. Connecting the external antenna improves the communication distance and reliability. However, the antenna requires resoldering the resistor at the *IPEX* connector to the second position. Figure 3.7 presents the front and back views of the board. The *IPEX* connector and the resistor are in the left panel below the integrated antenna. The right panel shows the SD card connector and the camera interface on the same side as the built-in flash lamp. The board comes with a **RST** button. This button resets the firmware's cycle and controls the flashing mode. The flashing mode is for uploading a new program. The board needs to restart before uploading new firmware.

Lastly, Fig. 3.8 shows the pinout of the development board. The board has sixteen accessible pins. Three of the pins are **GND** grounding pins, and two pins are the power supply for either 5 V or 3.3 V.

OV2640 and OV7670

Feature	OV2640	OV7670
Active array size	1600×1200 (UXGA)	640×480 (VGA)
Power consumption	125 mW UXGA YUV Mode 140 mW UXGA compressed mode Standby: 600 uA	60 mW VGA YUV Format Standby: <20 uA
Output formats	YUV (422/420)/YCbCr422 RGB 565/555 8-bit compressed data 8-/10-bit Raw RGB data	YUV/YCbCr 4:2:2 RGB 565/555/444 GRB 4:2:2 Raw RGB Data
Lens size	1/4”	1/6”
Maximum Image Transfer Rate	UXGA/SXGA - 15 fps SVGA - 30 fps CIF - 60 fps	30 fps for VGA
Sensitivity	0.6 V/(Lux-sec)	1.3 V/(Lux-sec)
S/N Ratio	40 dB	46 dB
Dynamic range	50 dB	52 dB
Pixel size	2.2 μm × 2.2 μm	3.6 μm × 3.6 μm
Mechanism	Rolling Shutter	Rolling Shutter

Table 3.2: Features of *OV2640* [63] and *OV7670* [64].

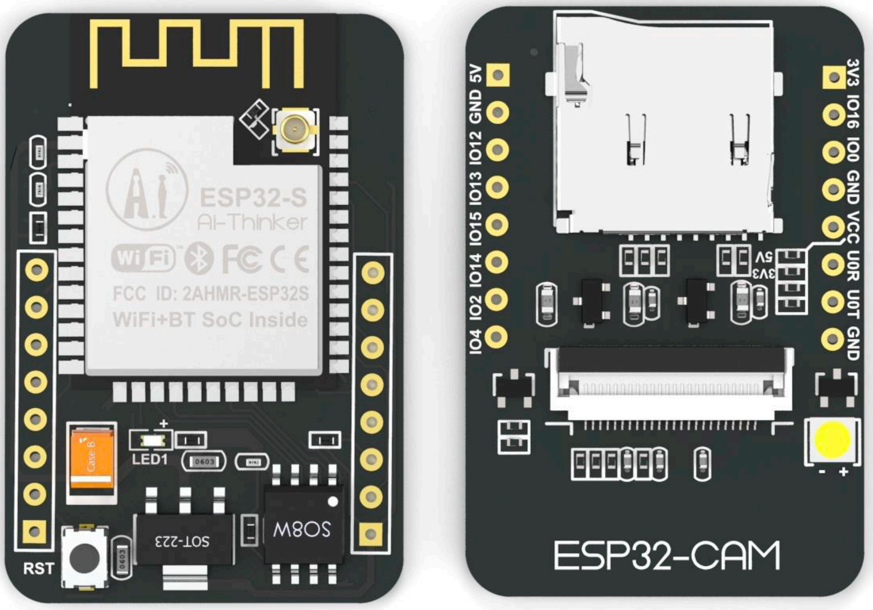


Figure 3.7: *ESP32-CAM* (AI-Thinker module). Image acquired from [11].

GPIO1 and **GPIO3** are serial communication pins. These pins attach the UART interface and upload new firmware into the microprocessor. **GPIO0** determines whether the device is in the flashing mode or not. **GPIO0** needs to be grounded to get the board into the flashing mode. [11]

The board does not come with a USB connector. Thus, firmware uploads via the serial interface. The simplest way of modifying the firmware is through an *FTDI* programmer. The programmers are simple USB to Time To Live (TTL) devices. Commonly, such devices come with a power output pin, and therefore the programmer can power the microcontroller and removes the additional need for a power source. The designed device utilizes a *CP2102* converter [65].

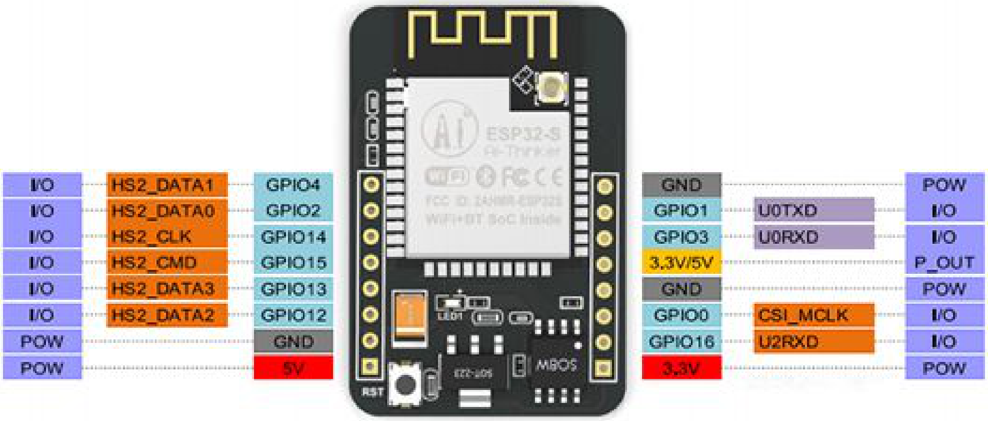


Figure 3.8: *ESP32-CAM* pinout (AI-Thinker module). Image acquired from [66].

Chapter 4

Mobile Platform’s Hardware Realization

The main goal of this project is to create a testing environment for various positioning algorithms. All positioning algorithms need a receiving end as the target of the localization. Chapter 3 describes the importance of the receiver and the selected device. The receiving station should be able to move and communicate with the users. Over the years, researchers developed different realizations of the receiving stations.

Authors in [67] focused on a three-dimensional VLP system. For testing the position accuracy, the authors developed an Field Programmable Gate Arrays (FPGA) based receiver that sits on construction, and the developers moved the entire construction. Such a system is applicable in laboratory environments, however, it would require modifications for a commercial application.

In [68], the researchers realized a novel indoor positioning system using a smartphone camera and a smartphone application that communicates with a personal computer. The computer then estimates the target’s location by executing the positioning algorithms on the received data from the smartphone. Such a model might have a potential real-life application since a smartphone equipped with a camera is today an ordinary possession.

This thesis presented several papers focusing on visible light positioning systems for robotics. Examples are [58] or [57]. These papers utilize photodiodes or image sensors installed on the robot’s body to collect data. The robots transport multiple sensors, including gyroscopes or accelerometers and communication interfaces, offering the moving functionality and the communication means.

Since the proposed receiver is the *ESP32-CAM* [11], using the camera *OV2640* [63], it already possesses a communication interface. The most straightforward application of the microcontroller is building a robotic body that carries the microcontroller itself, its power supply, the communication equipment, the moving parts and the driving circuit for the moving parts. The following parts of the chapter describe the hardware components and the realization of the mobile platform proposed for the VLP testbed. Maker forum [69] inspired the robot’s design.

4.1 Robot’s components

The mobile platform composes of several essential components. Apart from the microcontroller, the main parts of the robot are the power source of the entire device, the power supply for the microcontroller, which must be different from the motors’ power supply, the motors’ driver circuit, and the antenna, and the motors. This section illustrates the selected components and describes their function.

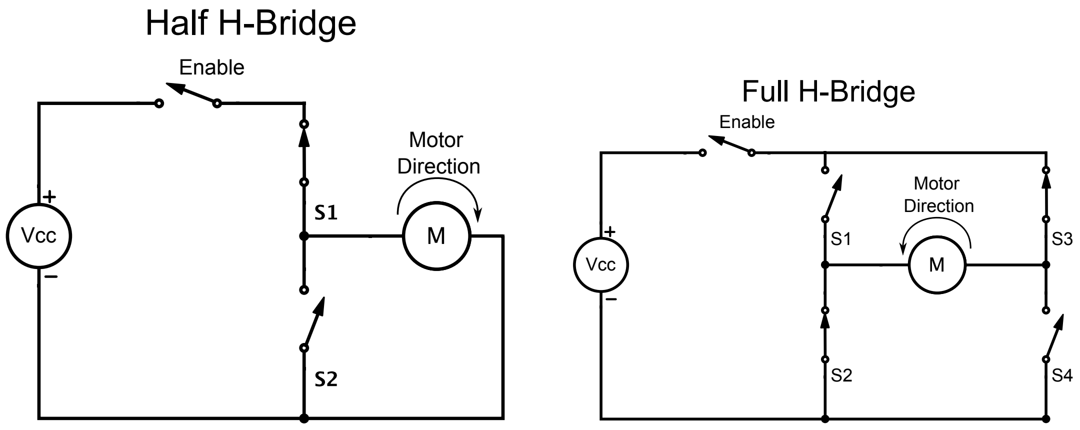
4.1.1 Motor driver

The proposed robotic platform uses two DC motors. The microcontroller needs a motor driver or *H-Bridge* to apply the necessary power to the motors. Motors are components generally having low resistances from 10 Ω to 100 Ω. Therefore, if the voltage requirement rises, the current requirement increases dramatically. The name *H-bridge* comes from the schematic diagram representation, which resembles the letter H with four switching elements and the load placed as the connecting crossbar. *H-Bridges* control the amount of power applied to the load through PWM signals. [70]

4.1.2 H-Bridge function

H-Bridges divide into two categories: *Half H-Bridge* and *Full H-Bridge*. Figures 4.1 show the diagrams of the two bridges.

The *Half-Bridge* comprises one voltage source, motor, and three switches: enable switch, switch S1 and switch S2. A PWM signal drives the enable switch. Altering the frequency and duty cycle of the



(a) Half H-Bridge Schematic. Adopted from [70].

(b) Full H-Bridge Schematic. Adopted from [70].

Figure 4.1: Half H-Bridge and Full H-Bridge schematics. Images adopted from [70].

PWM controls the average value of the signal and, therefore, the amount of current flowing through the motor. The current flow defines the rotation speed of the motor. Switches S1 and S2 are unnecessary for the function of the *H-Bridge*. Removing switches S1 and S2 and connecting the motor directly to the Enable switch results in the same effect. However, these switches become essential in the case of the full bridge. [70]

The *Full-Bridge* adds two more switches, S3 and S4, as Figure 4.1 suggests. The switches control the current flow's direction and, therefore, the rotation direction of the motor. Table 4.1 shows the bridge's operation. [71]

S1	S2	S3	S4	Result
1	0	0	1	Clockwise rotation
0	1	1	0	Counterclockwise rotation
0	0	0	0	Motor coasts
1	0	0	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	
0	1	0	1	Motor brakes
1	0	**1**	0	
x	x	1	1	Short circuit
1	1	x	x	

Table 4.1: *Full H-Bridge* logic table. [71]

4.1.3 Selected motor driver

The proposed robotic platform uses a motor driver module based on IC *TB6612FNG* [72]. The integrated circuit enables four DC motor regimes: Clockwise (CW), Counterclockwise (CCW), short break, and stop mode. The module utilizes laterally-diffused metal-oxide-semiconductor (LDMOS) transistors with low ON-resistor. Table 4.2 digests vital features of the integrated circuit. The maximum motor voltage $V_{M_{max}}$ is 15 V. The circuit provides a 1.2 A current on average, and the maximum output current is 3.2 A. Moreover, the output resistance is typically 0.5 Ω, and the IC has a power-saving mode. Lastly, the module's thermal shutdown and low voltage detection circuits shield the circuit from damage.

TB6612FNG
Power supply voltage: $V_{M_{max}} = 15$ V
Output current: AVG$\{I_{out}\} = 1.2$ A, max$\{I_{out}\} = 3.2$ A
Output low ON resistor: $0.5\ \Omega$
(Typical for $V_M \leq 5$ V)
Standby (Power save) system
Modes: CW, CCW, short brake, stop function
Built-in thermal shutdown circuit and low voltage detecting circuit

Table 4.2: Features of *TB6612FNG* motor driver. [72]

4.1.4 Motors

Two generic DC gear motors move the robot. The standard operating voltage of these devices stands between 3 V to 12 V. The maximum load current is approximately 250 mA for one device. The gear ratio is 1:48, and an Electromagnetic compatibility (EMC) isolation shields the motor gears. [73]

4.1.5 Power Supply

The power supply comprises two essential parts: the system's, and the microcontroller's power supply. Pair of *18650 Lithium-ion* batteries power the system. Each battery provides 3.7 V. The output voltage is 7.4 V with the batteries connected in a series. A voltage of 7.4 V is within the operating range of the motors. Moreover, the batteries are rechargeable and have a capacity of 2000 mAh. DC/DC buck converter supplies the microcontroller with the necessary 5 V voltage and required current.

DC/DC converter

The microcontroller requires either 5 V or 3.3 V supply voltage. Since the robot's supply consists of two batteries providing 7.4 V, the system needs a DC/DC buck converter. The platform utilizes a converter based on *LM2596* [74]. The input voltage range of the module is between 3 V to 40 V, and the output ranges from 1.5 V to 35 V. The maximum output current is 3 A. The module has adjustable output voltage. However, the converter can never boost voltage. Information about the module is under source [75].

4.1.6 Antenna

Chapter 3 explains that the *EPS32-CAM* module can use either the integrated or external antennas for wireless communications. The device communicates in the bandwidth from 2400 MHz to 2483.5 MHz. The integrated antenna on the module's PCB has a gain of 2 dBi. [11]

The external antenna used for the platform is a 3 dBi and 2.4 GHz *SMA* equipment. It comes with a *Pigtail* cable that has an impedance of 50 Ω, 20 cm length and diameter of 1.13 cm. [76]

However, there are no official sources available that show the difference between the signal strength of the integrated antenna and the external antenna. Since this research suggests that changing the antennas is necessary to secure higher quality communication, a measurement that proves the statement is necessary.

RSS measurement

Figure 4.2 depicts the experimental setup of the measurement. The image shows the robotic platform and a Wi-Fi router as the access point. Both the router and the robot sat on the ground during the experiment. There were no moving objects or obstacles present in the measurement area during the experiment. The router used for the experiment is the *AC1200*, Wireless Dual Band Router Model NO. *A800R* from TOTOLINK Company [77]. The *ESP32-CAM* used the 2.4 GHz Wi-Fi communication with a bandwidth of 20 MHz on the 6th channel. The applied mode of the physical layer is 802.11n.

The target moved towards the Wi-Fi router. The module acquired data every 20 cm of the movement. The starting position was 5 m from the router to the front wheel of the robotic platform. The distance from the antenna to the wheel is 21 cm for the integrated antenna and 26 cm for the external antenna. The final position was 20 cm in front of the router. The robot gathered four data samples for every discrete distance. Figures 4.3 illustrate the difference in the received signal strength between the *ESP32-CAM*'s integrated antenna and the external camera. The cross points in the graphs mark the measured data.

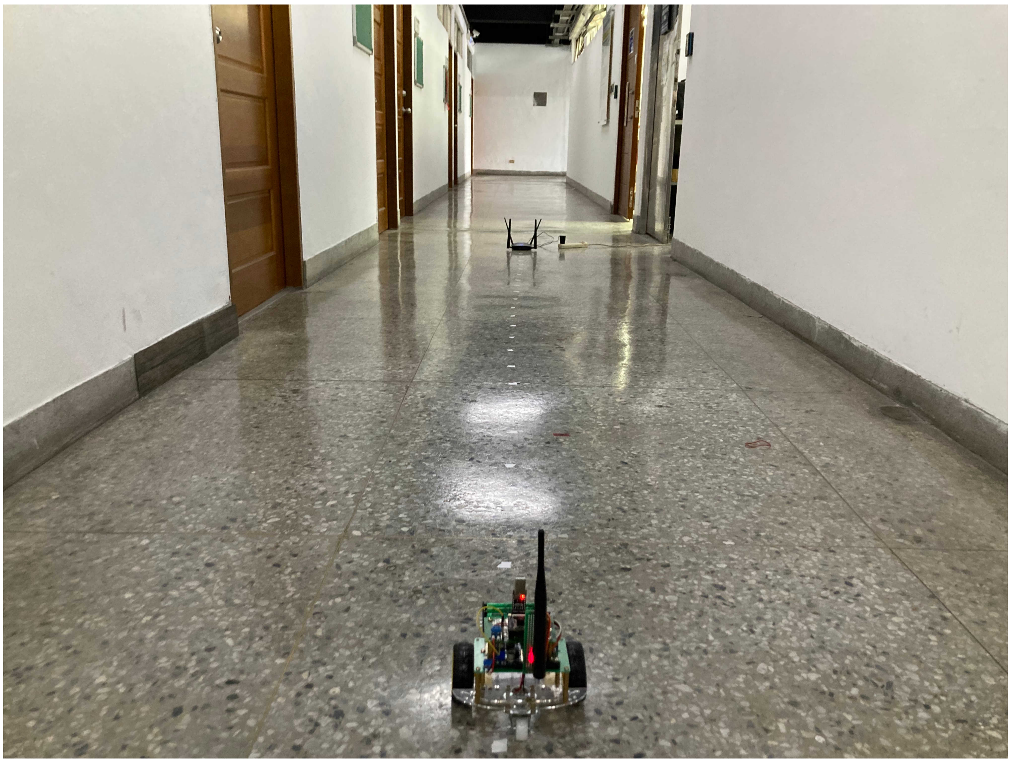
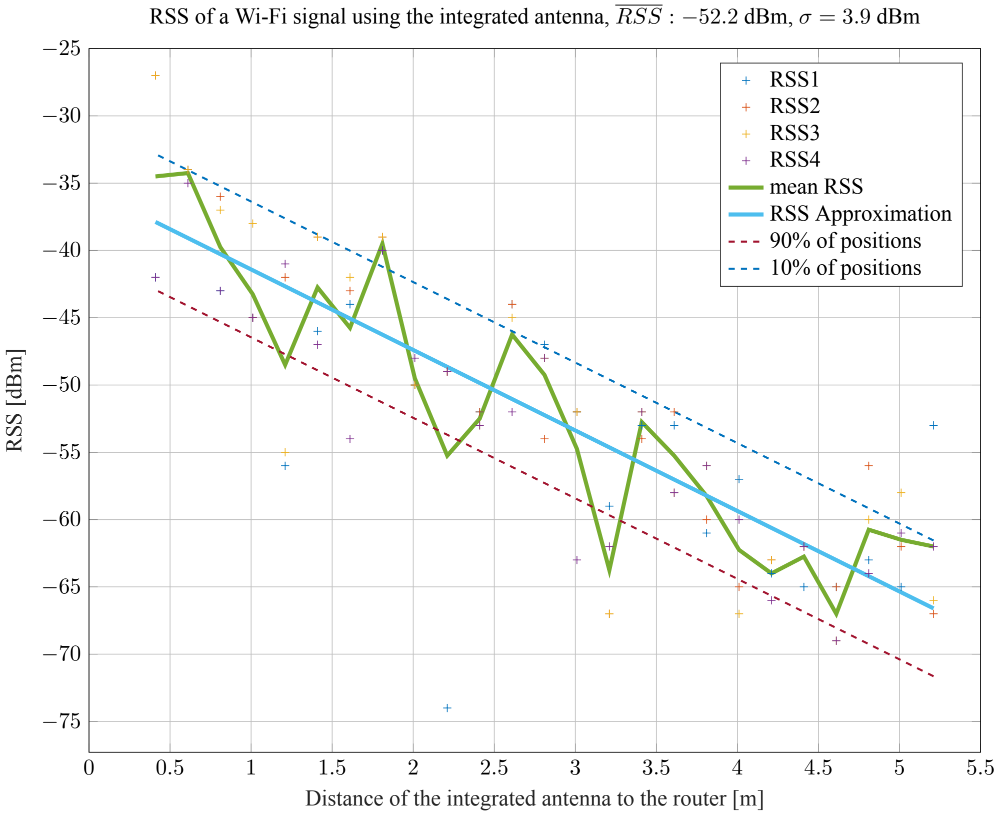


Figure 4.2: Experimental setup of the RSS measurement in a school hallway.

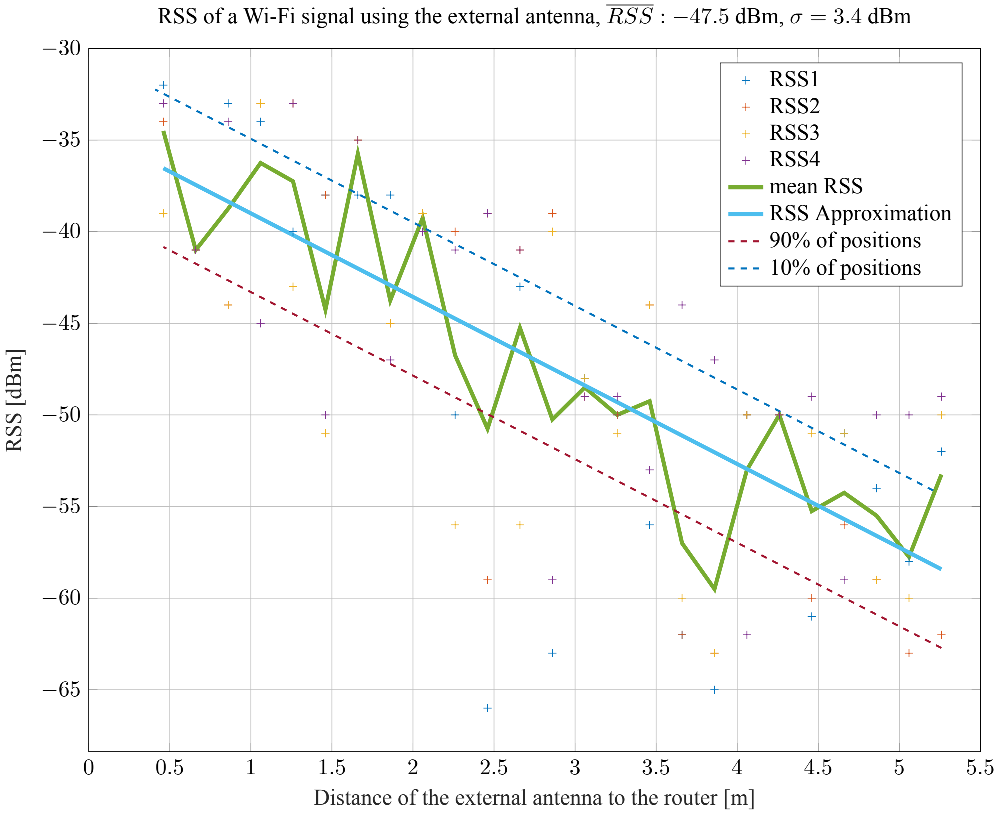
Graphs 4.3 depict the received signal intensity [dBm]. The green waveforms show the average values of RSS for all four measurements at every discrete distance. The blue lines show a polynomial fitted curve of degree 1. The dashed lines mark the percentage of positions where the received power is higher than the corresponding curve's. The thresholds use the standard deviation of the measured data and the mean values. The total average received power decreases with the increasing distance, as expected. The results of the measurement are in Table 4.3.

	Total $\overline{RSS}$ [dBm]	Total σ [dBm]	Slope coefficient [dBm/m]
External antenna	-47.5	3.4	-4.6
Integrated antenna	-52.2	3.9	-6.0

Table 4.3: Results of the RSS measurement using the *ESP32-CAM* module and its integrated and external antenna.



(a) Measurement of the RSS of a Wi-Fi signal using *ESP32-CAM* and its integrated antenna.



(b) Measurement of the RSS of a Wi-Fi signal using *ESP32-CAM* and an external antenna.

Figure 4.3: The Wi-Fi RSS measurements on a school hallway using *EPS32-CAM*'s integrated and external antenna.

Table 4.3 summarizes the total results of the measurement. The total average value of RSS for the integrated antenna is -47.5 dBm and -52.2 dBm for the external. The total standard deviation of the received power is 3.4 dBm for the external antenna and 3.9 dBm for the integrated antenna. The results show that the overall received power using the external antenna is more significant and stable. The most notable difference in the communication is in the more considerable distances, where the integrated antenna occasionally fails. For super short-range communications, the antenna swap plays an inessential role. Moreover, the results show that the slope of the curve for the integrated antenna is smaller, which means a faster decrease of the received power with the distance. The slope coefficient is approximately -6 dBm/m for the integrated antenna and -4.6 dBm/m for the external antenna.

4.2 Interconnection of the applied circuits

The mobile platform comprises three significant blocks, which secure its proper behaviour. The three blocks are the connection of *ESP32-CAM* and an FTDI programmer, the robot's and microcontroller's power supply, and the interconnection of the motor driver with the motors and the controlling module. This section further describes each block and the coupling of the robot's components.

4.2.1 ESP32-CAM and the FTDI programmer

The robotic platform needs firmware upgrades and should provide serial communication to the user. For the lack of available GPIO pins on the module, the platform can not carry a display or other user interfaces. Therefore, the simplest way of communicating with a user or a developer is through the serial interface. The FTDI module governs both functions, making it the robot's vital element. Figure 4.4 shows the interconnection of a generic FTDI programmer with the *AI Thinker* module. Furthermore, table 4.4 summarizes the interconnections between individual GPIO pins.

The diagram shows two jumper connectors. As was explained previously, flashing of the microcontroller requires that the **GPIO0** is grounded. The first jumper connects **GPIO0** to a **GND** pin. The second jumper connects the 5 V power supply provided by the FTDI to the board. This jumper serves as a security mechanism that prevents a state in which the DC/ DC converter, the FTDI programmer and the microcontroller are all shorted. Therefore, the user can choose whether the FTDI adapter or the DC/DC converter powers the board. Moreover, the FTDI programmer is applicable for firmware flashing or serial communication even when the second power source supplies the board due to the jumper.

ESP32-CAM	FTDI Programmer
5V	VCC (+5V)
U0RXD (GPIO3)	TX0 (TXD)
U0TXD (GPIO1)	RXI (RXD)
GND	GND

Table 4.4: *ESP32-CAM* and the FTDI programmer [65] coupling.

4.2.2 The robot's power supply

This chapter has already described the components used for powering the mobile platform. Diagram 4.5 illustrates the connection of the *18650* batteries to the DC/DC converter and the motor driver. A Single Pole Single Throw (SPST) switch controls the robot's power supply. After the switch, the batteries directly connect through terminal blocks to the DC/DC converter and the motor driver. The converter's output goes from the terminal block to a jumper junction, which controls the microcontroller's power supply. A simple 30 mA red LED signals the presence of the powering voltage. The LED connects to the converter via a 220 Ω resistor, which sets the LED's operating point.

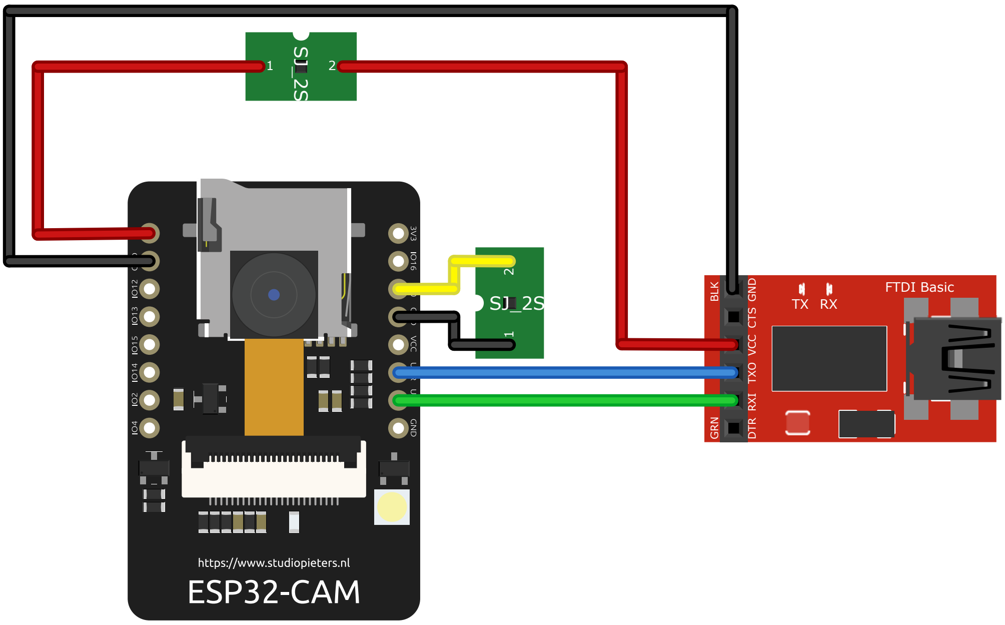


Figure 4.4: Connection of an FTDI adapter to the *ESP32-CAM*.

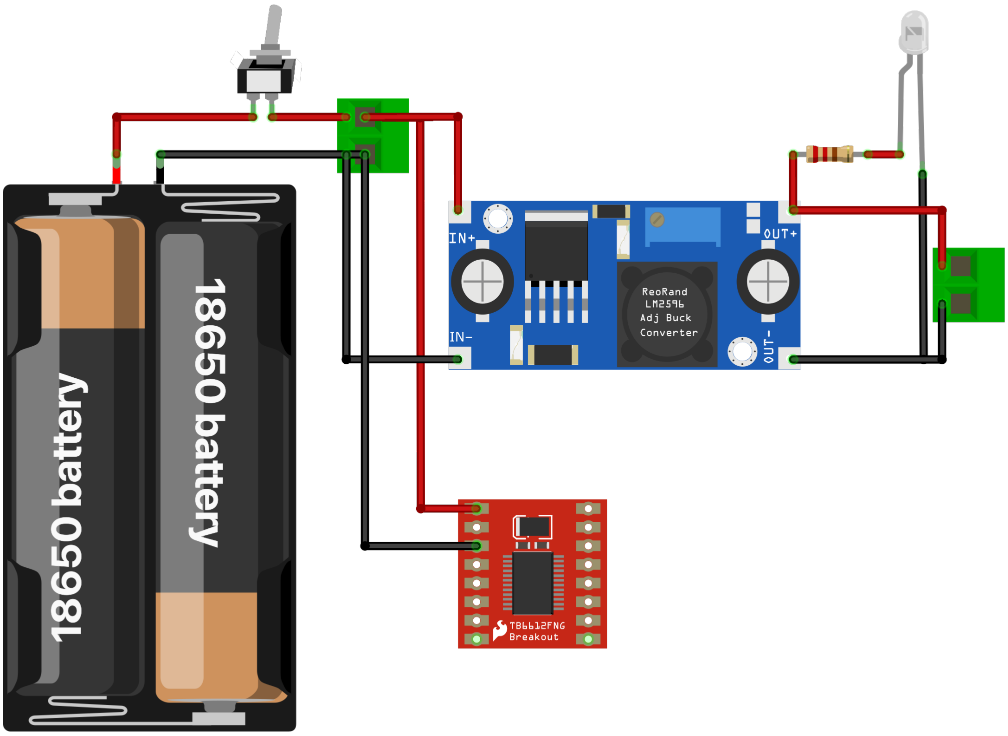


Figure 4.5: Robot's power supply.

4.2.3 Motor driver, button and the ESP32-CAM interconnections

Image 4.6 depicts the connection of the motor driver to the motors and the microcontroller. Table 4.5 summarizes the hookup of the motor driver to the evaluation board. The motor driver module splits into two sides. The left side of the module connects the motor power supply and the motors. The module's right side handles the control signals of the microcontroller. This division makes the module's installation easier.

Pins **PWMA** and **PWMB** control the rotation speed of the motors. Since controlling the left and the right motor at different speeds is undesirable, both pins attach to the module's **GPIO12** pin. The **STBY** needs to connect to the **VCC** pins of both the microcontroller and the motor driver. This pin enables the motor control function of the module. [72]

Lastly, image 4.6 shows a button connecting **GPIO16** and **GND**. This button is a simple interface between the user and the robot. The button manipulates the program inside the device. Chapter 5 describes the function of the button in detail.

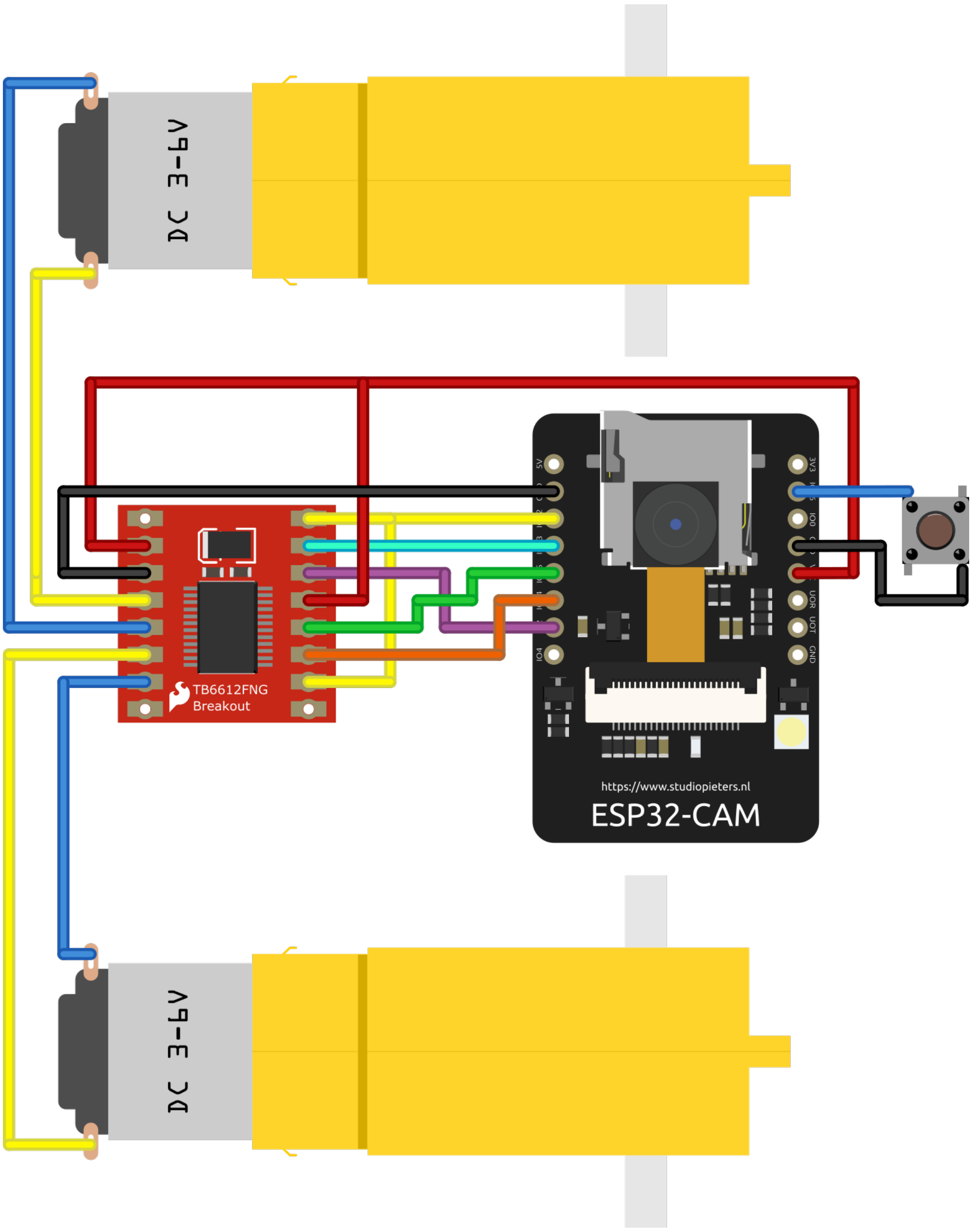


Figure 4.6: The driving circuit and the *ESP32-CAM* connecting a button and motors.

ESP32-CAM	TB6612FNG
VCC	VCC, STBY
GND	GND
GPIO12	PWMA
GPIO12	PWMB
GPIO13	AIN2
GPIO15	BIN1
GPIO14	BIN2
GPIO2	AIN1

Table 4.5: The connection between the *ESP32-CAM* and *TB6612FNG* Dual DC motor driver [72].

4.3 Mobile platform's realization

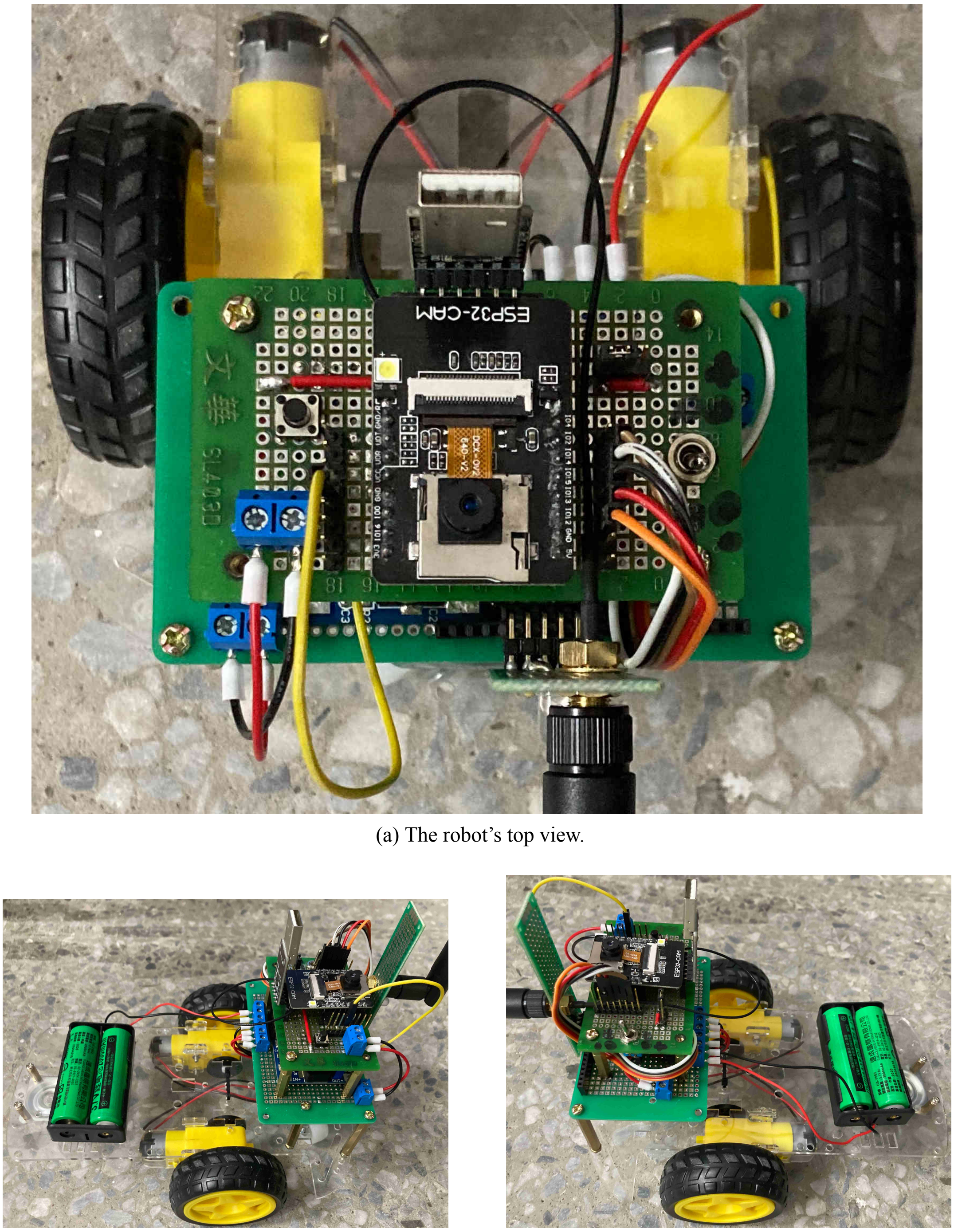
The last part of this chapter describes the functional blocks of the robot. The blocks couple together and compose the mobile platform. The realization of the robot is in Figures 4.7. The images show three views of the robot - a top view and two side views.

The robot's electronic components sit on a prototyping robot chassis kit [78]. All the mechanical parts are a part of the prototyping set. Perforated experimenter's breadboards mount the electronic components and carry the connective wires. The signal connections use 30 gauge wire. The power components interconnect via 25 gauge wires.

The robot consists of three breadboards mounted using metallic spacing columns and screws. The bottom floor of the robot carries the motor driver, the DC/DC converter and the terminal blocks for connecting the power supply and the motors and the microcontroller. The upper level holds the ESP module and the FTDI programmer, the switch for power control and the jumper junction. The ESP32 module hooks up to the motor driver via connecting cables. The last solder breadboard attaches the antenna and the LED with the resistor to the bottom breadboard. Female 2.54 mm signal connectors join the antenna breadboard to the rest of the device.

The ESP module faces upwards with the camera to successfully capture the proposed lights on the ceiling within the testing bed.

Lastly, the robot is back heavy due to the control system sitting at the back of the chassis. An additional castor wheel resolves the weight issue.



(a) The robot's top view.

(b) The robot's left side.

(c) The robot's right side.

Figure 4.7: The realization of the robotic platform.

Chapter 5

The Robot's Firmware

The vital element of the mobile platform is its software. As mentioned previously, the robot utilizes the ESP32 microcontroller as its brain. Since the robot's objective is acquiring image data using the CMOS camera chip, the software must allow total control over the camera's settings. Furthermore, the robot must move according to the user's needs. Both the movement function and image acquisition need a remote control. Therefore, the robot's firmware must utilize a wireless communication protocol.

The robot's firmware is written in *C++*, using the *Arduino* framework. The software comprises five `.cpp` source files and four `.h` header files. The following sections describe every file and its function in detail. The source files organize as follows:

- File `main.cpp` runs the two essential *Arduino* functions `setup()` and `loop()`. More detail is in section 5.1.
- Source file `gpio.cpp` controls the external hardware such as the motors or the *ON-DEMAND* button and defines behaviour for the torch LED. The file's details are under section 5.2
- File `esp32_cameraHandler.cpp` handles the camera's behaviour and defines the microcontroller's main features. 5.3
- The `spiffs_controller.cpp` initializes the Serial Peripheral Interface Flash File System (SPIFFS) and defines the critical functions for its operation. 5.4
- Lastly, source file `wifi_manager.cpp` handles the robot's Wi-Fi connection and initializes the Over-The-Air (OTA) firmware updates. 5.5

The header files arrange in the following manner:

- Header file `gpio.h` defines macros for the peripheral control and declares global variables and functions for the robot's movement.
- File `esp32_cameraHandler.h` defines the selected development board, the camera's GPIOs and a structure for saving the status of the camera's registers. Lastly, it declares functions for controlling the camera.
- Code file `spiffs_controller.h` only declares the functions for initializing and controlling SPIFFS.
- Finally, `wifi_manager.h` defines the timeout for the AP configuration mode, functions for controlling the OTA and the Wi-Fi manager, and lastly, the path to the Wi-Fi configuration file within SPIFFS.

5.1 The main loop

The primary loop and the program's initialization are inside the file `main.cpp`. Figure 5.1 depicts the firmware's flowchart.

The program initializes its variables in the first steps, primarily the movement and server variables. The server utilizes an architecture from the open-source library **ESPAsyncWebServer** available under [79]. Command `AsyncWebServer server(80)` declares the *AsyncWebServer* on port 80. Further, the program initializes the critical registers of the microcontroller in the `setup()` function. Code 5.1 shows the definition of the method. Firstly, the function starts the USART communication with a 115200 baud rate. Following the serial communication initialization, the function configures the peripherals and external hardware in the following order:

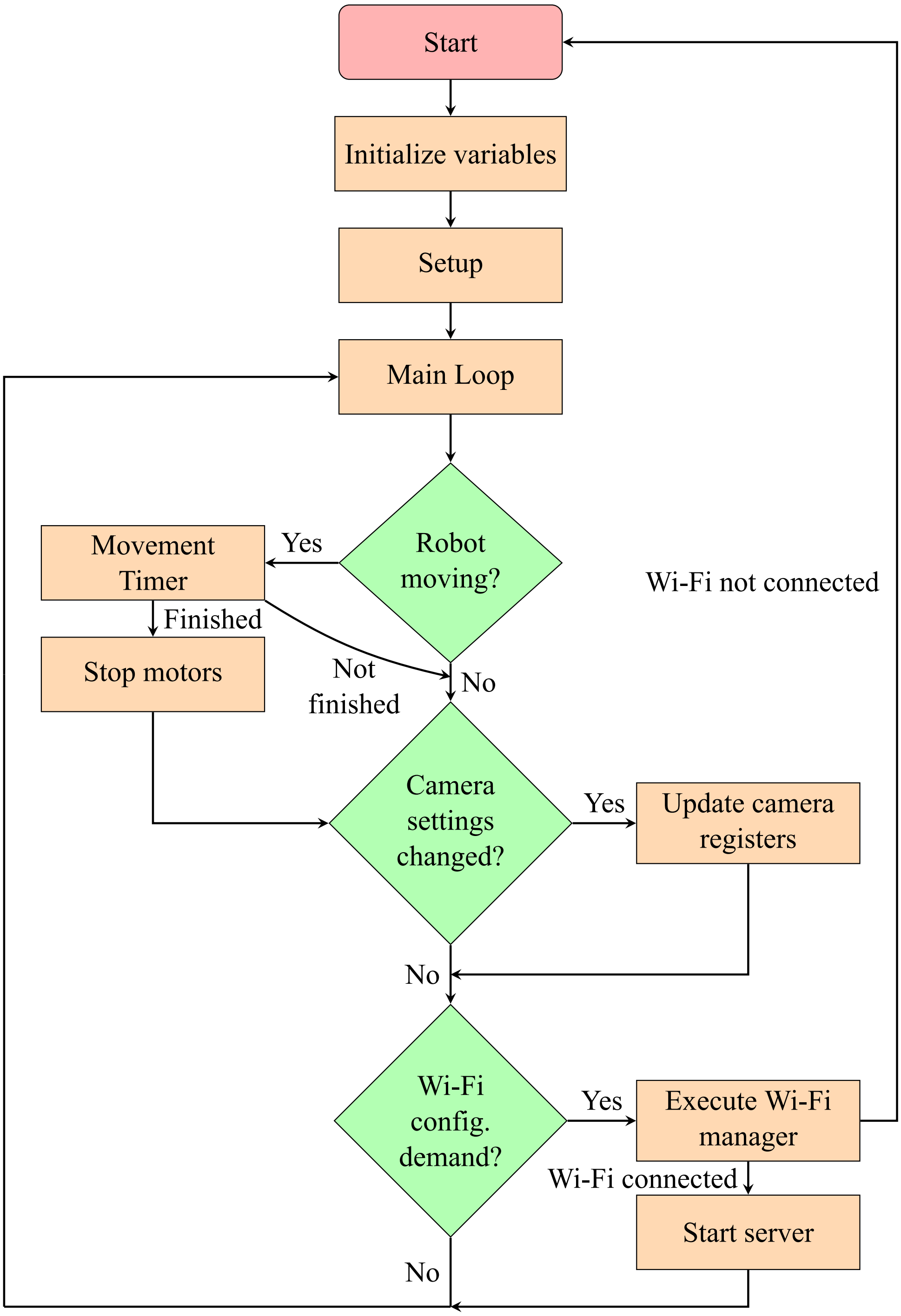


Figure 5.1: Firmware's flowchart.

- `initSPIFFS()` mounts SPIFFS,
- `cameraInit()` configures the camera chip registers,
- `initMotors()` sets the microcontroller's GPIOs and peripherals for the motors,
- `initLED()` sets GPIOs for the built-in LED,
- `initOnDemandButton()` sets **GPIO16** to input pull-up mode.

Lastly, the setup function defines the default HTTP headers for the incoming HTTP requests and outcoming responses. Header `Access-Control-Allow-Origin` indicates whether the requesting code can access the response or not. Directive `'*'` specifies a wildcard approach. Any requesting code can access the resource. [80]

Header `Access-Control-Allow-Headers` defines the headers of responses for the preflight requests. These requests indicate all possible HTTP headers of the actual requests. The most important header of the set is the `Content-Type` which defines the resource's media type. The wildcard value allows requests without credentials (without HTTP cookies or HTTP authentication information). [80]

Finally, headers `Access-Control-Allow-Methods` declare all available HTTP methods of the requests. The methods listed are:

- `GET` retrieves a representation of a specific resource,
- `POST` submits an entity to the specific resource,
- `PATCH` modifies the defined resource,
- `PUT` replaces representations of the resource with its payload,
- `DELETE` deletes a specific resource,
- `OPTIONS` elaborates the communication options of the target resource. [81]

After the initialization completes all processes successfully, the board informs the user by blinking the built-in LED seven times with a delay of 50 ms and intensity of 6, which refers to the PWM duty cycle of the corresponding timer.

Code 5.1: Robot's firmware `setup()` function.

```
void setup(){

    // Init USART communication with baud rate of 115200 - default
   baude for esp32 cam
    Serial.begin(115200);
    Serial.println();

    initSPIFFS();
    cameraInit();
    initMotors();
    initLED();
    initOnDemandButton();

    // add headers for corse policies - makes sure that OPTION
   Preflights are not ignored
    DefaultHeaders::Instance().addHeader("Access-Control-Allow-Origin
   ", "*");
    DefaultHeaders::Instance().addHeader("Access-Control-Allow-
   Headers", "Origin, Content-Type, X-Auth-Token, *");
    DefaultHeaders::Instance().addHeader("Access-Control-Allow-
   Methods" , "GET, POST, PATCH, PUT, DELETE, OPTIONS");

    // led blink after finishing setup for user indication
    ledBlink(7, 6, 50);
}
```

After the microcontroller's initialization, the program enters the infinite cycle. The cycle mainly checks three conditional states. Firstly, the loop watches the movement of the robot. If a correct flag raises, the loop starts making timestamps with *Arduino* function `millis()` [82]. If the robot moves, the loop measures the movement time and stops the motors when the movement time value exceeds a specified threshold. The thresholds are 250 ms for the robot moving forward and back and 100 ms for the robot turning left and right. If the robot does not finish moving, the cycle continues to the second step.

The second condition supervises the settings of the camera. If the microcontroller receives instructions on changing the camera registers, the loop makes the requested changes. After the program successfully applies the alternations, the loop continues to the last step of the program.

The last term controls the state of the *ON-DEMAND* button. The condition raises when **GPIO16** shorts to the ground. This exception starts the Wi-Fi manager that handles the connection to Wi-Fi and builds the configuration portal for the user to set the connection parameters. If the connection establishes successfully, the program starts the server and returns to the infinite cycle. When the manager fails, the microcontroller restarts, and the program starts from the beginning. Details about the Wi-Fi connectivity are in section 5.5.

File `main.cpp` defines the server initialization function. Code 5.2 shows the code snippet of the function. This function uses one additional open-source library called `AsyncJson.h`. Library `AsyncJson.h` enables asynchronous responses to requests with JSON payload and is a part of **ESPAsyncWebServer** [79].

Code 5.2 mainly defines the server's behaviour for requests on a specified Uniform Resource Locator (URL). One overload of function `AsyncCallbackWebHandler& AsyncWebServer::on()` takes the server's URL as the first argument, the HTTP request method as second, and the response handler as the third. The code shows the organization of the server and the handlers for the LED control, motor movement, camera register settings and control, and Favicon requests. The application sends the values for the camera registers in a JSON file. Object `AsyncCallbackJsonWebHandler* handler` defines the payload handler for the JSON files. Finally, this function initializes the board's OTA feature. Section 5.5 gives the feature's details. The server starts by calling the `void AsyncWebServer::begin()` method.

Code 5.2: Robot's firmware `startServer()` function.

```
void startServer()
{
    // preflights response
    server.on("/led-on-off", HTTP_OPTIONS, [](AsyncWebServerRequest*
   request){
    request->send(200);
    });

    // LED control
    server.on("/led-on-off", HTTP_POST, [](AsyncWebServerRequest*
   request) {}
    , NULL, ledServerHandler);

    // preflights response
    server.on("/motors", HTTP_OPTIONS, [](AsyncWebServerRequest*
   request){
    request->send(200);
    });

    //motor control
    server.on("/motors", HTTP_POST, [](AsyncWebServerRequest* request
   ) {}, NULL, motorServerHandler);

    // photo acquisition and responses
    server.on("/photo.jpg", HTTP_ANY, photoHandler);
    // default tab
    server.on("/", HTTP_ANY, [](AsyncWebServerRequest* request){
    request->send(200, "text/html", "Robot alive");
    });

    // serving favicon
    server.on("/favicon.ico", HTTP_GET, [](AsyncWebServerRequest*
   request){
    request->send(SPIFFS, "/favicon.ico", "image/x-icon");
    });

    // preflights response
    server.on("/camera-settings", HTTP_OPTIONS, [](
   AsyncWebServerRequest* request){
    request->send(200);
    });

    // handler for JSON body
    AsyncCallbackJsonWebHandler* handler = new
   AsyncCallbackJsonWebHandler("/camera-settings",
   cameraSettingsHandler);
    server.addHandler(handler);

    initOTA(&server); //start OTA
    server.begin(); // start the server
}
```

5.2 The control of the peripherals and the external hardware

Source code in file `gpio.cpp` and the corresponding header file `gpio.h` controls the external hardware and the board's peripherals. Namely, the code in these files manages the LED, the motor movement and the *ON-DEMAND* button. Moreover, the files declare and define server response handlers. This section further describes the movement and LED flashing algorithms.

5.2.1 The robot's movement algorithm

Algorithm 1 introduces moving the robot in a general direction. The motor initialization function defines the driving PWM specifications. The movement functions shift the motor switches to the desired states. The motors utilize **LEDC** [83] peripheral and its PWM generating abilities. **LEDC**'s channel 8, with a frequency of 2 kHz and 8-bit duty cycle resolution, drives the motor driver.

Algorithm 1 The robot's movement algorithm.

```
procedure MOVEROBOT(void)
    set the left motor GPIOs
    set the right motor GPIOs
    set the movement time duration
    mark a time stamp
    return movement duration and time stamp
end procedure
```

Code 5.3 shows the GPIO configuration for moving the robot forward. In this scenario, the motor driver pins **BI1** and **AI2** are pulled high, and pins **BI2** and **AI1** are low. This configuration causes the left motor to turn clockwise and the right motor to turn counterclockwise, resulting in a movement in the forward direction. The same logic applies to all movement directions.

Code 5.3: GPIO configuration for moving the robot forward.

```
digitalWrite(LEFT_M0,HIGH);
digitalWrite(LEFT_M1,LOW);
digitalWrite(RIGHT_M0,HIGH);
digitalWrite(RIGHT_M1,LOW);
```

The user controls the movement speed and direction through the control application. The application sends HTTP `POST` requests to the server. The server receives the movement directives on URL `https://<Device's IP address>/motors` and parses the valuable information. Handler `motorServerHandler(...)` decodes the incoming data and re-configures the peripherals according to Algorithm 2.

Algorithm 2 The robot's movement handler.

```
procedure MOTORSERVERHANDLER(request's data)
    receive the data
    if data's length is correct then
        parse the incoming data
        check data integrity                      ▷ Correct the data if necessary.
        set the PWM's duty cycle                  ▷ Set the movement speed.
        if movement direction flag is correct then
            call movement function 1
            raise the movement flag               ▷ For the main loop's time measurement.
            respond with HTTP code 200
            return
        else
            respond with HTTP code 500
            return
        end if
    else
        respond with HTTP code 500
        return
    end if
end procedure
```

5.2.2 LED operation

The built-in LED provides mainly two functions. The first function is signalling the state of the microcontroller to the user. The robot does not provide any user interface, therefore, determining the controller's state is difficult. The LED signals the following states to the user:

- initiated peripherals and established Wi-Fi connection - 7 flashes with an intensity of 6-bits of duty cycle resolution and 50 ms interval between the on/off states.
- The Wi-Fi interface entering the AP mode and scanning networks - 3 flashes with an intensity of 3-bits and 200 ms pause.

- The *ESP32-CAM* rebooting if the Wi-Fi manager fails - 3 flashes with an intensity of 4-bits and 600 ms interval.

The LED also uses the **LEDC** peripheral on channel 7 with a 5 kHz clocking frequency.

Furthermore, the LED provides extra light to the camera. The user turns the light on and off remotely using the server's URL `https://<Device's IP address>/led-on-off`. The server handler `ledServerHandler(...)` follows Algorithm 3.

Algorithm 3 LED's server request handler.

```
procedure LEDSERVERHANDLER(request's data)
    receive the data
    if data's length is correct then
        parse the data
        if data's values are within thresholds then
            set the LED pin state and PWM's duty cycle ▷ The duty cycle defines the light intensity.
            respond with HTTP code 200
            return
        else
            respond with HTTP code 500
            return
        end if
    else
        respond with HTTP code 500
        return
    end if
end procedure
```

5.3 Control of the camera module

Files `esp32_cameraHandler.cpp` and `esp32_cameraHandler.h` operate the camera module *OV2640* and its registers. The files comprise five essential control functions.

- The camera and board initialization - `cameraInit()`.
- The image request handler - `photoHandler(request)`.
- The camera setting request handler - `cameraSettingsHandler(request, JSON)`.
- The camera chip registers update function - `updateCameraSettings()`.
- The image capturing function - `camera_fb_t* capturePhoto()`.

The initialization function mainly sets the GPIO pins for the camera. Secondly, it disables the *brownout* detector. The *brownout* sensors signal insufficient power provided to the ESP32 board. However, this detector proved troublesome to developers. Therefore, the ESP32 framework includes two libraries, `soc/rtc_cntl_reg.h` and `soc/soc.h`, that offer a function and a macro for disabling the detector. The frequency of the `XCLK` peripheral is by default set to 20 MHz, the image's default format is JPEG, and the starting definition is SVGA, which is 800×600 pixels. The initial quality is 12 on a scale from 10 to 63, where lower means increased quality, and lastly, the camera allocates only one frame buffer for every image. Code 5.4 describes all mentioned settings.

Code 5.4: The critical part of the camera initialization.

```
// Turn-off the "brownout" detector
WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

camera_config_t config;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

config.frame_size = FRAMESIZE_SVGA;
config.jpeg_quality = 12;
config.fb_count = 1;
```

5.3.1 Camera's registers remote control

The HTTP request handler cameraSettingsHandler(request, JSON) behaves according to the algorithm shown in Code 4. The JSON body handling uses ArduinoJson.h library. The camera settings stay only in the volatile RAM.

Algorithm 4 Camera settings request handler.

```
procedure CAMERASETTINGSHANDLER(request, JSON)
    receive a JSON document
    parse the document into a data carrying structure
    raise camera settings flag                  ▷ Flag used in the main loop to update the settings.
    respond with HTTP code 200
end procedure
```

When the function cameraSettingsHandler(request, JSON) raises the appropriate flag, the main loop changes the registers settings using the updateCameraSettings() function 5.1. The updateCameraSettings() method utilizes the core libraries to acquire the settings data structure by calling sensor_t * esp_camera_sensor_get(). Code 5.5 shows how the method updateCameraSettings() assesses the registers of the camera. The variable currentSettings is a data-carrying structure mentioned in the Algorithm 4.

Code 5.5: Example of assessing the camera chip registers.

```
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, (framesize_t)currentSettings.image_resolution);
s->set_quality(s, currentSettings.video_quality);
s->set_brightness(s, currentSettings.video_brightness);
s->set_contrast(s, currentSettings.video_contrast);
s->set_saturation(s, currentSettings.video_saturation);
```

Furthermore, Code 5.6 reveals the structure of the JSON document sent by the application to the microcontroller. The key-value pairs directly correspond to the names of the camera's registers. Chapter 6 describes the sending process. All available registers and the core library documentation are under Ref. [84] and [63].

Code 5.6: Example JSON of the camera settings.

```
{
    "image_resolution":"9",      // 7 resolution choices
    "video-quality":"13",        // 10 to 63 (10 is the best)
    "video-brightness":"0",      // -2 to 2
    "video-contrast":"0",        // -2 to 2
    "video-saturation":"0",      // -2 to 2
    "image_special_effect":"0", // 0 to 6
    "awb":"1",                   // 0 = disable , 1 = enable
    "awb_gain":"1",              // 0 = disable , 1 = enable
    "wb_mode":"0",               // 0 to 4
    "aec":"1",                   // 0 = disable , 1 = enable
    "aec_dsp":"1",               // 0 = disable , 1 = enable
    "video-aelvl":"0",           // -2 to 2
    "video-aecvalue":"300",      // 0 to 1200
    "agc":"1",                   // 0 = disable , 1 = enable
    "agc-gain":"0",              // 0 to 30
    "video-gain-ceiling":"3",    // 0 to 6
    "bpc":"1",                   // 0 = disable , 1 = enable
    "wpc":"1",                   // 0 = disable , 1 = enable
    "raw_gma":"0",               // 0 = disable , 1 = enable
    "lens":"1",                  // 0 = disable , 1 = enable
    "h_mirror":"0",              // 0 = disable , 1 = enable
    "v_flip":"0",                // 0 = disable , 1 = enable
    "dwc":"0",                   // 0 = disable , 1 = enable
    "colorbar":"0"               // 0 = disable , 1 = enable
}
```

5.3.2 Image requests and acquisition

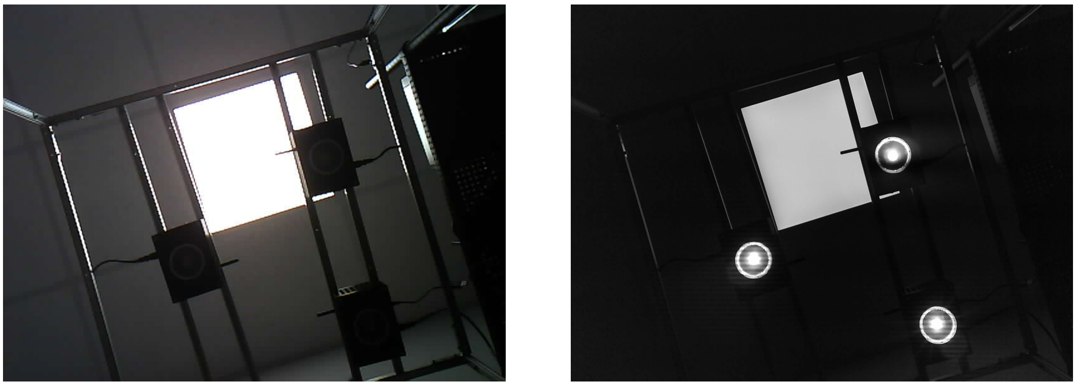
Function `capturePhoto()` defined in `esp32_cameraHandler.cpp`, uses function `camera_fb_t* esp_camera_fb_get()` from core `esp_camera.h` library to take a photo. Both the core function and the added function return the camera frame buffer. If the server receives any request on URL `https://<Device's IP address>/photo.jpg`, it responds with an image in the JPEG format. Figure 5.2 shows the photos from the robot. Figure 5.2a uses the register settings described in Code 5.6.

5.4 Flash memory's operation

The SPIFFS provides a non-volatile memory file system to the program. The program accesses the file system in the following scenarios:

- The user requests Favicon.
- The EPS32's Wi-Fi interface successfully established a connection.
- The microcontroller starts the Wi-Fi manager.

Favicon, also known as a website icon, is a file containing one or more icons associated with a particular website. The requesting side can request the Favicon and use it in its source code. Typically, browsers display Favicons in the browser's address bar. [85] [86] If the file system stores a file called `favicon.ico`



(a) Image from *OV2640*. JSON Document 5.6 shows the camera's settings.

(b) Image with tuned camera parameters for showing LED's fingerprints on the image.

Figure 5.2: Images acquired by *ESP32-CAM* and *OV2640* CMOS sensor.

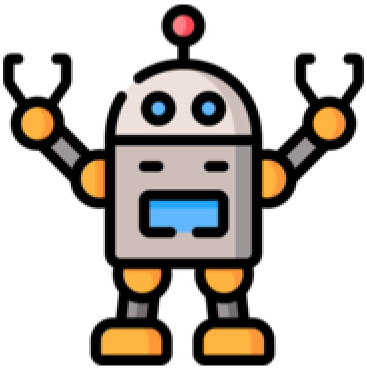


Figure 5.3: Server's Favicon. Image downloaded from [87].

in its root, the server provides the file with `{"Content-Type":"image/x-icon"}` header to the browser of other requesting applications. The designed firmware uses the Favicon illustrated in Fig. 5.3.

The Wi-Fi manager primarily calls two functions for writing and reading JSON files into and from SPIFFS. For serializing and deserializing JSON files, the program uses the open-source JSON library **ArduinoJson** [88]. Algorithm 5 digests the reading function. Algorithm 6 summarizes the writing function. The writing and reading functions are general JSON file handling methods, which can use any file stream, including SPIFFS. The described code is in files `spiffs_controller.cpp` and `spiffs_controller.h` respectively.

Algorithm 5 Reading a JSON document from a file stream.

```
procedure READJSONFILE(file stream, file path)
    open the file
    if the file does not exist then
        return NULL
    else
        read contents of the file
        deserialize the contents to a JsonObject
        close the file
        return the JsonObject
    end if
end procedure
```

Algorithm 6 Writing a JSON to a file stream.

```
procedure WRITEJSONFILE(file stream, file path, JSON document)
    open the file in the writing mode
    if the file stream is not available then
        return
    else
        serialize the JSON document to the file
        close the file
        return
    end if
end procedure
```

5.5 Wi-Fi connectivity

To properly run the server and enable the system's remote control, the robot needs a Wi-Fi Local Area Network (LAN) connection. The robot does not require access to the Internet. Therefore, a Wi-Fi router without Internet connectivity, and a device running the control application (described mainly by [14]), are all the necessary components to manage the system. Files `wifi_manager.cpp` and `wifi_manager.h` import and initialize the **ESPAsyncWiFiManager**. **ESPAsyncWiFiManager** is a third-party open-source library [89]. This library is compatible with the **ESPAsyncWebServer**. Finally, the program also imports the **AsyncElegantOTA**, a tool for upgrading firmware wirelessly. The documentation of **AsyncElegantOTA** is under Reference [90].

The files declare and define two essential functions:

- `initWifiManager()`, which sets the manager's parameters and opens the configuration portal,
- `initOTA(...)`, which enables the OTA firmware upgrades.

Algorithm 7 describes the algorithm of the first function. Pressing the *ON-DEMAND* button triggers the Wi-Fi manager. The function `initWifiManager()` adds two callbacks. The program associates one callback with the Wi-Fi interface, the callback signals when the Wi-Fi interface finishes scanning surrounding networks. The documentation of the event handler is in the Reference [91].

The other callback connects with the **ESPAsyncWiFiManager** library and triggers when the device enters the AP mode. These events blink the built-in LED and inform the user about the connection state. Other parameters of the manager are the AP name, the timeout, 180 s, a static IP, a subnet mask, a gateway IP and minimal signal quality which helps the device filter inappropriate networks. The timeout prevents the microcontroller from getting stuck in the AP mode forever and works as a security measure for the user. If the manager successfully connects to a Wi-Fi network, the static IP, subnet mask and gateway IP are saved into the flash memory using the write and read functions described in section 5.4. Otherwise, the microcontroller reboots and starts the program again.

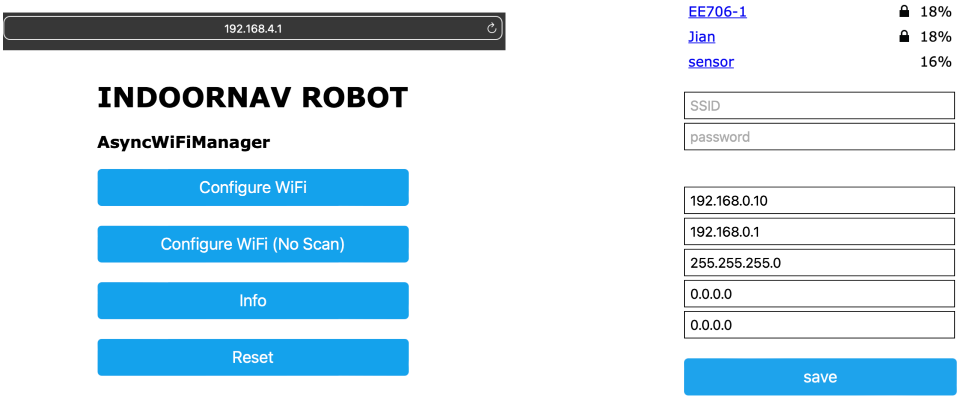
The second function initializes the OTA manager. The function takes `AsyncWebServer*` as a parameter and uses the `AsyncElegantOTA` object to begin the OTA. The OTA upgrades are only available after the Wi-Fi connection establishes and the server starts successfully. The OTA's tab is under `https://<Device's IP address>/update`.

When the user connects to the AP, the device's configuration portal is reachable. The manager itself hides under the IP address `192.168.4.1`. Figure 5.4 shows the Wi-Fi configuration portal. Figure 5.4a is the default tab of the portal, and Fig. 5.4b is the Wi-Fi selection and configuration tab.

Finally, accessing `https://<Device's IP address>/update` welcomes the user with a login tab. The web page opens the OTA firmware upgrade environment if the user enters the correct username and password. Figure 5.5 shows the OTA manager's front end.

Algorithm 7 Initialization of the Wi-Fi manager.

```
procedure INITWIFIMANAGER(server, DNS server)
    read the configuration JSON if it exists
    set the Wi-Fi mode and the AP callback
    initialize the AsyncWiFiManager
    set the manager's parameters
    open the configuration portal
    if connecting to a Wi-Fi fails then
        restart the ESP board
    else
        Save the robot's IP, the gateway's IP and the subnet mask
    end if
end procedure
```



192.168.4.1

INDOORNAV ROBOT

AsyncWiFiManager

EE706-1	🔒 18%
Jian	🔒 18%
sensor	16%

SSID

password

192.168.0.10

192.168.0.1

255.255.255.0

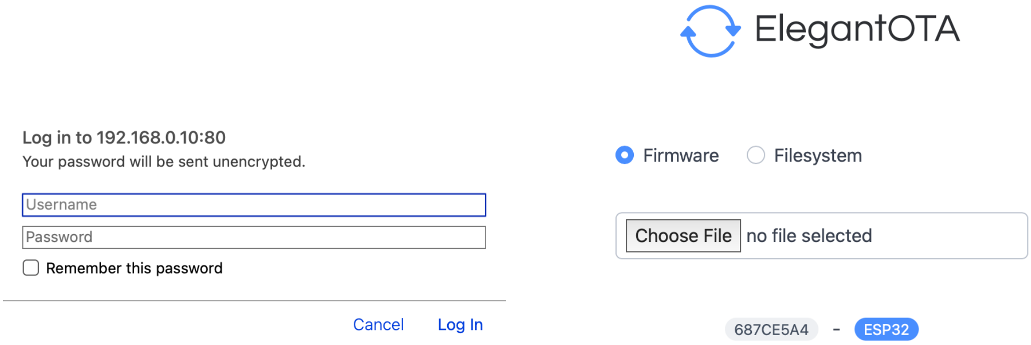
0.0.0.0

0.0.0.0

(a) The default page of the portal.

(b) A tab with Wi-Fi networks and the connection settings.

Figure 5.4: The Wi-Fi configuration portal.



Log in to 192.168.0.10:80

Your password will be sent unencrypted.

Username

Password

□ Remember this password

ElegantOTA

● Firmware ○ Filesystem

no file selected

687CE5A4 - ESP32

(a) The login pop-up form.

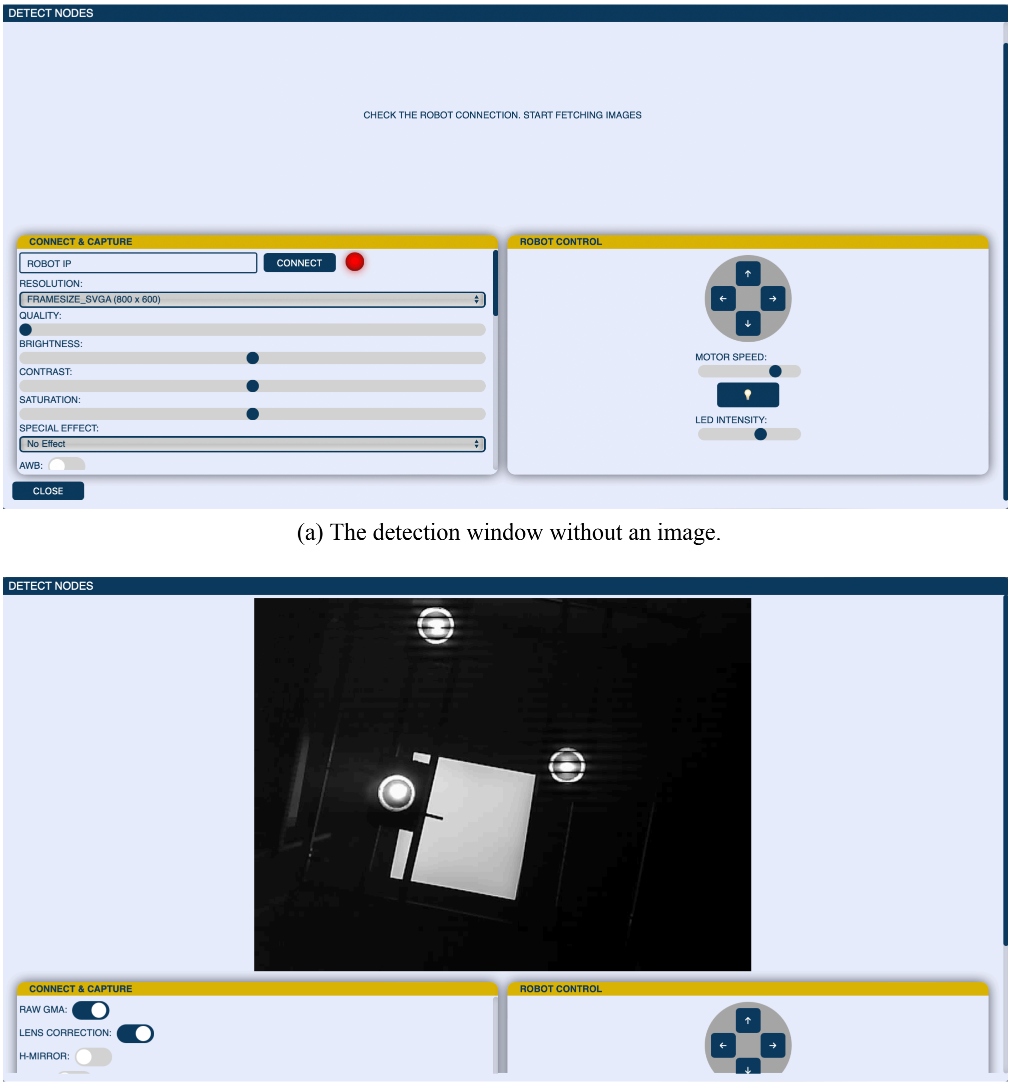
(b) The OTA's web page.

Figure 5.5: The firmware OTA upgrade portal.

Chapter 6

Robot’s control application

The control application proposed in this paper is an integrated part of the testbed system application designed by Suda, M. [14]. The contribution of this paper stands in an implementation of a tab controlling the robot. The designed User Interface (UI) enables connecting to the robot, directing the robot’s movement, switching the built-in LED, optimizing the camera’s parameters and fetching either a single image or periodical fetches. The software extension is only a front-end application consisting of *JavaScript* scripts containing *HTML* code and *CSS* styling sheets. The code’s architecture inherits the rules of Suda’s application [14]. Figure 6.1 depicts the UI’s design with and without an image. The window scales itself according to the content. The user can always access all its elements through scrollbars on the widow’s side.


DETECT NODES

CHECK THE ROBOT CONNECTION. START FETCHING IMAGES

CONNECT & CAPTURE

RESOLUTION: FRAMESIZE_SVGA (800 x 600)

QUALITY:

BRIGHTNESS:

CONTRAST:

SATURATION:

SPECIAL EFFECT: No Effect

AWB:

ROBOT CONTROL

MOTOR SPEED:

LED INTENSITY:

(a) The detection window without an image.

DETECT NODES

CONNECT & CAPTURE

RAW GMA:

LENS CORRECTION:

H-MIRROR:

ROBOT CONTROL

(b) The detection window with an image.

Figure 6.1: The robot’s control application.

The extension comprises two vital scripts and modifications to the rendering scripts and styling sheets. The extension has the following design:

- file `api.js` located in `./structure/app/jsApp/frontend/statics/scripts/robot`, manages the connection to the robot and fetches the images. This file communicates with the application's back-end.

- Script `control.js` sends the HTTP requests for moving the robot and flashing the LED. Moreover, it sends JSON files to the robot containing the camera configurations. The file is in the same folder as `api.js`.

- Script `render.js` renders the detection window. The extension added only one function called `renderDetectNodePage()`. This function wraps the widow's *HTML* code and defines the event listeners for the window's elements. Clinking the *Detect Nodes* button on the main page calls the render function, which returns all necessary *HTML* references. The file is inside `./structure/app/jsApp/frontend/statics/scripts/utils` folder.

- Lastly, sheet `app.css` modifies the window's visuals. The sheet finds itself under `./structure/app/jsApp/frontend/statics/styles`. The extension implements styles and classes related to the detection window.

6.1 Checking the robot's status

Script `api.js` checks the robot's status within the network. Asynchronous function `connectToRobot()` verifies the robot's presence. Algorithm 8 summarizes the function's algorithm. When the application successfully registers the robot, the function starts timers that periodically check the robot's presence in the network and repeatedly call the function. Therefore, the function checks for the existence of the detection window and clears the timers. The function sends a `POST` request to the back-end, opening the URL and retrieving its contents. The contents then return to the front end, which validates its integrity. The reason for communicating with the back-end is to save the robot's IP address to the database designed by [14]. The user can stop the periodic checks and disconnect the application from the robot's server by clicking the *DISCONNECT* button on the UI. The button's function resets the timers and signals the connection state to the user.

Algorithm 8 Checking the robot's presence within a network.

```
procedure CONNECTTOROBOT
    if the detection window exists then
        if user entered a valid IP address then
            send a POST request to the back-end
            if the response matches the robot's identification then
                the robot is connected
                start the timers and return
            else
                the robot is not in the network or the IP address is incorrect
                clear the timers and return
            end if
        else
            clear the timers and return
        end if
    else
        clear timers and return
    end if
end procedure
```

6.2 Controlling the robot

The robot's controls consist of three main elements: the built-in LED switching, the movement control and the camera settings control. This section further describes the algorithm of all the elements.

6.2.1 Method `flashLight`

This function operates the LED. When the user presses the lightbulb button on the UI (Figure 6.1a), the front-end sends a `POST` request to the robot containing the desired state of the LED and the light intensity. The `POST` request is only sent to the robot's server if the application verifies the robot's connection.

The button changes its state. Thus, the user uses the same button to switch the LED on and off. The `POST` request awaits a correct response from the robot. If the response does not come, the application notifies the user.

6.2.2 Method `sendMotorData`

Method `sendMotorData(buttonID)` sends `POST` requests to the server. This function controls the robot's movement direction and speed. The input parameter `buttonID` signals the type of the pressed button. Every direction has an ID, which helps the firmware differentiate between the requests. If the application registers the robot, the function sends the `POST` request containing the direction and the motor's speed. The user can change the speed through a slider on the UI. If the robot responds correctly to the request, the function ends. Otherwise, the interface informs the user about the request's failure.

6.2.3 Method `setCamera`

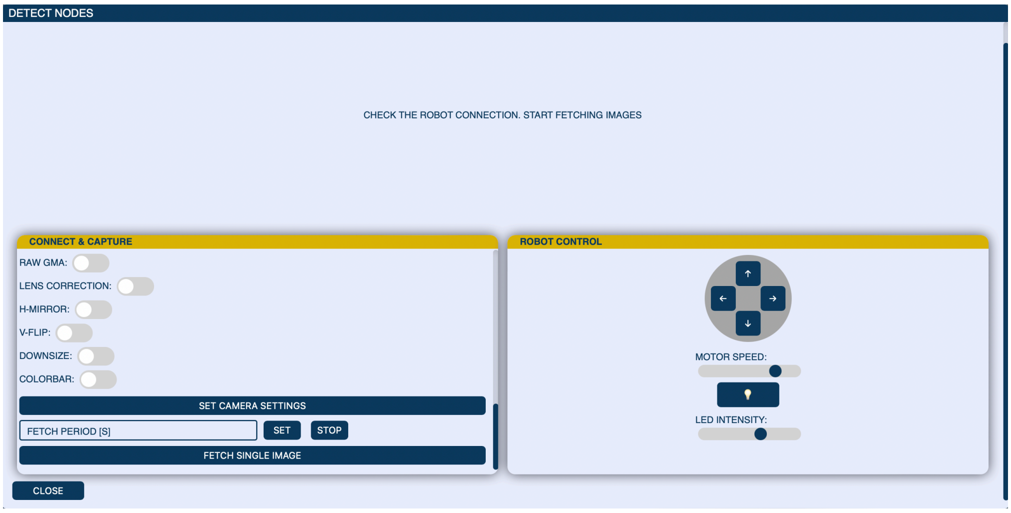
The method collects the information entered by the user and transmits a JSON file to the robot's firmware. The form for setting the parameters consists of sliders, checkboxes and drop-down menus. Figure 6.1a presents a portion of the customization window. A `for` loop collects the form's values and IDs. After the data retrieval, the function uses *JavaScript* built-in function `JSON.stringify()` [92] to build the JSON document. Afterwards, a `POST` request dispatches the document to the server if the robot replies to connection requests. If the request is unsuccessful, the application notifies the user.

6.3 Fetching the robot's images

The user can request the images in two ways. The first option is to take a single image. The second variant is periodically requesting new images from the robot. Figure 6.2 displays the UI and the fetching options.

Asynchronous function `fetchImage()` defined in `api.js` fetches a single image from the robot's server. Algorithm 9 digests the algorithm of the function. The function sends a `GET` request to the back-end containing all necessary credentials for retrieving the image. After the back-end finishes the image processing designed by [14], it returns the result to the front-end for display. If the back-end retrieves incorrect content, the user gets a notification, and the robot disconnects. The back-end returns the image as a *Base64-encoded ASCII string*, which is a compatible format with the *HTML* directive for displaying images. Code 6.1 shows the code of the image displaying function. The constant variable `img_div` refers to an *HTML* element that shows the image. Parameter `_image` represents the *Base64-encoded ASCII string*, a.k.a. the image.

Finally, function `setPeriod()` executes Algorithm 9 in periodic time intervals set by the user. The period must not be lower than 0.01 s, which proved to be this request-style architecture's limit. The user can stop the image requests by pressing the *STOP* button (see Fig. 6.2). The *STOP* button resets the timer that executes `fetchImage()`.



CHECK THE ROBOT CONNECTION. START FETCHING IMAGES

Figure 6.2: The image fetching options of the detection window.

Algorithm 9 Retrieving images from the robot's server.

```
procedure FETCHIMAGE
    if the robot is connected to the application then
        send a GET request to the back-end for the image
        await unit the back-end finishes image processing procedures and returns the processed image
        if the robot answers correctly to the request then
            display the image and return
        else
            notify user about the request's failure and return
        end if
    else
        inform the user and return
    end if
end procedure
```

Code 6.1: The image displaying function.

```javascript
function displayImage(_image) {
    const img_div = document.getElementById("node_detect_img");
    img_div.innerHTML = `<img src="data:image/jpeg;base64,${_image}"
   />`;
}
```

Chapter 7

Conclusion and future work

The contribution of this paper is three-fold. Firstly, the paper describes the previously designed light source issues and suggests one possible solution. The newly invented Visible Light Communication (VLC) node adds a compensation network that solves the stability issues of the previous prototype. The essential semiconductor components of the light source are a power NMOS transistor and an Operational Amplifier (OPA). This paper simulates and tests different variants of these components. The results show that changing the OPA increases the node's slew rate and suppresses the imperfections of the previous design. In this case OPA *AD817* [19] replaces amplifier *UA741CD* [17]. Transistor *IRF540NPbF* [20] replaces *IRF710* [18], decreasing the transistor's power dissipation. Therefore, the total temperature decreases, allowing the transistor to function without additional heat sink. However, the newly utilized transistor introduces lower dynamic response to its predecessor, resulting in an undesirable behaviour during the Voltage-to-Current conversion.

Moreover, the new prototype adds a noise suppressing circuit to the OPA's power supply. This attenuation block filters all disruptive signals except for the DC component. Filtering noises from the supply voltage is vital for the amplifier.

Secondly, this thesis categorizes the designed VLC node in a taxonomy proposed by the latest VLP reviews [40] [7]. The paper summarizes all known and available positioning algorithms based on the categorization. Furthermore, following the system's theoretical evaluation, the research introduces the most commonly used VLC receivers and selects a particular one - the *ESP32-CAM* [11] development board with *OV2640* [63] CMOS sensor.

Lastly, the main goal of the thesis is to construct a mobile platform for testing various positioning algorithms. The robot comprises three key components:

1. the hardware and mechanical realization of the robot;

2. the *ESP32-CAM*'s embedded software;

3. the robot's control application and User Interface (UI).

The mobile robot's hardware comprises multiple electronic blocks such as the robot's power supply branch, programmer, or motor control branch. The paper also compares the integrated and external antenna of the evaluation board with a specific RSS Wi-Fi measurement. The results show that the external antenna performs better on more considerable distances but brings a minor improvement for close-distance communications.

The robot's firmware defines the robot's behaviour and enables all critical functions for VLP. The software secures features such as the robot's movement, torch control, Serial Peripheral Interface Flash File System (SPIFFS) writing and reading, Over-The-Air (OTA) firmware upgrades and on-demand Wi-Fi configuration. Most importantly, the software obtains photos above the robot and makes them accessible on its server through HyperText Transfer Protocol (HTTP). Moreover, all the named functions and the camera settings control are accessible via HTTP. Therefore, the mobile platform is fully operational without a wired connection.

Finally, based on parallel research presented by Suda, M. [14], the paper proposes a software extension to the system control application. This extension connects to the robot's server, downloads the images and controls the robot through HTTP requests. This thesis shows images of the UI and images captured by the robot placed in the testing environment.

This research and the co-operate research [14] lay a solid background for testing various positioning and indoor navigation algorithms. Using Suda's edge-detection software and implementing proximity positioning algorithms should be the next steps of the research. Additionally, adding various sensors to the robot's apparatus, such as an accelerometer, gyroscope, geomagnetic sensor or a photodiode, would increase its versatility.

Appendix A

References

[1] D. Dardari, P. Closas and P. M. Djurić, 'Indoor tracking: Theory, methods, and technologies,' *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015. doi: 10.1109/TVT.2015.2403868.

[2] R. Joseph and S. B. Sasi, 'Indoor positioning using wifi fingerprint,' in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 2018, pp. 1–3. doi: 10.1109/ICCSDET.2018.8821184.

[3] S. Sophia, B. M. Shankar, K. Akshya, A. C. Arunachalam, V. T. Y. Avanthika and S. Deepak, 'Bluetooth low energy based indoor positioning system using esp32,' in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2021, pp. 1698–1702. doi: 10.1109/ICIRCA51532.2021.9544975.

[4] A. Vena, I. Illanes, L. Alidieres, B. Sorli and F. Perea, 'Rfid based indoor localization system to analyze visitor behavior in a museum,' in *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, 2021, pp. 183–186. doi: 10.1109/RFID-TA53372.2021.9617265.

[5] A. Aldalbahi, M. Rahaim, A. Khreishah, M. Ayyash and T. Little, 'Visible light communication module: An open source extension to the ns3 network simulator with real system validation two scenarios are used in the validation of the vlc module:' *IEEE Access*, vol. PP, pp. 1–1, Oct. 2017. doi: 10.1109/ACCESS.2017.2759779.

[6] S. Afifah, S. Angelina, P. Susamti, R. Priramadhi and D. Darlis, 'Energy efficient transmitter for guided indoor navigation using visible light,' *IOP Conference Series: Materials Science and Engineering*, vol. 434, p. 012 213, Dec. 2018. doi: 10.1088/1757-899X/434/1/012213.

[7] Y. Zhuang, L. Hua, L. Qi *et al.*, 'A survey of positioning systems using visible led lights,' *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1963–1988, 2018. doi: 10.1109/COMST.2018.2806558.

[8] Š. Bosák, 'Hw for indoor visible light positioning testbed,' Bachelor's Thesis, CTU FEE, Technická 2, May 2020. [Online]. Available: http://hdl.handle.net/10467/87622.

[9] M. Suda, 'Sw for indoor visible light positioning testbed,' Bachelor's Thesis, CTU FEE, Technická 2, May 2020. [Online]. Available: http://hdl.handle.net/10467/87626.

[10] S. Li, Z. Wang, L. Shi and X. Zhang, 'Experimental indoor tracking testbed based on visible light communication,' in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 129–130. doi: 10.1109/ICECS46596.2019.8965061.

[11] Ai-Thinker, *ESP32-CAM*, 2017. [Online]. Available: http://www.ai-thinker.com/pro_view-24.html.

[12] *Device classes*. [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/classes/.

[13] STMicroelectronics, *I-NUCLEO-LRWAN1*, Original product from USI in partnership with STMicroelectronics., 2018. [Online]. Available: https://www.st.com/resource/en/data_brief/i-nucleo-lrwan1.pdf.

[14] M. Suda, 'Infrastructure of indoor vlp system,' Master's Thesis, NTUST DECE, Taipei, Taiwan (R.O.C.), Jun. 2022.

[15] STMicroelectronics, *B-L072Z-LRWAN1*, The B-L072Z-LRWAN1 LoRa/Sigfox Discovery kit is a development tool to learn and develop solutions based on LoRa, Sigfox., 2019. [Online]. Available: https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html.

[16] A. Devices, *ADG819BRTZ-500RL7*, Original product from Analog Devices., 2012. [Online]. Available: https://cz.mouser.com/datasheet/2/609/ADG819-1503098.pdf.

[17] T. Instruments, *UA741CD*, Original product from Texas Instruments., 2018. [Online]. Available: https://www.gme.cz/data/attachments/dsh.310-515.1.pdf.

[18] VISHAY, *IRF710*, Original product from VISHAY., 2011. [Online]. Available: https://www.gme.cz/data/attachments/dsh.213-180.1.pdf.

[19] *AD817*. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD817.pdf.

[20] I. Rectifier, *IRF540NPBF*, Original product from International Rectifier., 2011. [Online]. Available: https://www.infineon.com/dgdl/irf540npbf.pdf?fileId=5546d462533600a4015355e39f0d19a1.

[21] *Parasitic oscillation and ringing of power mosfets*, Jul. 2018. [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjWiY6SmPL1AhUQjVYBHesnDUAQFnoECA4QAw&url=https%5C%3A%5C%2F%5C%2Ftoshiba.semicon-storage.com%5C%2Finfo%5C%2Fdocget.jsp%5C%3Fdid%5C%3D59456&usg=AOvVaw1QFazvSVKwhRVd6-XnKqUp.

[22] *Ltspice: Stability of op amp circuits*. [Online]. Available: https://www.analog.com/en/education/education-library/videos/5579254320001.html.

[23] O. O. S. GmbH., *LE UW U1A3 01*, Original document from OSRAM Opto Semiconductors., 2018. [Online]. Available: https://dammedia.osram.info/media/resource/hires/osram-dam-6715835/LE%5C%20UW%5C%20U1A3%5C%2001_EN.pdf.

[24] *Ti precision labs-op amps: Stability-phase margin*, Oct. 2021. [Online]. Available: https://training.ti.com/ti-precision-labs-op-amps-stability-phase-margin?context=1139747-1139745-14685-1138805-13849.

[25] C. Xie, W. Guan, Y. Wu, L. Fang and Y. Cai, 'The led-id detection and recognition method based on visible light positioning using proximity method,' *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–16, 2018. doi: 10.1109/JPHOT.2018.2809731.

[26] Rohde and Schwarz, *ROHDE AND SCHWARZ RTM3000 OSCILLOSCOPE*, 2022. [Online]. Available: https://www.rohde-schwarz.com/ae/products/test-and-measurement/oscilloscopes/rs-rtm3000-oscilloscope_63493-427459.html.

[27] R. Semiconductor, *Thermal calculation for linear regulator*, 2017. [Online]. Available: https://fscdn.rohm.com/en/products/databook/applinote/ic/power/linear_regulator/linearreg_heat_calculation_appli-e.pdf.

[28] T. Instruments, *LM340T-12/NOPB*, Original product from Texas Instruments., 2020. [Online]. Available: http://www.ti.com/general/docs/suppproductinfo.tsp?distId=26&gotoUrl=http%5C%3A%5C%2F%5C%2Fwww.ti.com%5C%2Flit%5C%2Fgpn%5C%2Flm340.

[29] M. WELL, *RPS-30-12*, 2021. [Online]. Available: https://www.meanwell-web.com/en-gb/ac-dc-single-output-medical-open-frame-power-rps--30--12.

[30] X. Power, *Iz series*, 2021. [Online]. Available: https://www.xppower.com/product/IZ-Series#.

[31] M. P. Solutions, *Dc-dc converters*, 2020. [Online]. Available: https://www.murata.com/-/media/webrenewal/products/power/appnote/dcan-68.ashx?la=ja-jp&amp;cvid=20201014090810000000.

[32] *BLM21AG102SN1#*, 2022. [Online]. Available: https://www.murata.com/en-global/products/productdetail?partno=BLM21AG102SN1%5C%23.

[33] B. Reeve, *The capacitance multiplier*, Feb. 2021. [Online]. Available: https://audioxpress.com/article/the-capacitance-multiplier.

[34] D. Mercer, *16.3.1 transistor based capacitance multiplier*, 2021. [Online]. Available: `https://wiki.analog.com/university/courses/electronics/text/chapter-16`.

[35] *BCX54-16*, 2017. [Online]. Available: `https://www.nexperia.com/products/bipolar-transistors/general-purpose-and-low-vcesat-bipolar-transistors/single-bipolar-transistors/single-bipolar-transistors-100-v/BCX54-16.html`.

[36] *BCX51/52/53*, Aug. 2021. [Online]. Available: `https://www.diodes.com/assets/Datasheets/BCX51_52_53.pdf`.

[37] Littlefuse, *TPSMB150A-VR-TPSMB-VR series*, Dec. 2020. [Online]. Available: `https://www.littelfuse.com/products/tvs-diodes/automotive-and-high-reliability-tvs/tpsmb_vr/tpsmb150a_vr.aspx`.

[38] T. F. LLC, *FLIR E75*, The product has been discontinued. Recommended replacement: FLIR E76., 2021. [Online]. Available: `https://www.flir.com/support/products/e75/#Overview`.

[39] W. Thermal, *Wakefield thermal 40x10mm 2 wire dc fan 12.2cfm —DC0401012V2B-2T0*, 40mm x 40mm x 10mm 2 Wire DC Fan 12.2CFM. [Online]. Available: `https://wakefieldthermal.com/wakefield-thermal-40x10mm-2-wire-dc-fan-12-2cfm-dc0401012v2b-2t0/`.

[40] P. Chen, M. Pang, D. Che, Y. Yin, D. Hu and S. Gao, 'A survey on visible light positioning from software algorithms to hardware,' *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–20, Feb. 2021. doi: `10.1155/2021/9739577`.

[41] M. Nakajima and S. Haruyama, 'New indoor navigation system for visually impaired people using visible light communication,' *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, Dec. 2013. doi: `10.1186/1687-1499-2013-37`.

[42] C. Sertthin, T. Fujii, O. Takyu, Y. Umeda and T. Ohtsuki, 'On physical layer simulation model for 6-axis sensor assisted vlc based positioning system,' in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–5. doi: `10.1109/GLOCOM.2011.6134119`.

[43] Y. U. Lee and M. Kavehrad, 'Two hybrid positioning system design techniques with lighting leds and ad-hoc wireless network,' *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1176–1184, 2012. doi: `10.1109/TCE.2012.6414983`.

[44] S.-Y. Jung, S. Hann, S. Park and C.-S. Park, 'Optical wireless indoor positioning system using light emitting diode ceiling lights,' *Microwave and Optical Technology Letters*, vol. 54, Jul. 2012. doi: `10.1002/mop.26880`.

[45] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao and P. Dutta, 'Luxapose: Indoor positioning with mobile phones and visible light,' in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14, Maui, Hawaii, USA: Association for Computing Machinery, 2014, pp. 447–458, isbn: 9781450327831. doi: `10.1145/2639108.2639109`. [Online]. Available: `https://doi.org/10.1145/2639108.2639109`.

[46] Z. Yang, Z. Wang, J. Zhang, C. Huang and Q. Zhang, 'Wearables can afford: Light-weight indoor positioning with visible light,' *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015.

[47] S. Yamaguchi, V. V. Mai, T. C. Thang and A. T. Pham, 'Design and performance evaluation of vlc indoor positioning system using optical orthogonal codes,' in *2014 IEEE Fifth International Conference on Communications and Electronics (ICCE)*, 2014, pp. 54–59. doi: `10.1109/CCE.2014.6916679`.

[48] P. Hu, L. Li, C. Peng, G. Shen and F. Zhao, 'Pharos: Enable physical analytics through visible light based indoor localization,' Nov. 2013. doi: `10.1145/2535771.2535790`.

[49] L. Li, P. Hu, C. Peng, G. Shen and F. Zhao, 'Epsilon: A visible light based positioning system,' *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pp. 331–343, Jan. 2014.

[50] T. Q. Wang, Y. A. Sekercioglu, A. Neild and J. Armstrong, 'Position accuracy of time-of-arrival based ranging using visible light with application in indoor localization systems,' *Journal of Lightwave Technology*, vol. 31, no. 20, pp. 3302–3308, 2013. doi: 10.1109/JLT.2013.2281592.

[51] Y. H. Choi, I. H. Park, Y. H. Kim and J. Y. Kim, 'Novel lbs technique based on visible light communications,' in *2012 IEEE International Conference on Consumer Electronics (ICCE)*, 2012, pp. 576–577. doi: 10.1109/ICCE.2012.6161980.

[52] T.-H. Do and M. Yoo, 'Tdoa-based indoor positioning using visible light,' *Photonic Network Communications*, vol. 27, pp. 80–88, Apr. 2014. doi: 10.1007/s11107-014-0428-4.

[53] A. Arafa, S. Dalmiya, R. Klukas and J. Holzman, 'Angle-of-arrival reception for optical wireless location technology,' *Optics Express*, vol. 23, Mar. 2015. doi: 10.1364/OE.23.007755.

[54] M. Yasir, S.-W. Ho and B. N. Vellambi, 'Indoor positioning system using visible light and accelerometer,' *Journal of Lightwave Technology*, vol. 32, no. 19, pp. 3306–3316, 2014. doi: 10.1109/JLT.2014.2344772.

[55] M. Yasir, S.-W. Ho and B. N. Vellambi, 'Indoor position tracking using multiple optical receivers,' *Journal of Lightwave Technology*, vol. 34, no. 4, pp. 1166–1176, 2016. doi: 10.1109/JLT.2015.2507182.

[56] G. B. Prince and T. D. Little, 'A two phase hybrid rss/ aoa algorithm for indoor device localization using visible light,' in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 3347–3352. doi: 10.1109/GLOCOM.2012.6503631.

[57] K. N. e. a. Hirokazu Nishikata Hideo Makino, 'Basic research of indoor positioning method using visible light communication and dead reckoning,' *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2011.

[58] N.-T. Nguyen, N.-H. Nguyen, V.-H. Nguyen, K. Sripimanwat and A. Suebsomran, 'Improvement of the vlc localization method using the extended kalman filter,' in *TENCON 2014 - 2014 IEEE Region 10 Conference*, 2014, pp. 1–6. doi: 10.1109/TENCON.2014.7022416.

[59] E. O. Inc., *Imaging electronics 101: Understanding camera sensors for machine vision applications*. [Online]. Available: https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/.

[60] W. Guan, Y. Wu, C. Xie, L. Fang, X. Liu and Y. Chen, 'Performance analysis and enhancement for visible light communication using cmos sensors,' *Optics Communications*, vol. 410, pp. 531–551, 2018, issn: 0030-4018. doi: https://doi.org/10.1016/j.optcom.2017.10.038. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401817309458.

[61] W. Guan, X. Chen, M. Huang, Z. Liu, Y. Wu and Y. Chen, 'High-speed robust dynamic positioning and tracking method based on visual visible light communication using optical flow detection and bayesian forecast,' *IEEE Photonics Journal*, vol. 10, no. 3, pp. 1–22, 2018. doi: 10.1109/JPHOT.2018.2841979.

[62] O. Technologies, *Freertos real-time operating system for microcontrollers*. [Online]. Available: https://www.freertos.org.

[63] O. Technologies, *Ov2460 color cmos uxga (2.0 megapixel) camerachip with omnipixel2 technology*, Feb. 2006. [Online]. Available: https://www.uctronics.com/download/cam_module/OV2640DS.pdf.

[64] O. Technologies, *Ov7670/ov7171 cmos vga (640x480) camerachip with omnipixel technology*, May 2006. [Online]. Available: https://www.uctronics.com/download/Datasheet/ov7670_full_datasheet.pdf.

[65] LaskaKit, *Převodník 6pin usb ttl uart, cp2102, dtr pin*. [Online]. Available: https://www.laskakit.cz/prevodnik-6pin-usb-ttl-uart--cp2102--dtr-pin/.

[66] S. Studio, *Image of ESP32-CAM*. [Online]. Available: https://www.seeedstudio.com/media/catalog/product/cache/ef3164306500b1080e8560b2e8b5cc0f/b/a/bazaar1003541_esp32cam3.jpg.

[67] D. Kim, J. K. Park and J. T. Kim, 'Three-dimensional vlc positioning system model and method considering receiver tilt,' *IEEE Access*, vol. 7, pp. 132 205–132 216, 2019. doi: 10.1109/ACCESS.2019.2940759.

[68] Y. Li, Z. Ghassemlooy, X. Tang, B. Lin and Y. Zhang, 'A vlc smartphone camera based indoor positioning system,' *IEEE Photonics Technology Letters*, vol. 30, no. 13, pp. 1171–1174, 2018. doi: 10.1109/LPT.2018.2834930.

[69] D. Workshop, *Build an esp32cam robot car*, Apr. 2021. [Online]. Available: https://dronebotworkshop.com/esp32cam-robot-car/.

[70] D. Inc., *H-bridges*, 2014. [Online]. Available: https://learn.digilentinc.com/Documents/325.

[71] B. J. Jim Brown, *Brief h-bridge theory of operation*, Sep. 2002. [Online]. Available: https://archive.ph/20130112153214/http://www.dprg.org/tutorials/1998-04a/#selection-267.0-267.34.

[72] Toshiba, *TB6612FNG*, Jun. 2007. [Online]. Available: https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf.

[73] MagiDeal, *Magideal 1 piece dc geared motor, 3-6v tt gear motor, motor support/bracket for arduino diy robot smart car robot tool chassis (multicolor)*. [Online]. Available: https://www.flipkart.com/magideal-1-piece-dc-geared-motor-3-6v-tt-gear-motor-support-bracket-arduino-diy-robot-smart-car-tool-chassis/p/itmf3h45ybkzbjzw.

[74] T. Instruments, *Lm2596 simple switcher power converter 150-khz 3-a step-down voltage regulator*, May 2021. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm2596.pdf?ts=1649206965309&ref_url=https%253A%252F%252Fwww.google.com%252F.

[75] NooElec, *10 pcs lm2596 dc-dc buck converter step down module power supply output 1.23v-30v*, Jun. 2012. [Online]. Available: https://www.amazon.com/LM2596-Converter-Module-Supply-1-23V-30V/dp/B008BHBEE0.

[76] diymore Store, *Diymore esp32 cam ov2640 2mp w-bt camera module+2.4g extension cable 20cm ipx adapter cable*, Jun. 2019. [Online]. Available: https://www.amazon.com/dp/B07T1PG77D?tag=makeradvisor-20&linkCode=ogi&th=1&psc=1.

[77] TOTOLINK, *Ac1200, wireless dual band router, model no. a800r*. [Online]. Available: https://www.totolink.net/home/menu/detail/menu_listtpl/products/id/166/ids/33.html.

[78] T. Chip, *2wd smart robot car chassis kit*. [Online]. Available: https://www.twinschip.com/2WD_Smart_Robot_Chassis.

[79] *Async web server for esp8266 and esp32*, Mar. 2022. [Online]. Available: https://github.com/me-no-dev/ESPAsyncWebServer.

[80] M. W. Docs, *Http headers*, Apr. 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers.

[81] M. W. Docs, *Http request methods*, Oct. 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods.

[82] Ardunio, *Millis()*, Mar. 2021. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/time/millis/.

[83] L. Espressif Systems (Shanghai) Co., *Led control (ledc)*, 2022. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/ledc.html#ledc-api-supported-range-frequency-duty-resolution.

[84] *Esp32 camera driver*, Licensed under the Apache License 2.0, Feb. 2022. [Online]. Available: https://github.com/espressif/esp32-camera.

[85] D. Lane, *Creating a multi-resolution favicon including transparency with the gimp*, Aug. 2011. [Online]. Available: https://web.archive.org/web/20101225130900/http://egressive.com/tutorial/creating-a-multi-resolution-favicon-including-transparency-with-the-gimp.

[86] B. Jan, *Express.js tutorial*, Oct. 2021. [Online]. Available: https://zetcode.com/javascript/expressjs/.

[87] Freepik, *Robot free icon*. [Online]. Available: https://www.flaticon.com/free-icon/robot_3662817?related_id=3662817&origin=search.

[88] B. Blanchon, *Arduino json*, 2022. [Online]. Available: https://arduinojson.org.

[89] tzapu, *AsyncWiFiManager*, Licensed under the MIT License., Jan. 2022. [Online]. Available: https://github.com/alanswx/ESPAsyncWiFiManager.

[90] A. Sharma, *AsyncElegantOTA*, Licensed under the MIT License., Apr. 2022. [Online]. Available: https://github.com/ayushsharma82/AsyncElegantOTA.

[91] I. Grokhotkov, *Generic class*, Licensed under the GNU Lesser General Public License v2.1, 2017. [Online]. Available: https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/generic-class.html.

[92] MDN, *Json.stringify()*, Licensed under a Creative Commons license, Mar. 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify.

Attachments

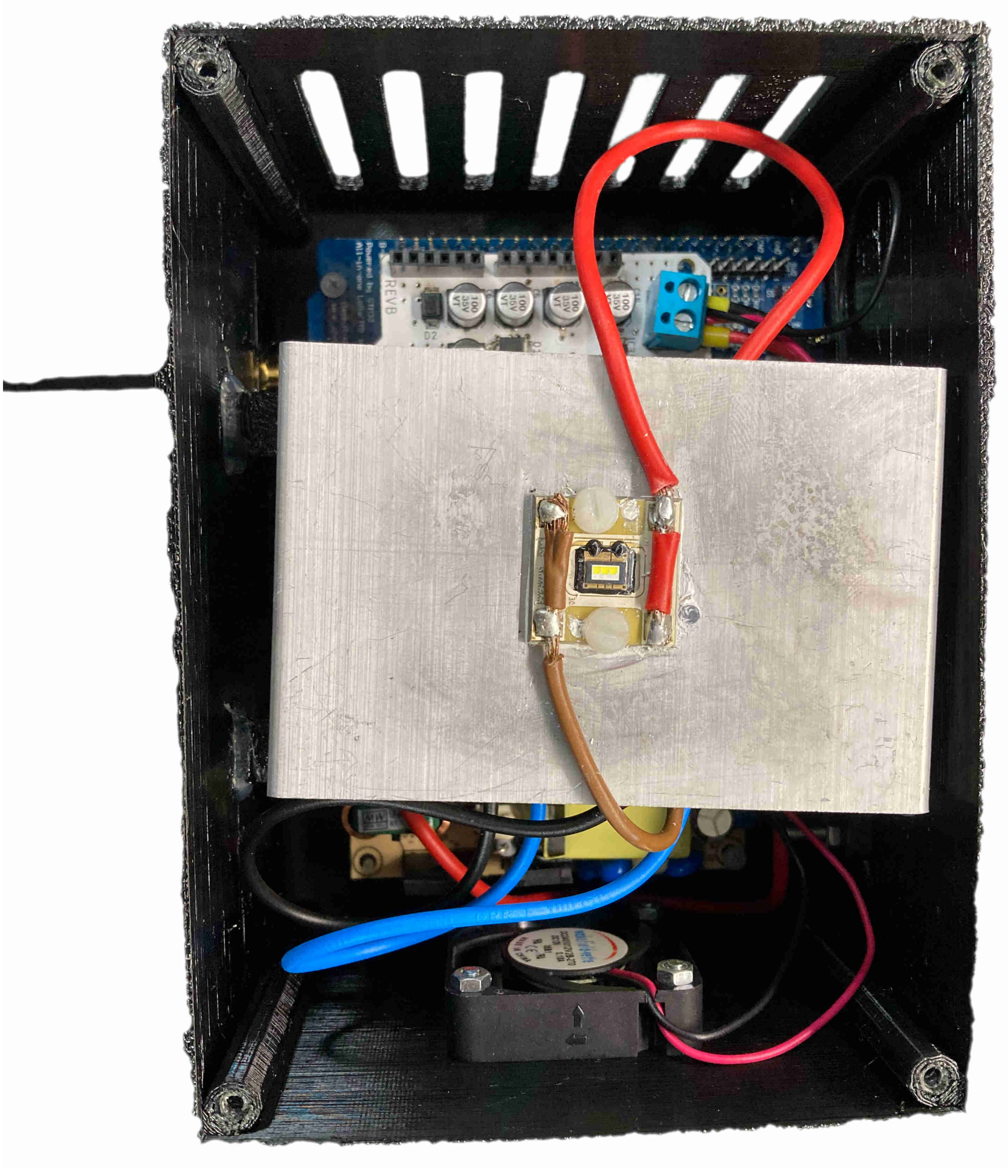


Figure B.1: VLC node realization - The node's case with the LED and the heat sink.

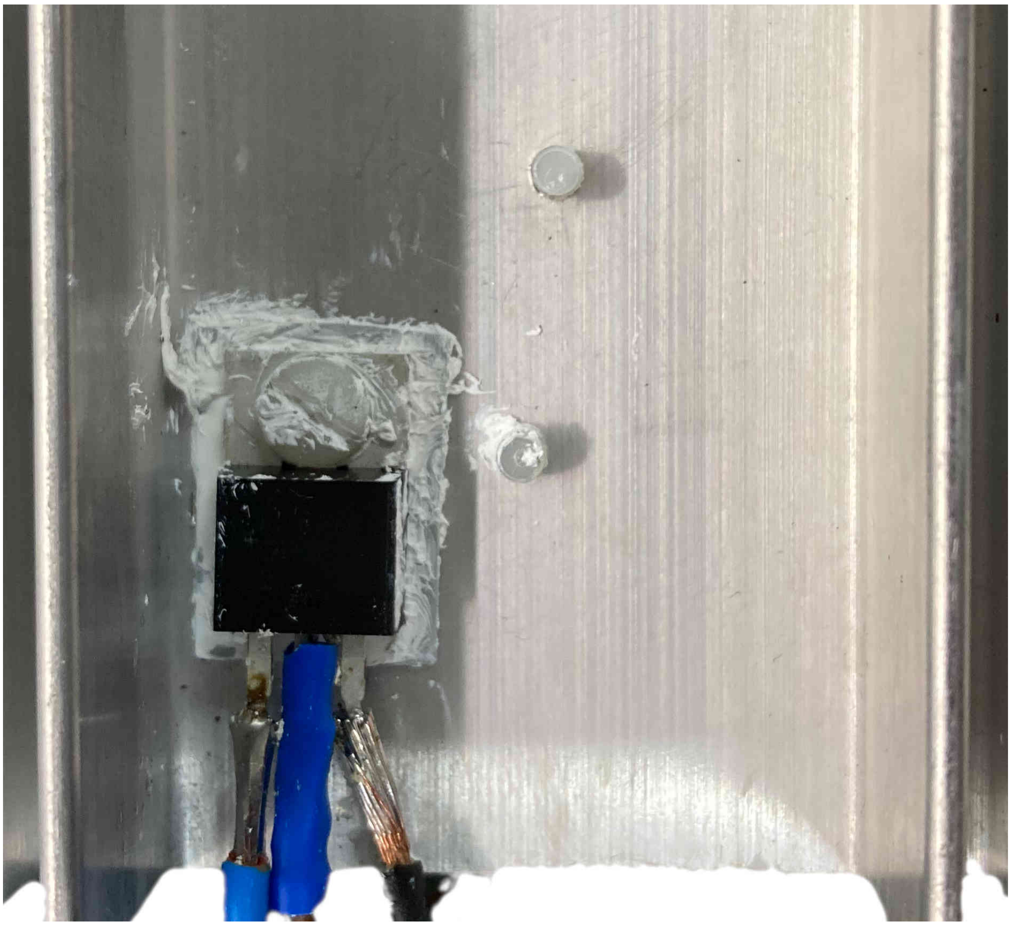


Figure B.2: Installation of the power NMOS transistor onto the heat sink's bottom side.

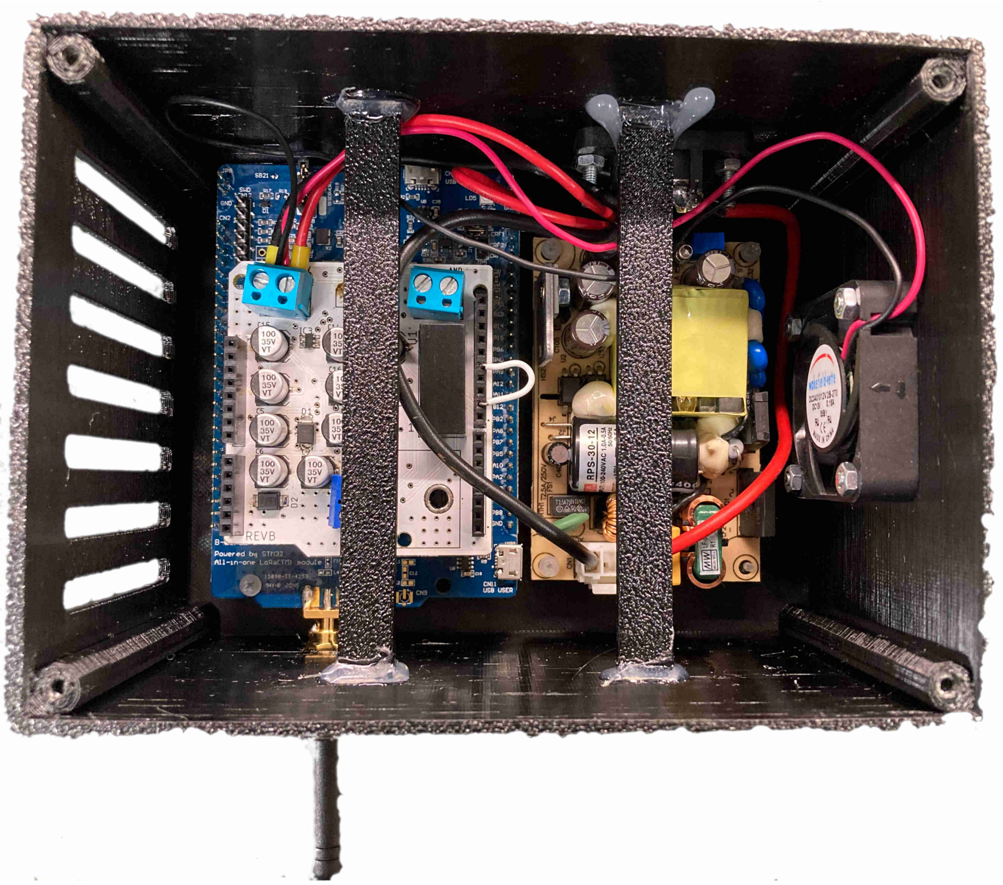


Figure B.3: VLC node realization - the node’s case with the *NUCLEO* board and the AC/DC supply.