People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University Yahia Fares of Medea

Faculty of Technology

Department of Electrical Engineering



Master thesis

Submitted for the Master's degree in Electrical Engineering

Option : Automation and Systems

Thesis subject

Vision-Based Control of a Mobile Robot: A Real-Time Approach

Realised by:

REBOUH Aymene

DALI AHMED Rafik

Supervised by:

Dr. MENASRI Wahiba

Promotion 2024/2025

Acknowledgments

First of all, we thank Allah, the Almighty, for granting us the strength, patience, and determination to successfully complete this modest work.

We express our deepest gratitude to Dre. Menasri Wahiba for her kind acceptance to supervise this thesis. Her guidance, availability, insightful remarks, and unwavering support were instrumental throughout every stage of this project. Her expertise and constructive feedback helped us overcome challenges and improved the quality of our work significantly.

We would also like to extend our heartfelt thanks to everyone who contributed whether directly or indirectly to the realization of this project. Your encouragement, advice, and assistance have been greatly appreciated.

Thank you.

Dedication

Before thinking about "dedications," I want to dedicate this humble work to the person who

has always been, and will always remain, a source of inspiration and strength:

To my dear father Ms Mustapha

To my mother Benabdellah Latifa

To my brothers Moncef et Heithem.

To my sister Ines.

To my Family

To the entire Class of 2025 Automation and Systems, with special thoughts for my fellow project partner.

To my friends and everyone who believed in me, guided me, or simply stood by me during this long and challenging road.

Aymen

Dedication

Before thinking about "dedications," I want to dedicate this modest work to the one who has always been a role model of strength, kindness, and wisdom:

To my dear father Mr. Mahieddine

To my motherMs. Amina Moulai Mustpha

To my brother Watik

To my sisters Meroua and Sirine.

To my family

To my partner Aymen, to my close friends and colleagues who stood by me throughout this academic journey.

To the Class of 2025 Automation and Systems, especially those who shared this final year with me. Who supported me and stood with me during this whole challenging journey. Thanks for the great memories. And I wish you all the best.

Rafik

ملخص

يتناول هذا المشروع تصميم وتنفيذ روبوت متنقل ذاتي الحركة يعتمد على الرؤية الحاسوبية لتتبع كائن ملون (مثال: كرة خضراء) في الزمن الحقيقي. يستخدم النظام كاميرا ESP32-CAM لالتقاط الفيديو ومعالج ESP32 للتحكم في حركة الروبوت بناءً على بيانات يتم إرسالها عبر شبكة الواي فاي من حاسوب يشغّل خوارزمية تتبع الكائن. كان من المخطط استخدام وحدة GPS لتحديد الموقع بدقة، لكن تم الاستغناء عنها بسبب مشاكل في الاتصال. يوضح هذا العمل كيف يمكن باستخدام مكونات منخفضة التكلفة تحقيق سلوك مستقل للروبوت في بيئة شبه منظمة باستخدام تقنيات بسيطة للرؤية والتحكم.

الكلمات المفتاحية : روبوت متنقل،تتبع أجسام، الملاحة، رؤية، الكاميرا ESP32.

Abstract

This thesis presents the design and implementation of an autonomous mobile robot that uses computer vision to detect and follow a colored object (e.g. green ball) in real-time. The system employs an ESP32-CAM module for video streaming and an ESP32 microcontroller for motion control, receiving commands over Wi-Fi from a host computer running the object tracking algorithm. A GPS module was initially planned for absolute localization but was removed due to technical failures. The project demonstrates that with low-cost components and basic vision techniques, effective autonomous behavior can be achieved in a semi-structured environment.

Keywords : mobile robot, object tracking, navigation, vision, ESP32 CAM.

Résumé

Ce mémoire présente la conception et la réalisation d’un robot mobile autonome capable de détecter et de suivre un objet coloré (par ex: balle verte) en temps réel à l’aide de la vision par ordinateur. Le système repose sur un module ESP32-CAM pour la capture vidéo et une carte ESP32 pour le contrôle du mouvement, recevant des commandes via Wi-Fi depuis un ordinateur exécutant l’algorithme de suivi. Une unité GPS était prévue pour la localisation absolue, mais a été retirée suite à des dysfonctionnements techniques. Ce travail montre qu’un comportement autonome efficace peut être obtenu dans un environnement semi-structuré en utilisant des composants peu coûteux et des techniques de vision simples.

Mots-clés : robot mobile, Suivi d’objet, navigation, vision, ESP32 CAM

Table of Contents

General Introduction

General Inroduction 2
I.1 Introduction 4
I.2 Navigation 4
I.3 Perception System 5
I.4 Sensors 6
I.4.1 Definition of a Sensor 6
I.4.2 Classification of Sensors 6
I.4.3 Internal Sensors 7
I.4.3.1 Displacement Sensors 7
I.4.3.2 Atitude Sensors 8
I.4.4 Exteroceptive Sensors 9
I.4.4.1 Telemetric Sensors 10
I.4.4.2 Vision Systems 12
I.5 Localization Methods 12
I.5.1 Relative Localization 13
I.5.2 Absolute Localization 13
I.6 Conclusion 14

CHAPTER I: Generality

II.1 Introduction 15
II.2 Vision 15
II.3 Vision systems 15
II.3.1 Monocular vision 15
II.3.2 Stereovision 20
II.3.3 Omnidirectional vision 20
II.3.3.1 Fusion of multiple images 21
II.3.3.2 Use of a Wide-Angle lens 22
II.3.3.3 Use of Catadioptric Sensors 22
II.4 Navigation by Global Positioning System (GPS) 23
II.4.1 History of Space Communication 23
II.4.2 Geo-Positioning 24

II.4.2.1 General Overview 24
II.4.2.2 GPS Localization System 25
II.4.3 Different Segments of the GPS System 25
II.4.3.1 Space segment 25
II.4.3.2 Control segment 26
II.4.3.3 User segment 27
II.4.4 Differential GPS (D-GPS) 27
II.4.4.1 Some Applications of GPS 28
II.5 Conclustion 28

CHAPTER II: Vision and navigation

III.1 Inroduction 30
III.2. The Hardware Part 30
III.2.1 Definition of the ESP32 card 30
III.2.1.1 ESP32-WROOM-32 microcontroller 31
III.2.1.2 ESP 32 Card Features 32
III.2.1.3 Composition of the ESP 32 card 33
III.2.2 ESP32-CAM 34
III.2.2.1 ESP32-CAM Features 34
III.2.2.2 ESP32-CAM Pinout 35
III.2.3 LS20030~3 GPS Receivers 35
III.2.3.1 Features of LS20030~3 series 36
III.2.3.2 Application of LS20030~3 series 36
III.2.4 Mechanical platform 36
III.2.4.1 Mobiles on wheels 37
III.2.4.2 Engines 37
III.2.4.2.1 Direct Current Motors 37
III.2.5 L298N Controller 38
III.2.6 Power supply 39
III.2.6.1 18650 battery 39
III.2.6.2 Powerbank 39
III.3 The Software part 40
III.3.1 The Arduino IDE programming interface 40
III.3.2 Visual Studio Code 41

III.4 Conclusion 41

CHAPTER IV: Test and results

IV.1 Introduction 42
IV.2 Robot Assembly 42
IV.2 Communication Between Components 44
IV.3 Functional Logic Diagram 45
IV.3 Test Setup Description 46
IV.3.1 Object Tracking 47
IV.3.2 Movement Response 48
I.3.3 Observations 51
IV.4 Problems and Limitations 51
IV.6 Discussion and Analysis 52
IV.5 Conclusion 53

General Conclusion

General Conclusion 54

Table of Figurs

Figure I. 1: Navigation of a mobile robot in an environment full of obstacles. 5
Figure I. 2: Functional chain of a navigation system. 5
Figure I. 3: real-time estimation of the position (x, y) and heading θ. 8
Figure I. 4: Gyroscope Axes (PmodGYRO L3G4200D). 9
Figure I. 5: Example of a laser rangefinder (SICK). 10
Figure I. 6: How a laser rangefinder works. 10
Figure I. 7: Example of an infrared sensor 11
Figure I. 8: Principale emitted and received infrared sensor. 11
Figure I. 9: example of an ultrasonic sensor 12
Figure I. 10: CCD Camera (Conrad BC-103 Sony) 12

Figure II. 1: 3D reconstruction of a face using Shape-From-Shading. 16
Figure II. 2: Example of 3D reconstruction of a cheetah using Shape-From-Texture. 17
Figure II. 3: Application of Shape-From-Contour to image rectification. 17
Figure II. 4: 3D reconstruction of a bust using photometric stereo. 18
Figure II. 5: Example of 3D models obtained using Shape-From-Motion. 19
Figure II. 6: Results obtained using Shape-From-Defocus. 19
Figure II. 7: Stereovision sensor. 20
Figure II. 8: Principle of stereovision. 20
Figure II. 9: 3D reconstruction of a face using stereovision. 20
Figure II. 10: Field of view of vision systems (represented in gray). 21
Figure II. 11: (a) Fish-eye lens, (b) Image acquired by a fish-eye camera. 22
Figure II. 12: Example of a catadioptric sensor. 23
Figure II. 13: Control Segment. 25
Figure II. 14: Space segment 26
Figure II. 15: Locations of Control Segments Worldwide. 26
Figure II. 16: Example of a GPS receiver. 27
Figure II. 17: Example of D-GPS localization. 28

Figure III. 1: ESP32 card. 30
Figure III. 2: Block diagram of a microcontroller. 31
Figure III. 3: ESP32-WROOM-32 microcontroller. 32
Figure III. 4: Pins of an ESP32 Wroom board. 33
Figure III. 5: ESP 32 CAM module 34
Figure III. 6: Pins of an ESP32-CAM. 35
Figure III. 7: Front and rear panels of the LS20031 receiver. 36
Figure III. 8: KIT 4WD SMART CAR. 37
Figure III. 9: Wheel used. 37
Figure III. 10: 6V DC motors. 38
Figure III. 11: Pont-H. 38
Figure III. 12: L298N motor driver. 38
Figure III. 13: 18650 battery. 39
Figure III. 14: Powerbank. 40
Figure III. 15: General window of the Arduino IDE. 40
Figure III. 16: Software vs code (visual studio code). 41

Figure IV. 1: Robot Prototype. 43
Figure IV. 2: Circuit diagram 43
Figure IV. 3: Communication diagram 44
Figure IV. 4: Functional Logic Diagram 45
Figure IV. 5: Testing the robot 46
Figure IV. 6: Web interface 46
Figure IV. 7: Intrface of python video stream. 47
Figure IV. 8: Video stream with color mask applied 48
Figure IV. 9: Tracking the object above the rectangle. 49
Figure IV. 10: Tracking the object left the center. 49
Figure IV. 11: Tracking the object right of the center. 50
Figure IV. 12: Tracking the object under the rectangle. 50

Nomenclature

ADC	Analog-to-Digital Converter
BGR	Blue-Green-Red
CAM	Camera Module
CV	Computer Vision
DC	Direct Current
GPS	Global Positioning System
HSV	Hue, Saturation, Value
I/O	Input/Output
IDE	Integrated Development Environment
IP	Internet Protocol
PWM	Pulse Width Modulation
RAM	Random Access Memory
ROI	Region of Interest
S	Stop Command
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UART	Universal Asynchronous Receiver/Transmitter
Wi-Fi	Wireless Fidelity

General Inroduction

Autonomous mobile robotics has become a vital area of research and innovation across industries ranging from logistics and agriculture to defense and healthcare. At the core of a mobile robot's autonomy lies its ability to navigate to move safely and intelligently through an environment by perceiving, interpreting, and responding to its surroundings without human intervention. [1]

Three key capabilities underpin the autonomy of a mobile robot: navigation, localization, and perception through vision. Navigation is the decision-making process that enables a robot to plan and execute movements toward a specific goal while avoiding obstacles. It requires continuous evaluation of the robot's current position, orientation, and its understanding of the surrounding environment. Complementing navigation is the process of localization, which allows the robot to determine its precise position within a known or unknown space. This can be accomplished using relative techniques such as odometry and inertial measurements, or absolute methods like GPS and visual landmarks. Accurate localization is crucial for ensuring that navigation decisions are reliable, especially in dynamic or partially mapped environments. Perception, particularly computer vision, provides the robot with a sensory understanding of its surroundings. Using monocular or stereo cameras, vision systems enable the robot to detect, recognize, and track objects. When integrated with intelligent processing algorithms, visual input becomes a powerful tool for enabling behaviors such as path following, obstacle avoidance, and real-time target tracking. [2]

This thesis focuses on the development of a low-cost autonomous robot that uses monocular vision for real-time object tracking and navigation. The robot identifies and follows a green-colored object in this case, a ball based on its position within a video feed captured by an onboard ESP32-CAM module. A Python-based computer vision script processes the video stream, detects the object using color segmentation in the HSV color space, and sends movement commands wirelessly to an ESP32-based motor controller. [3]

Originally, the project also included a GPS module for absolute localization. However, due to technical failures, the module was excluded from the final design. Instead, the system operates entirely on relative, vision-based navigation, offering insights into the effectiveness of simple perception strategies for autonomous behavior. [4]

The project highlights the integration of visual perception, real-time communication, and embedded control in a unified platform. It demonstrates how even basic hardware can enable autonomous functionality when guided by a clear control strategy and a well-structured perception pipeline. The work contributes to ongoing efforts in making autonomous robotic systems more accessible, adaptable, and application-ready. [5]

Organization of the Thesis

Chapter I: Generality

This chapter introduces the fundamental concepts of mobile robotics, including navigation, perception systems, and localization methods. It also provides an overview of the sensors used in robotic systems, with distinctions between internal (proprioceptive) and external (exteroceptive) sensors.

Chapter II: Vision and Navigation

This chapter focuses on vision-based perception and its role in robotic navigation. It explores different vision system architectures such as monocular vision, stereovision, and omnidirectional vision, along with an introduction to GPS-based localization and satellite navigation systems.

Chapter III: Hardware and Software Tools

This chapter presents the components used to build the robot, including the ESP32 microcontroller, ESP32-CAM, motor drivers, power systems, and mechanical platform. It also introduces the software tools used for development and system integration, such as Arduino IDE and Visual Studio Code.

Chapter IV: Test and Results

The final chapter details the physical assembly, functional logic, and testing of the robot. It presents the results of object tracking experiments, discusses system behavior and performance, and outlines encountered limitations—such as the failure of the GPS module and proposes future improvements.

CHAPTER I

GENERALITY

I.1 Introduction

This chapter provides an introduction to the field of localization in mobile robotics. The problem of localization and the importance of knowing the position and orientation of the robot relative to its environment are presented. Various localization approaches and the associated sensors are also discussed.

I.2 Navigation

Navigation is a crucial step in mobile robotics, enabling a high degree of autonomy for mobile robots. The navigation system consists of several modules, which can be handled differently, including localization and obstacle avoidance. When the vehicle navigates in a perfectly known and non-evolving environment, its trajectory can be predetermined offline. In cases where navigation occurs in a known environment but with the possibility of unexpected obstacles, specific sensors must be used to control the navigable space in real- time and provide the necessary measurements to adjust the vehicle's trajectory online. In a poorly known environment, the trajectory must be planned online using an exteroceptive perception system capable of simultaneously building a model of the navigable space. This type of task requires the vehicle to have decision-making capabilities, which may be supplemented by human intervention. [6]

The navigation method employed by a mobile robot varies in complexity based on the operational environment, which can range from indoor settings to natural terrains, flat or uneven surfaces, and even terrestrial, underwater, or space environments. Additionally, the system's effectiveness is influenced by the level of environmental knowledge whether static or dynamic—and the approach used to determine the trajectory, such as prior learning or online planning. The performance of the navigation system is heavily dependent on the precision, reliability, and responsiveness of the sensors and localization techniques utilized. The figure demonstrates the capabilities of a mobile robot navigating an arbitrary environment while overcoming the diverse obstacles previously mentioned **(Figure I.1)**.

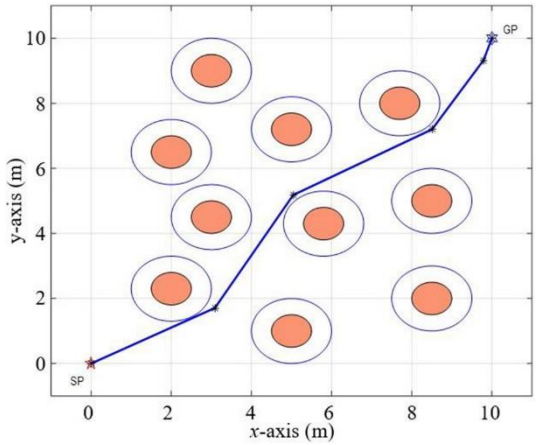


Figure I. 1: Navigation of a mobile robot in an environment full of obstacles.

In our work, we place significant emphasis on localization, a critical component for enabling robot navigation. We have developed a localization system designed to allow the robot to operate autonomously without human intervention in environments that are only partially known. By examining this basic robotics scenario, we aim to shed light on the challenges involved in creating more advanced automation systems. Key issues include selecting appropriate sensors, determining their optimal placement, and accurately localizing the mobile platform.

I.3 Perception System

Perception typically entails gathering various sensory inputs to gain knowledge and comprehension of the surrounding environment. As highlighted earlier, it serves as a critical prerequisite for the following stages, which, in the case of a mobile robot, usually involve localization and updating the environment map (**Figure I.2**). [7]

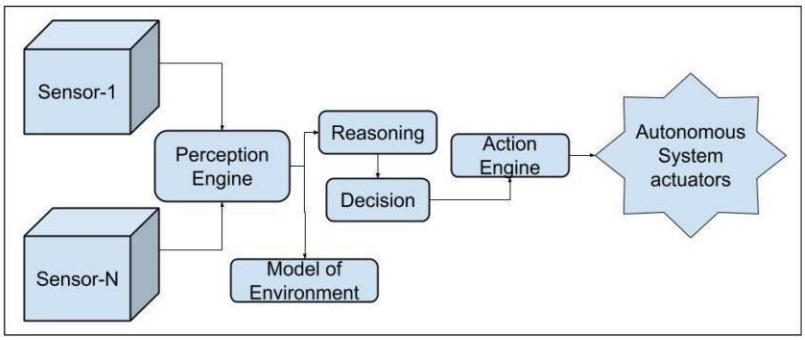


Figure I. 2: Functional chain of a navigation system.

Several observations can be made about this essential component of the functional chain of navigation:

- ✓ First, the selection of a perception system is heavily influenced by the operational environment of the mobile robot and the specific functionalities required to accomplish its mission.
- ✓ Second, a broader observation is that relying on a single sensor is rarely adequate for achieving precise environmental perception. Mobile robots typically employ perception systems that integrate multiple sensors, which may be complementary to enhance sensory data or redundant to mitigate issues related to sensor failure or degraded performance. In such cases, data fusion techniques are commonly utilized to process and integrate these sensory inputs effectively.
- ✓ Lastly, the third observation highlights the cost implications of incorporating sensors into an autonomous vehicle. Factors such as desired accuracy and high data acquisition rates often drive up sensor costs, creating a significant constraint that inevitably impacts the choice of a perception system.

I.4 Sensors

I.4.1 Definition of a Sensor

A sensor is a device designed to detect and gather information from an object or a phenomenon. It converts non-electrical measurements into electrical signals at its output. This instrument enables the continuous monitoring and measurement of specific physical quantities. [8]

I.4.2 Classification of Sensors

Sensors can be categorized based on their operational principles: active sensors, which utilize a physical phenomenon to generate voltage or current, and passive sensors, which require an external power source to convert variations in a phenomenon into voltage. Additionally, sensors can be classified by the type of output they produce, either analog or digital. The figure below illustrates the two main types of sensors commonly used in mobile robots:

• External sensors (exteroceptive): These sensors gather information about the

robot's surrounding environment.

- Internal sensors (proprioceptive): These sensors monitor the robot's internal state and enable the tracking of its movement and trajectory.

I.4.3 Internal Sensors

Proprioceptive sensors are designed to measure a robot's movement and are highly useful for localization tasks. However, they are prone to drift over time, which limits their effectiveness when used independently.

These sensors can be categorized into two main types:

- Displacement sensors: These include devices such as odometers, accelerometers, Doppler radars, and optical sensors. They are used to measure basic movements, speed changes, or accelerations, whether the robot is moving in a straight line or along a curved path.
- Attitude sensors: These sensors focus on measuring orientation data, specifically heading angles and roll angles. Common examples include gyroscopes, gyrometers, inertial measurement units (IMUs), inclinometers, and magnetometers. Most of these sensors operate on inertial principles.

I.4.3.1 Displacement Sensors

A.Odometres

Odometers provide a quantification of the robot's curvilinear displacements by measuring the rotation of its wheels. The calculation of the robot's relative position is achieved by integrating the elementary rotations of the wheels.

Odometers are generally composed of incremental encoders that measure rotation angles with a precision that depends on the encoder's resolution. The displacement information requires knowledge of the wheel diameter, the wheelbase, the mechanical and kinematic structure of the vehicle. This sensor is widely used in mobile robotics due to its simplicity of implementation and low cost. [9]

The graph below shows an example of real-time estimation of the position (x, y) and heading θ of the mobile robot (**Figure I.3)**

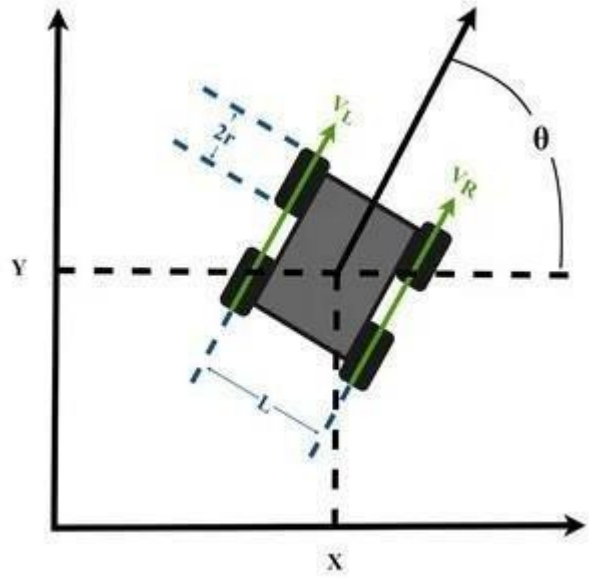


Figure I. 3: real-time estimation of the position (x, y) and heading θ.

B.Accelerometers

An accelerometer is a sensor that measures linear acceleration at a given point. In practice, acceleration is measured using a test mass M, of mass m, connected to the sensor's housing. The principle of this sensor is to measure the non-gravitational mass force required to keep M in place within the housing when an acceleration is applied to the housing. The calculation of the robot's elementary displacement is obtained by double integration of this information. [9]

I.4.3.2 Atitude Sensors

Attitude sensors measure the robot's internal parameters, including heading, roll, and pitch angles. These sensors are primarily inertial-based and tend to be costly and prone to noise interference. As a result, they are less commonly incorporated into embedded systems when compared to odometers.

A.Gyroscopes

A gyroscope consists of a solid rotating around a fixed point. It is shown that variations in angular momentum are collinear with the resultant moment of all external forces applied, implying that the axis of rotation maintains its position if no external force is applied. The assumption of zero internal forces is difficult to apply in real cases. Friction, in particular,

induces parasitic moments that cause inevitable drift. Highly sophisticated systems with reduced friction are used, especially in aircraft, but their cost is too high for rational use in mobile robotics. [10]

Here is an illustration of a gyroscope (PmodGYRO L3G4200D) **(Figure 1.4).**

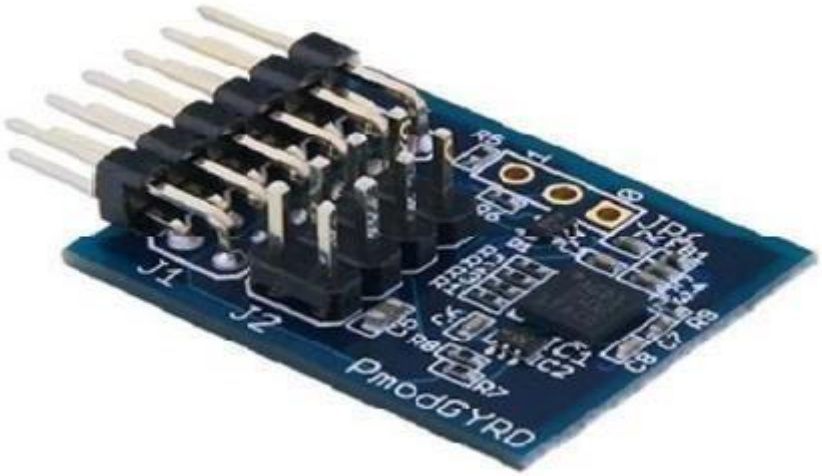


Figure I. 4: Gyroscope Axes (PmodGYRO L3G4200D).

B. Gyrocompass

A gyrocompass is a heading measurement device that integrates a gyroscope with a magnetic compass. Once it autonomously establishes the magnetic north, it consistently maintains this reference direction regardless of the vehicle's motion.

C. Magnetometer

A magnetometer is a system that measures the Earth's magnetic field to serve as an angular reference for orientation. It is also sensitive to the environment. A metallic mass nearby can alter the local magnetic field. Compensation methods are used to limit these influences, particularly those caused by the mobile platform itself. [11]

I.4.4 Exteroceptive Sensors

Exteroceptive sensors allow the robot to perceive its environment. The functioning of these sensors, commonly used in mobile robotics, is presented in the following sections: first, lasers, then infrared sensors, and finally ultrasonic sensors. Indeed, due to their lower cost compared to other sensors.

Exteroceptive sensors on board the robot serve the following purposes:

- Verification and improvement of the trajectory followed by the robot.
- Measurement of interactions between the robot and the environment.
- Perception of the environment.
- Prevention and safety [6].

I.4.4.1 Telemetric Sensors

A.Laser Rangefinders

Laser rangefinders are currently the most widely used means in mobile robotics to obtain precise distance measurements. Their principle of operation is as follows:

- At a given instant, a very short light pulse is emitted via a low-power laser diode. The reflection of this wave produces an echo detected after a time proportional to the distance between the sensor and the obstacle. The direction of the pulses is changed by rotating a mirror. [11]

The following illustrations depict: in Figure I.5 a photograph of the SICK, one of the currently available laser rangefinders, and in figure I.6 a diagram illustrating its operational mechanism.



Figure I. 5: Example of a laser rangefinder (SICK).

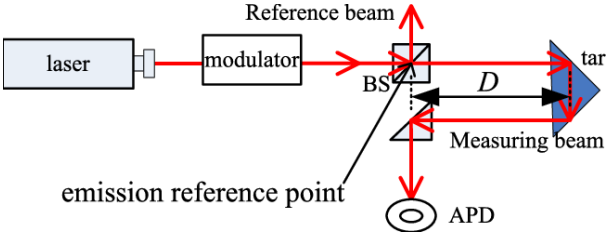


Figure I. 6: How a laser rangefinder works.

B. Infrared Sensors

Their use is similar to that of lasers. Distance measurements can be made by triangulation, intensity measurement, and time-of-flight measurement. They sometimes have wide emission angles **Figure I.7** and **Figure I.8** . [11]

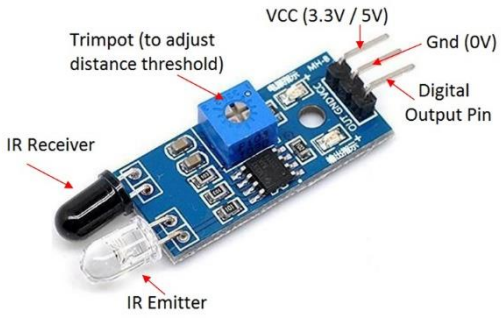


Figure I. 7: Example of an infrared sensor

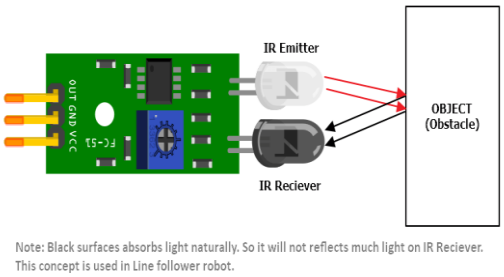


Figure I. 8: Principale emitted and received infrared sensor.

C.Ultrasonic Sensors

Unlike other types of sensors, these emit a wave that propagates through the air. This significantly increases the acquisition time. The wave attenuation is high in this medium: the measured distances remain short (a few tens of meters) but sufficient for indoor use. Two main characteristics should be noted **Figure I.9**:

- ✓ The first is the reflection of the wave on an object, which uses the laws of Descartes.
- ✓ The second arises from the sensor's operation as an emitter-receiver: the blind zone[6].

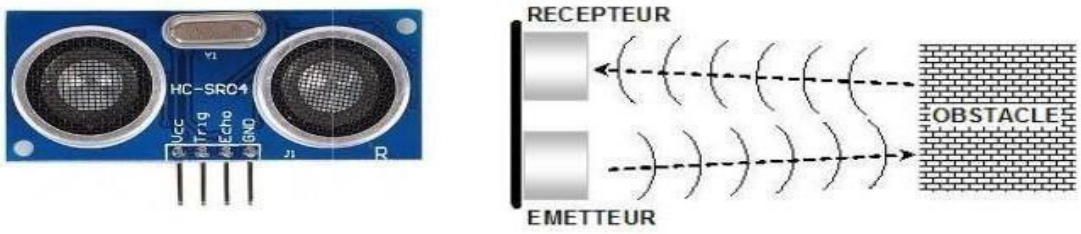


Figure I. 9: example of an ultrasonic sensor

I.4.4.2 Vision Systems

Vision systems in robotics are based on the use of a Charge Coupled Device (CCD) camera. The advent of CCD sensors in 1975 was decisive in the evolution of vision: acquisition speed, robustness, and miniaturization are advantages that have facilitated their integration. Vision systems are highly performant in terms of range, accuracy, and the amount of exploitable information. They are also the only systems capable of providing a sensory image of the environment closest to that perceived by humans **(Figure I.10)**



Figure I. 10: CCD Camera (Conrad BC-103 Sony).

I.5 Localization Methods

Instantaneous localization is one of the most important and delicate aspects of mobile robotics. It allows for the definition of positioning [11]:

- Of the mobile platform in the environment.
- Of specific elements of the environment relative to the mobile platform.

Several techniques and methods have been developed to ensure the accurate and autonomous knowledge of a mobile robot's position in its environment. To date, these techniques can be grouped into two main categories:

- Relative localization methods.
- Absolute localization methods [11].

I.5.1 Relative Localization

This method determines the current position of a mobile robot based on previous positions and the measurement of its displacements. The information used is provided by proprioceptive sensors. The odometer measures relative displacement with respect to the ground. Inertial systems measure accelerations relative to a Galilean reference frame [6].

I.5.2 Absolute Localization

Absolute localization is a method that enables a robot to determine its position and orientation within its environment, whether outdoors (such as at sea, on land, or in space) or indoors (like workshops, buildings, or nuclear power plants). This technique relies on exteroceptive sensors to continuously measure the mobile platform's position and orientation. There are numerous and diverse localization methods, but some may impose constraints that can limit the robot's mobility. To tackle the challenge of robot localization, two primary strategies are employed:

- ✓ The first strategy involves the use of artificial landmarks:

Artificial landmarks are characteristic beacons added to the robot's environment, the positions of which are known. The disadvantage of this type of technique lies mainly in its lack of flexibility and its cumbersome use. Indeed, any change in the environment's configuration will require a reconfiguration of the beacon network.

However, this technique has the significant advantage of being precise, robust, and, most importantly, capable of meeting real-time constraints.

- ✓ The second strategy utilizes natural landmarks: This approach leverages existing features of the environment to estimate the robot's position. Its main

advantage is flexibility, as it does not require altering the environment. However, it relies on the presence of identifiable natural features..

In both cases, absolute localization requires a representation of the environment, typically in the form of a map. The robot accesses a database containing the characteristics of external references, which serves as the environment map for localization purposes.

I.6 Conclusion

There are various methods and techniques available for localizing a mobile robot, which can broadly be categorized into two groups: relative localization and absolute localization. Relative localization relies on proprioceptive sensors such as odometers, accelerometers, and gyroscopes, while absolute localization utilizes exteroceptive sensors like ultrasonic sensors, infrared sensors, laser rangefinders, as well as vision-based systems (cameras) and geo-positioning systems (GPS). Currently, no single method is universally applicable. The selection of a specific technique depends on the nature of the task at hand.

In our localization project, we have opted to use one of the exteroceptive sensors, specifically the GPS receiver (Global Positioning System) and vision system by ESP 32 camera

.

CHAPITRE II

VISION AND NAVIGATION

II.1 Introduction

Navigation and vision are critical capabilities for autonomous mobile robots. This chapter explores the systems that enable robots to perceive their environment. It introduces key concepts in localization, mapping, and path planning. Vision systems provide essential data for identifying objects and features. We examine both sensor-based and visual approaches to navigation. The integration of data from various sources ensures robust movement. Challenges in dynamic environments and uncertainty are also discussed.

II.2 Vision

The development of an intelligent mobile robotic system hinges on its capacity to learn from and adapt to its surroundings. To achieve this, it is essential to employ sensors that enable the robot to perceive its environment effectively, facilitating informed decision-making. In this context, a webcam serves as the primary sensor.

This thesis focuses on addressing the navigation challenges faced by a mobile robot guided by a monocular camera mounted above it. In this section, we explore various vision systems that are well-suited for robotic applications. [12]

II.3 Vision systems

In robotics, vision systems typically rely on CCD (Charge-Coupled Device) cameras. A single camera can only capture 2D information, a method referred to as monocular vision. To achieve 3D information, additional sensors or techniques are often employed. For instance, using two cameras enables the reconstruction of an object's 3D structure through stereovision, which processes two images simultaneously. Alternatively, incorporating a mirror-based reflection system can provide omnidirectional vision, expanding the field of view. As a result, the three primary techniques in robotic vision are monocular vision, stereovision, and omnidirectional vision. [12]

II.3.1 Monocular vision

In this thesis, we concentrate on monocular vision, a system that relies on a single camera to capture the scene directly. However, this approach has notable limitations, including a restricted field of view and the inability to inherently provide depth information. Monocular vision only delivers a two-dimensional representation of both the image and the three-dimensional space, which is inherently incomplete. Depth must instead be inferred from one

or more images. In the field of computer vision, techniques for reconstructing 3D information from images are collectively referred to as "Shape-From-X" methods, where X can represent various cues such as "Stereo," "Motion," "Shading," and others. [12]

- **Shape-From-Shading analysis**

  The "Shape-From-Shading" (SFS) method, introduced by Horn in the 1970s, is a technique used to reconstruct a 3D surface from a 2D image, provided the reflective properties of the surface are known. This method is particularly effective for surfaces exhibiting lambertian reflectance.

  Modeling the image formation process allows the gray level of a pixel to be expressed as a function of the light source direction and the surface normal of the object. The goal of SFS is to recover the light source and the surface normal for each pixel in the image, given the gray level of the image **(Figure II.1)** .

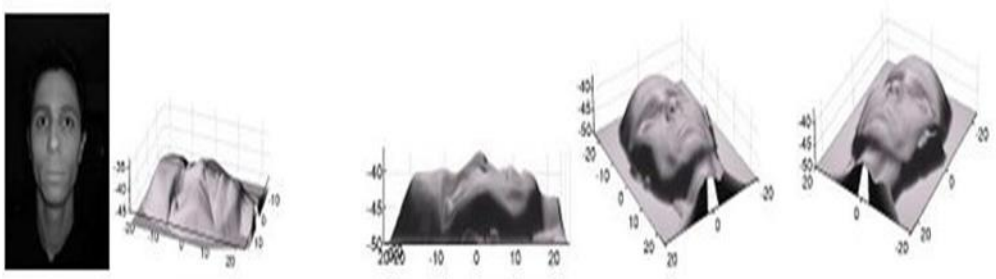


Figure II. 1: 3D reconstruction of a face using Shape-From-Shading.

- **Shape-From-Texture analysis**

  The "Shape-From-Texture" technique was initially introduced by Gibson in 1950. He explored the process of reconstructing the 3D coordinates of a surface in a scene by examining the deformation of its texture as projected in an image. The "Shape-From-Texture" problem is typically approached in two distinct stages: the first stage focuses on quantifying the texture distortion within the image, while the second stage involves deducing the surface coordinates based on this observed distortion **(Figure II.2).** [12]

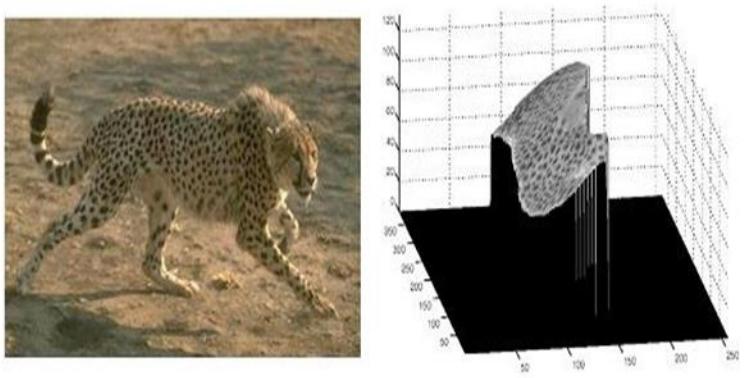


Figure II. 2: Example of 3D reconstruction of a cheetah using Shape-From-Texture.

- **Shape-From-Contour:**

  This approach offers an estimation of the 3D shape of an object's contours by constructing a 3D model from images captured from multiple viewing angles, utilizing the "volume-carving" technique[7]. While this method is effective in generating a preliminary 3D representation, it has limitations in reconstructing certain concave regions on the object's surface. Additionally, the accuracy of the resulting 3D model is influenced by the resolution of the viewing directions. Despite these constraints, the method is less sensitive to lighting conditions, making it a robust solution for obtaining an initial 3D model of an object **(Figure II.3).** [14]



Figure II. 3: Application of Shape-From-Contour to image rectification.

- **Shape-From-Photometric stereo**

  Woodham introduced the technique called "Shape-From-Photometric Stereo" in 1980. This method uses two or more intensity images of the object, keeping the same viewpoint under different lighting conditions, by changing the

light direction between shots [12]. The Photometric Stereo method efficiently derives local surface orientations from these intensity images, which are illuminated by different light sources. These surface orientation vectors are then integrated, either globally or locally, to determine the surface depth values. As demonstrated in **Figure II.4**, this technique produces superior results.

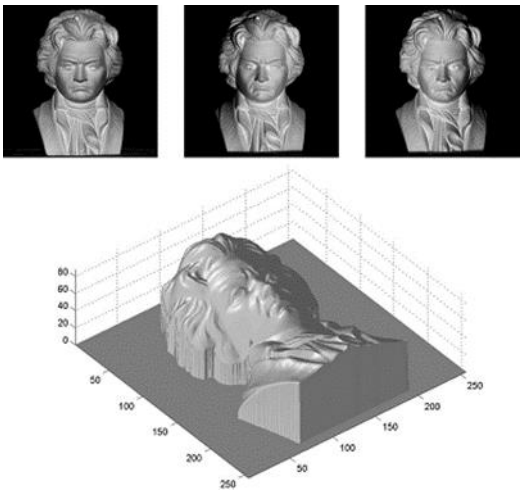


Figure II. 4: 3D reconstruction of a bust using photometric stereo.

- **Shape-From-Motion**

  This method is referred to as "Structure-from-Motion" (SFM). It leverages the relative movement between the camera and the scene. Like the stereo technique, the process involves identifying correspondences between consecutive frames and reconstructing the scene. However, the differences between consecutive frames are generally smaller compared to those in a typical stereo pair, as image sequences are captured at a high sampling rate. Unlike stereo, in SFM, the 3D relative displacement between the camera's viewpoint and the scene is not always the result of a single 3D transformation **(Figure II.5)**.

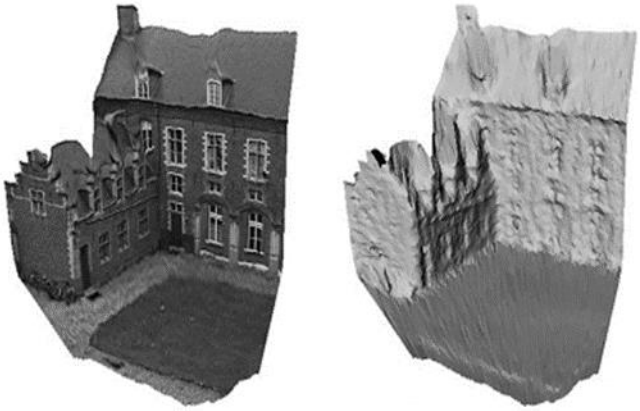


Figure II. 5: Example of 3D models obtained using Shape-From-Motion.

- **Shape-From-Focus/Defocus**

The "Shape-From-Focus/Defocus" problem focuses on reconstructing the surface of a 3D scene using two or more images captured from the same viewpoint. These images are acquired by adjusting camera parameters, such as the focal length or the axial position of the image plane.

The distinction between "Shape-From-Focus" and "Shape-From-Defocus" lies in the flexibility of modifying camera parameters during the reconstruction process. In "Shape-From-Focus," the camera parameters can be dynamically adjusted while estimating the surface, whereas in "Shape-From-Defocus," such adjustments are not permitted. [15]



(a) (b) (c)

Figure II. 6: Results obtained using Shape-From-Defocus.

II.3.2 Stereovision

Stereovision involves using two cameras separated by a known distance and observing the same scene **(Figure II.7).** When two cameras are used, it is called a binocular system, which is sufficient to reconstruct the observed scene in 3D or retrieve depth information through triangulation. This can be used for obstacle avoidance or mapping.

The main difficulty in stereovision lies in pixel matching, i.e., determining the projections **m1** and **m2** of the same point **M** in the images. Once the pixels are matched, the coordinates of point **M** are obtained by triangulating points **m1** and **m2** from the optical centers **C1** and **C2** of the cameras , as illustrated in **(Figure II.8)** [13].



Figure II. 7: Stereovision sensor.

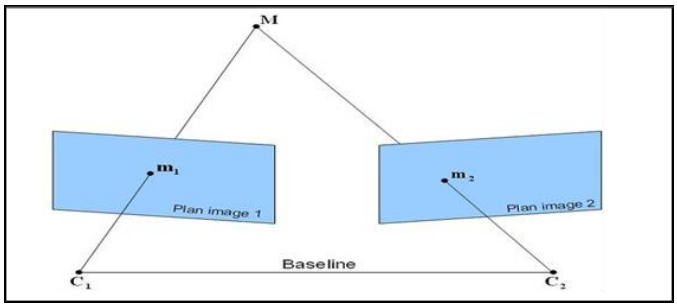


Figure II. 8: Principle of stereovision.

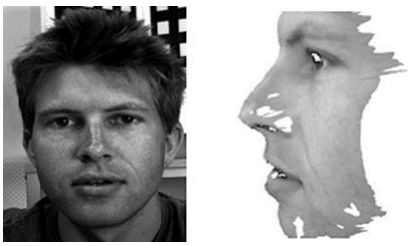


Figure II. 9: 3D reconstruction of a face using stereovision.

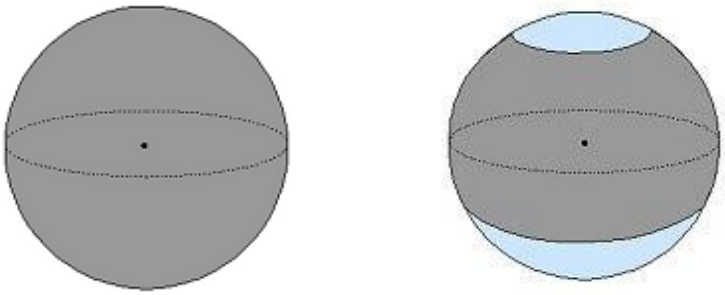
II.3.3 Omnidirectional vision

The term omnidirectional vision refers to a vision sensor with a very wide field of view, a sensor with a 360° field of view. Another term for a wide field of view is panoramic vision.

These two terms are often used interchangeably, although there is a difference between them. [12]

- Omnidirectional vision is the process of vision with a 360° horizontal field of view and a vertical field of view that may or may not be limited, so its field of view must be spherical, allowing the world to be observed from its center, as illustrated in **Figure 1.10**(a).

- Panoramic vision is a reduction of the first process, with a limited field of view. It has a 360° horizontal field of view with a limited vertical field of view. Therefore, it does not cover the entire environment in another sense, as illustrated in **Figure 1.10**(b).



(a) (b)

Figure II. 10: Field of view of vision systems (represented in gray).

(a) Panoramic vision - (b) Omnidirectional vision.

Several techniques have been developed to increase the panoramic and omnidirectional field of view, which can be classified into three categories :

- Using multiple images to form a panorama.
- Using wide-angle lenses.
- Using a mirror.

II.3.3.1 Fusion of multiple images

The first approach involves creating a panoramic image by combining multiple images, this method presents several practical challenges. Image acquisition can be achieved through two techniques:

- Rotating a single camera around a fixed axis.
- Using multiple cameras simultaneously.

II.3.3.2 Use of a Wide-Angle lens

A wide-angle lens, characterized by its short focal length, is designed to capture a broad field of view. Among such lenses, fisheye lenses offer an extreme wide-angle perspective. A fisheye camera is a catadioptric system that combines a very short focal length with a strategically placed mirror to significantly expand the viewing angle. This configuration enables an impressive field of view nearly 180 degrees far exceeding that of conventional lenses, and allows for semi-spherical spatial processing. Due to their compact size, these cameras can be conveniently mounted on vehicles or robots. When installed on a mobile robot, they are typically angled downward, mimicking the downward gaze of a bird in flight, this system has a notable drawback: while image quality is sharp at the center, resolution degrades dramatically toward the edges, as seen in **Figure 1.11**(b), where peripheral distortion is evident. Because of this limitation and the resulting lengthy image processing time, fisheye cameras are often unsuitable for robotic applications.



(a) (b)

Figure II. 11: (a) Fish-eye lens, (b) Image acquired by a fish-eye camera.

II.3.3.3 Use of Catadioptric Sensors

Cameras that depend entirely on the lens's refraction effect to direct light are known as dioptric cameras (dioptrics refers to the study of refractive elements). Those that combine reflection from a mirror with refraction from a lens are called catadioptric cameras (derived from catoptrics, the science of reflective surfaces, and dioptrics). A catadioptric sensor (Figure 1.12) typically includes a camera and a mirror, often with a rotational symmetry axis.

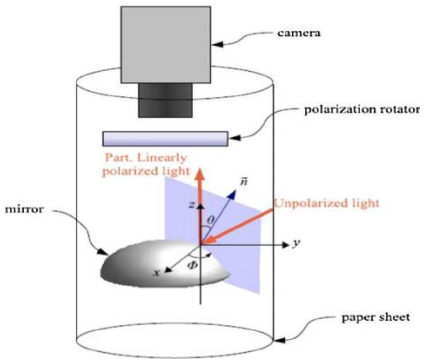


Figure II. 12: Example of a catadioptric sensor.

Here is a brief overview of the most commonly used types of omnidirectional mirrors. Only certain shapes of mirrors of revolution maintain the single viewpoint constraint, namely spherical, ellipsoidal, and hyperboloidal mirrors with a perspective camera, and paraboloidal mirrors with an orthographic camera. These shapes have been widely used in mobile robots and can be referred to as classic mirror shapes. Each of these mirror shapes creat es a different mapping between world points and image points. [12]

II.4 Navigation by Global Positioning System (GPS)

GPS is a satellite-driven navigation system that delivers real-time data on location, speed, and timing, accessible globally at any moment and in all weather conditions. To introduce this chapter, we will first explore the history of space communication. Following that, we will provide a comprehensive explanation of the GPS positioning system and its fundamental operating principles.

II.4.1 History of Space Communication

The advent of radio waves as a navigation tool emerged alongside the development of directional loop antennas and radio direction-finding technology. Initially, radio beacons were used to aid ships navigating the North Atlantic before being adopted for aviation purposes.

As early as 1912, radio direction-finding stations operating on the 450-meter wavelength were established in numerous countries to determine the positions of ships, aircraft, and dirigibles upon request. By 1920, the Ouessant Gonio radio direction-finding station, identified by the call sign FFU (French Fixed Station of Ushant), was providing up to 10 bearings daily.

A significant breakthrough occurred between 1940 and 1945 with the introduction of the first hyperbolic navigation systems, LORAN and Decca. These systems calculated positions by measuring the time difference in signal propagation between at least two transmitters, with the points of equal time differences forming hyperbolas on a map. To pinpoint a location, three transmitters were required to create intersecting hyperbolas, and to ensure accuracy, three or more synchronized transmitters were organized into chains. The first hyperbolic system, LORAN-A, operated at 1,800 KHz. Early measurements relied on oscilloscopes onboard aircraft, but advancements in electronics eventually enabled the direct display of time differences (TD) and, ultimately, geographic coordinates.

Satellite positioning systems, known as Global Navigation Satellite Systems (GNSS), consist of a network of artificial satellites that provide users with precise 3D positioning, velocity, and time data via portable receivers. These systems are renowned for their metric-level accuracy, global coverage, and compact terminals, though their performance can be affected by obstacles blocking the line of sight between the receiver and satellites. To enhance reliability and performance, various augmentation systems—both regional and global, free and paid—have been developed, such as DGPS, EGNOS, and Assisted GPS (A-GNSS).

The first satellite positioning system, TRANSIT, was developed by the United States in 1964 for military use. This was followed by the Global Positioning System (GPS), which became operational in 1995 and set the standard for satellite navigation systems worldwide. Today, GPS comprises 24 satellites orbiting at altitudes exceeding 20,000 km and was made available for civilian use in 2000. Its principles have since been adopted by other nations developing their own satellite navigation systems. [16]

II.4.2 Geo-Positioning

II.4.2.1 General Overview

Satellite geo-positioning enables individuals to determine their precise location, whether on land or at sea, with the accuracy depending on the quality of the receiver. In everyday life, GPS technology is extensively utilized by drivers to plan and navigate routes. The receiver calculates the distance between the starting point and the destination, providing an estimated travel time for the journey.

II.4.2.2 GPS Localization System

GPS is a satellite-based navigation system designed to provide instant information on position, velocity, and time virtually anywhere on Earth, at any time, and under any weather conditions [17].

II.4.3 Different Segments of the GPS System

When discussing GPS, most people typically think of the device that delivers location information. However, the GPS system is far more comprehensive than just this single component, as it is made up of three separate parts, known as segments or sectors (**Figure II.13)**:

- The initial segment, which involves satellites, is referred to as the "space segment."
- The second segment, encompassing control stations, is known as the "control segment."
- The final segment, which includes GPS receivers, is called the "user segment."

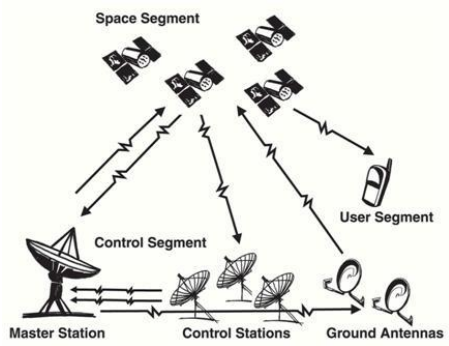


Figure II. 13: Control Segment.

II.4.3.1 Space segment

The space segment comprises a constellation of 24 satellites orbiting the Earth in quasi-circular paths at an altitude of roughly 20,000 km. These satellites are positioned across six orbital planes, each inclined at 55° to the equator. This arrangement guarantees that at least six satellites are continuously visible from any location on Earth. The system is designed to operate for a minimum of 7 years **(Figure II.14)**.



Figure II. 14: Space segment

II.4.3.2 Control segment

The control segment consists of five tracking stations situated in Colorado Springs, Hawaii, Ascension Island, Diego Garcia, and Kwajalein (Refer to **Figure II.15**) .

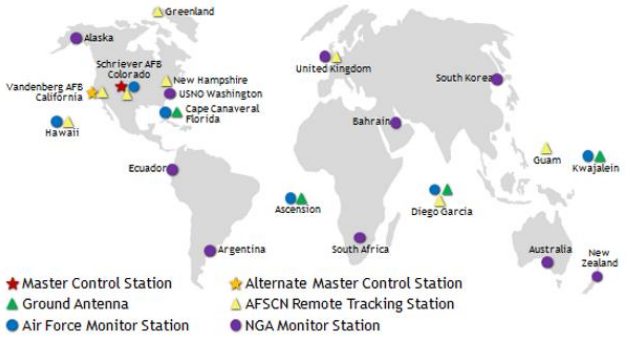


Figure II. 15: Locations of Control Segments Worldwide.

These ground stations serve as the vigilant eyes and ears of the GPS system, continuously monitoring the status of satellites as they pass overhead. The data collected by these stations is relayed to the central facility in Colorado Springs, where precise calculations are made to determine the satellites' orbital parameters and assess the accuracy of their onboard clocks. Additionally, the system monitors the overall health of the satellites and determines if any adjustments or repositioning are required. This critical information is then sent to three upload stations, which transmit the updates back to the satellites. Due to the strategically uniform distribution of these control stations, all GPS satellites are tracked and monitored 92%

of the time, ensuring the system's reliability and accuracy. Control stations on Earth constantly track the satellites' trajectories, regularly making adjustments to their orbits as needed..

II.4.3.3 User segment

The user segment comprises receivers specifically designed to decode signals transmitted by satellites, enabling the determination of position, velocity, and time. These receivers capture signals from satellites using GPS technology, which allows them to compute their three-dimensional location (latitude, longitude, altitude) and precise time through triangulation. To grasp how this system functions, it is essential to recognize that the signals emitted by satellites travel at the speed of light. By comparing the time a signal is sent from a satellite with the time it is received, the duration of the signal's journey can be measured. This data is then used to calculate the distance between the satellite and the receiver, ultimately determining the target's exact position.



Figure II. 16: Example of a GPS receiver.

II.4.4 Differential GPS (D-GPS)

Differential positioning techniques aim to enhance accuracy by correcting errors at any given location using error measurements from a known reference point. In the context of D-GPS, a reference receiver, known as the D-GPS Base Station, computes corrections for each satellite signal it receives and transmits these corrections to all receivers within its range. By applying this correction data, the position estimation is significantly improved, reducing errors to approximately 1 meter **(Figure II.17).** [18]

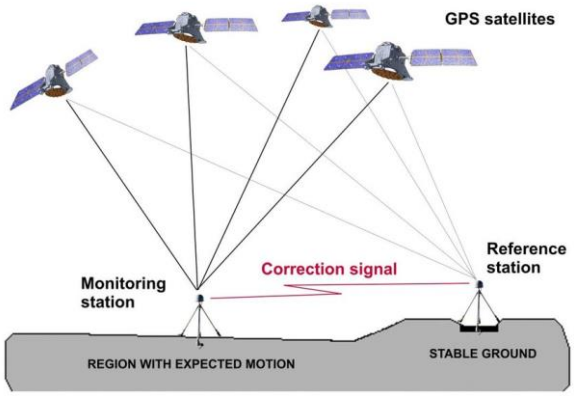


Figure II. 17: Example of D-GPS localization.

II.4.4.1 Some Applications of GPS

A. Civilian Applications

- GPS serves as a vital measurement tool in scientific fields like geophysics and tectonics.
- In the industrial sector, GPS is extensively used for oil platform exploration.
- Rescue operations benefit significantly from GPS technology, enhancing efficiency and accuracy.
- Leading automotive manufacturers prioritize navigation assistance as a key area of applied research.
- GPS plays a critical role in air and maritime navigation.
- It supports geodetic and geographic studies, enabling precise data collection and analysis.
- Rescue teams rely on GPS for improved coordination and response during emergencies.

B. Militaires

- GPS has a wide range of military uses, including target acquisition, missile guidance, and all-weather parachuting.
- It facilitates the coordination of bombings and enhances the precision of remote command, control, and surveillance operations for vehicles.

II.5 Conclusion

This chapter examined GPS-based navigation and vision systems for autonomous robots. GPS delivers real-time global positioning via satellite triangulation, while vision systems (including monocular, stereovision, and omnidirectional setups) facilitate

environmental perception. Monocular methods, such as Shape-From-X techniques, derive 3D information from 2D images, whereas stereovision employs dual cameras to estimate depth. Omnidirectional systems, which use mirrors or wide-angle lenses, provide 360° coverage but compromise on resolution. Combining GPS with vision enhances reliability in dynamic settings. Future developments may prioritize higher accuracy and computational efficiency for real-time performance. Collectively, these technologies underpin autonomous robotic navigation and localization. The following chapter will present our GPS reception module, the LS20031 and ESP32-CAM along with the ESP WROOM 32 board employed to command it.

CHAPITRE III

HARDWARE AND SOFTWARE TOOLS

III.1 Inroduction

This chapter explores the hardware and software tools essential for building an autonomous robotic platform. The hardware section covers the ESP 32 microcontroller, ESP32-CAM module, ultrasonic sensors, and motor drivers, detailing their specifications and functionalities. The software part introduces Arduino IDE and Visual Studio Code for programming and system control. These components form the backbone of the robot's navigation, perception, and processing capabilities. By integrating these tools, the system achieves efficient real-time operation. The chapter also highlights the mechanical platform and power supply to ensure robust physical execution. Together, these elements enable the development of a fully functional autonomous robot.

III.2. The Hardware Part

III.2.1 Definition of the ESP32 card

The ESP32 is a low-power microcontroller equipped with wireless connectivity features such as WiFi and Bluetooth, developed by the Chinese company Espressif Systems.

The full reference for the ESP32 is "ESP32-WROOM-32," which is a system-on-a-chip (SoC) that integrates a 32-bit dual-core processor, WiFi and Bluetooth connectivity, Ethernet interfaces, camera interfaces, and advanced digital signal processing capabilities.

The ESP32-WROOM-32 is commonly used in IoT projects, smart devices, and other applications requiring wireless connectivity and low power consumption. It can be programmed using various development environments such as the Arduino IDE and MicroPython, and it supports a wide range of software libraries to simplify programming and integration into projects.[19]



Figure III. 1: ESP32 card.

III.2.1.1 ESP32-WROOM-32 microcontroller

A microcontroller is typically composed of three elements:

- Central Processing Unit (CPU)
- Memory (read-only memory and random-access memory),
- Input and output ports.

A microcontroller is an integrated electronic circuit that includes a processor, memory, and input/output interfaces. It is programmable and is used to control the operations of an electronic system. Microcontrollers are often used in embedded applications such as temperature control systems, lighting systems, security systems, monitoring systems, and household appliances. The advantages of microcontrollers are their low cost, low power consumption, and small size [19].

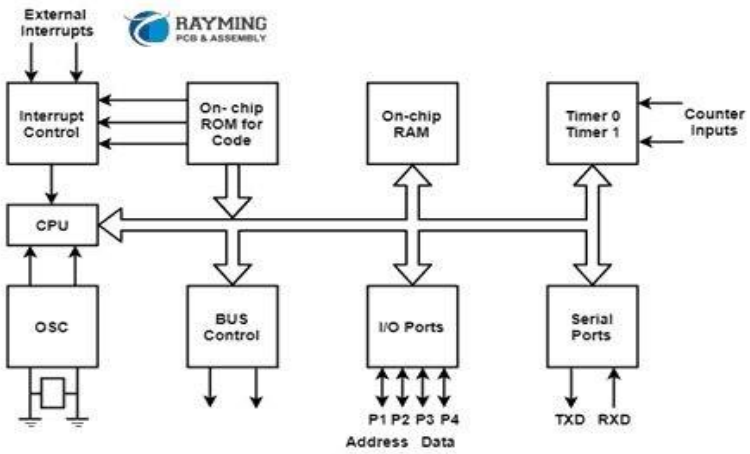


Figure III. 2: Block diagram of a microcontroller.

The ESP32-WROOM-32 microcontroller is a system-on-chip (SoC) developed by the Chinese company Espressif Systems, integrating a microcontroller based on a dual-core Xtensa architecture, along with Wi-Fi and Bluetooth connectivity systems. The ESP32-WROOM-32 microcontroller is equipped with two 32-bit Xtensa LX6 processor cores clocked at up to 240 MHz, enabling it to run heavy processing applications while consuming low power. It also includes 4 MB of integrated flash memory for program and data storage, as well as 520 KB of SRAM for program execution. In terms of wireless connectivity, the ESP32-WROOM-32 microcontroller supports Wi-Fi 802.11 b/g/n/e/i (2.4 GHz) with transfer rates ranging from 23

to 150 Mbps, as well as Bluetooth v4.2 BR/EDR and BLE for wireless communication with other devices.

The ESP32-WROOM-32 microcontroller also features a variety of peripheral interfaces, such as UART, SPI, I2C, I2S, PWM, ADC, and DAC, which can be used to communicate with a wide range of peripherals, including sensors, actuators, and display modules [19].



Figure III. 3: ESP32-WROOM-32 microcontroller.

III.2.1.2 ESP 32 Card Features

The features of the ESP32 board may vary depending on the specific model, but here are some of the general characteristics common to most ESP32 boards:

- **Microcontroller**: Dual-core ESP32 based on Cadence's Xtensa LX6 architecture.
- **Processor Speed**: Up to 240 MHz.
- **Memory**: 520 KB to 4 MB RAM, 4 MB to 16 MB flash memory.
- **Wireless Connectivity**: Wi-Fi 802.11 b/g/n, Bluetooth 4.2 BR/EDR, and BLE.
- **Input/Output Interfaces**: GPIO, SPI, I2C, UART, etc.
- **Other Peripherals**: Real-time clock (RTC), analog-to-digital converter (ADC), digital-to-analog converter (DAC), etc.
- **Programming**: Support for various development environments such as Arduino IDE, MicroPython, ESP-IDF, etc.
- **Power Supply**: Can be powered via USB or an external power source.
- **Dimensions**: May vary by model, but typically around 5 cm x 2.5 cm. [19]

III.2.1.3 Composition of the ESP 32 card

The 38-pin ESP32 board is also known as the ESP32-WROOM-32D. Here is the typical composition of this board:

- **Microcontroller**: Dual-core ESP32 based on Cadence's Xtensa LX6 architecture.
- **Memory**: 520 KB to 4 MB RAM, 4 MB to 16 MB flash memory.
- **Wireless Connectivity**: Antennas for Wi-Fi 802.11 b/g/n and Bluetooth 4.2 BR/EDR & BLE.
- **Input/Output Interfaces**: 38 pins for connecting various peripherals, including GPIO, SPI, I2C, UART, etc.
- **Converters**: Analog-to-digital converter (ADC), digital-to-analog converter (DAC).
- **Real-Time Clock (RTC)**: Provides timekeeping for real-time applications.
- **Power Ports**: Connectors for external power and/or a USB port for power and data transfer.
- **Buttons and LEDs**: Some models may include buttons for programming and configuration, as well as LEDs to indicate device status. [19]



Figure III. 4: Pins of an ESP32 Wroom board.

III.2.2 ESP32-CAM

The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera and microSD card socket. It's inexpensive and easy to use, and is perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition. [20]

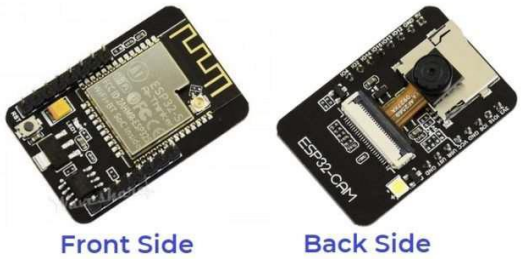


Figure III. 5: ESP 32 CAM module

III.2.2.1 ESP32-CAM Features

- **Module Model:** ESP32-CAM.
- **Package:** DIP-16.
- **Size:** 27*40.5*4.5 mm.
- **SPI Flash:** Default 32Mbit.
- **RAM:** 520KB SRAM +4M PSRAM.
- **Bluetooth:** Bluetooth 4.2 BR/EDR and BLE standards.
- **Wi-Fi:** 802.11 b/g/n/.
- **Support interface:** UART/SPI/I2C/PWM.
- **Support TF card:** Maximum support 4G.
- **IO port :** 9.
- **UART Baudrate :** Default 115200 bps.
- **Image Output Format:** JPEG(OV2640 support only), BMP, GRAYSCALE.
- **Spectrum Range:** 2412 ~2484MHz.
- **Antenna:** Onboard PCB antenna, gain 2dBi.
- **Transmit Power:** 802.11b: 17±2 dBm (@11Mbps).
  802.11g: 14±2 dBm (@54Mbps).
  802.11n: 13±2 dBm (@MCS7).
- **Receiving Sensitivity:** CCK, 1 Mbps: -90dBm.
  CCK, 11 Mbps: -85dBm.
  6 Mbps (1/2 BPSK): -88dBm.

54 Mbps (3/4 64-QAM): -70Dbz.

MCS7 (65 Mbps, 72.2 Mbps): -67dBm.

- **Power Dissipation:** Deep-sleep: 6mA@5V.
- **Moderm-sleep:** 20mA@5V.
- **Light-sleep:** 6.7mA@5V.
- **Security:** WPA/WPA2/WPA2-Enterprise/WPS.
- **Power Supply Range:** 5V.
- **Operating Temperature:** -20 °C ~ 85 °C.
- **Storage Environment:** -40 °C ~ 90 °C, < 90%RH.
- **Weight:** 10g.

III.2.2.2 ESP32-CAM Pinout

The ESP32-CAM module has fewer I/O pins than the previous ESP-32 module we looked at. Many of the GPIO pins are used internally for the camera and the microSD card port. [20]

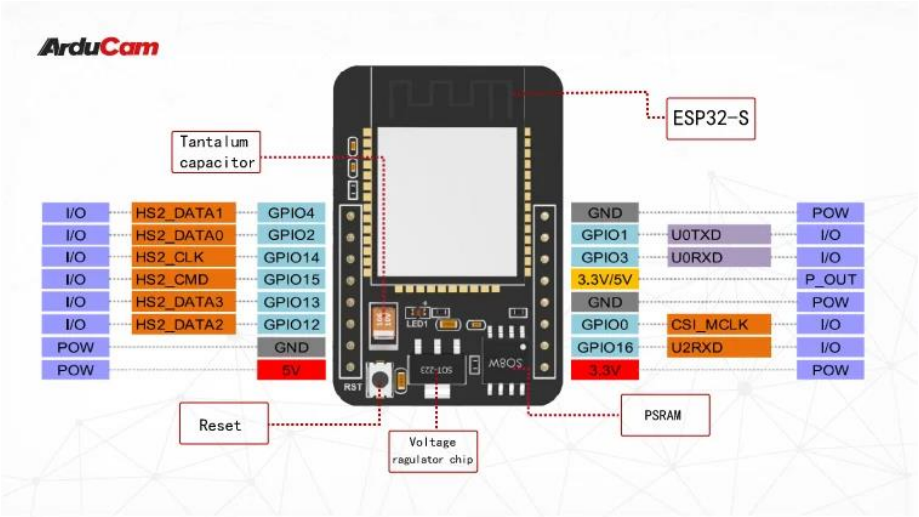


Figure III. 6: Pins of an ESP32-CAM.

III.2.3 LS20030~3 GPS Receivers

LS20030~3 series products are complete GPS smart antenna receivers, including an embedded antenna and GPS receiver circuits, designed for a broad spectrum of OEM system applications. The product is based on the proven technology found in LOCOSYS 66 channel GPS SMD type receivers MC-1513 that use MediaTek chip solution. The GPS smart antenna will acquire up to 66 satellites at a time while providing fast time-to-first-fix, one-second

navigation update and low power consumption. It can provide you with superior sensitivity and performance even in urban canyon and dense foliage environment. Its far-reaching capability meets the sensitivity requirements of car navigation as well as other location-based applications. [21]

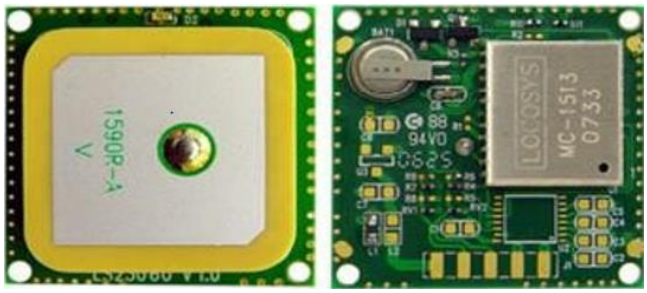


Figure III. 7: Front and rear panels of the LS20031 receiver.

III.2.3.1 Features of LS20030~3 series

- MediaTek high sensitivity solution.
- Support 66-channel GPS.
- Low power consumption.
- Fast TTFF at low signal level.
- Built-in 12 multi-tone active interference canceller.
- Free hybrid ephemeris prediction to achieve faster cold start.
- Built-in data logger.
- Up to 10 Hz update rate.
- Capable of SBAS (WAAS, EGNOS, MSAS, GAGAN).
- Support Japan QZSS.
- Indoor and outdoor multi-path detection and compensation.

III.2.3.2 Application of LS20030~3 series

- Personal positioning and navigation
- Automotive navigation
- Marine navigation

III.2.4 Mechanical platform

The KIT 4WD SMART CAR is an advanced robotics kit designed for building a four-wheel-drive (4WD) autonomous vehicle, offering better stability and control than 2WD models.

It includes four motors, reinforced wheels, a sturdy chassis, and various sensors (such as ultrasonic, infrared, or LiDAR) for obstacle detection and navigation. Powered by a microcontroller like Arduino or Raspberry Pi, it supports programming for smart functions like self-driving, line following, and remote control via Bluetooth or Wi-Fi. Ideal for robotics enthusiasts and students, this kit provides hands-on experience in coding, electronics, and mechatronics. With its robust 4WD system, it excels in off-road terrain and complex robotics projects.



Figure III. 8: KIT 4WD SMART CAR.

III.2.4.1 Mobiles on wheels

The robot moves with the help of four parallel wheels and they are equipped with rubber tires that allow it to roll outside without slipping.



Figure III. 9: Wheel used.

III.2.4.2 Engines

III.2.4.2.1 Direct Current Motors

Robot movements are carried out by electric motors, DC motors. The stator consists of a permanent magnet for small motors or an electromagnet, powered by direct current, for larger

motors. The rotor consists of an electromagnet that is powered by direct current in one direction and then the other.



Figure III. 10: 6V DC motors.

III.2.5 L298N Controller



Figure III. 11: Pont-H.

This breakoutboard is a Dual H-Bridge for DC motor control (H-Bridge Motor Driver). It is based on the L298N component which is a dual H-Bridge designed specifically for this use case. It is an extremely useful module for controlling robots and mechanized assemblies. It can control two DC motors or one 4-wire, 2-phase stepper motor. It is designed to support higher voltages, high currents while offering TTL logic control. (Transistor-Transistor Logic) Both types of motors can be controlled in both PWM (Pulse Width Modulation) speed and direction. All power outputs are already protected by anti-return diodes. This is a ready-to-use module. [22]

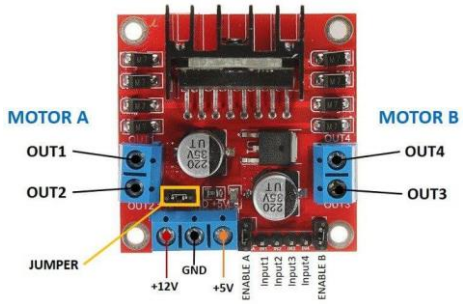


Figure III. 12: L298N motor driver.

III.2.6 Power supply

III.2.6.1 18650 battery

An 18650 battery 4.2V 6800 mAh or 1865 cell is a cylindrical battery size (often lithium-ion battery or sodium ion battery) common in electronic devices. The batteries measure 18 mm (0.71 in) in diameter by 65.0 mm (2.56 in) in length, giving them the name 18650.The battery comes in many nominal voltages depending on the specific chemistry used.[23]

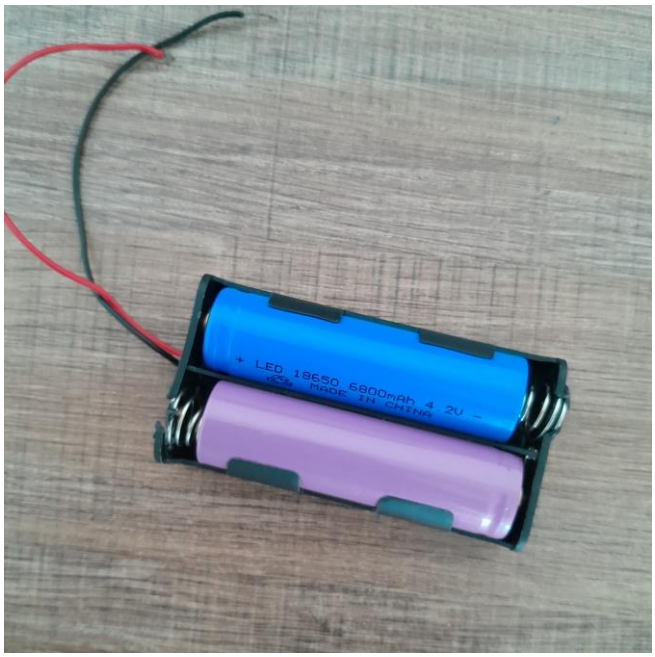


Figure III. 13: 18650 battery.

III.2.6.2 Powerbank

It is used to power a cell phone. It is characterized by high performance and is rechargeable. It operates at a temperature of 25 degrees Celsius, has an initial current of 2 A, and has a battery life of one hour.

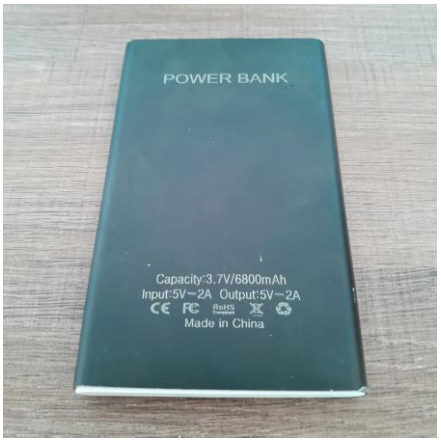


Figure III. 14: Powerbank.

III.3 The Software part

III.3.1 The Arduino IDE programming interface

Interactive Development Environment (IDE) is a programmer used to compile and interpret codes to develop programs used on an Arduino, ESP32, ESP8266, etc. board. IDE is an application written in Java based on the Processing language. The IDE allows you to write, modify a program, and convert it into a set of instructions that are understandable to the board.[19]

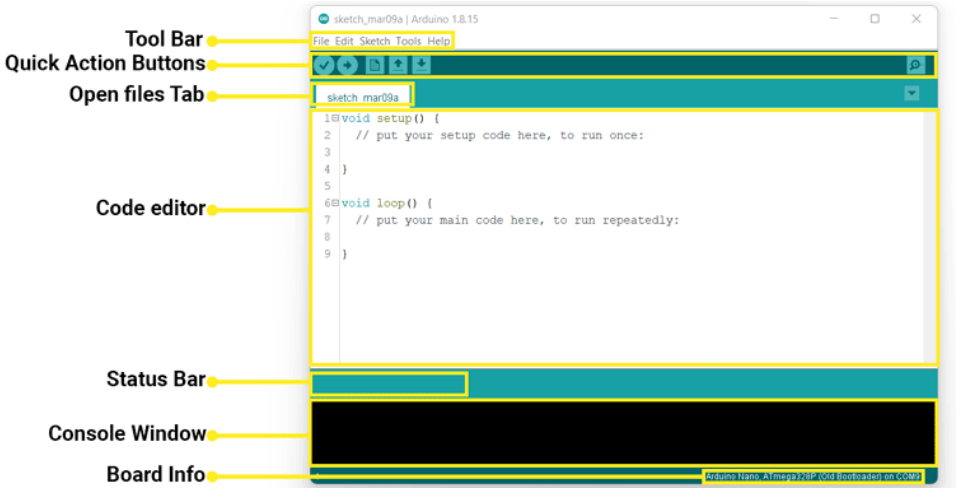


Figure III. 15: General window of the Arduino IDE.

The program is read by the microcontroller from top to bottom. A variable must be declared before it can be used by a function. The minimal structure includes:

- A header: variable declarations, constants, library usage information, etc.

- A setup: this section is read only once; it includes the functions to be executed at startup (using pins as inputs or outputs, powering up via MIDI, I2C serial port, etc.).
- A loop: this section is played within a loop; this is where the functions are executed. In addition to this minimalist structure, we can add "subroutines" or "routines" that can be called at any point within the loop, which is very useful for producing repetitive pieces of code.
- Callbacks are functions that are automatically recalled from a library.[19]

III.3.2 Visual Studio Code

Visual Studio Code, as shown in Figure III.30, is a lightweight yet powerful source code editor that runs on your desktop and is available for Windows, macOS, and Linux. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, HTML, Css).

It is used to program the program the robot interface into a web page. [24]

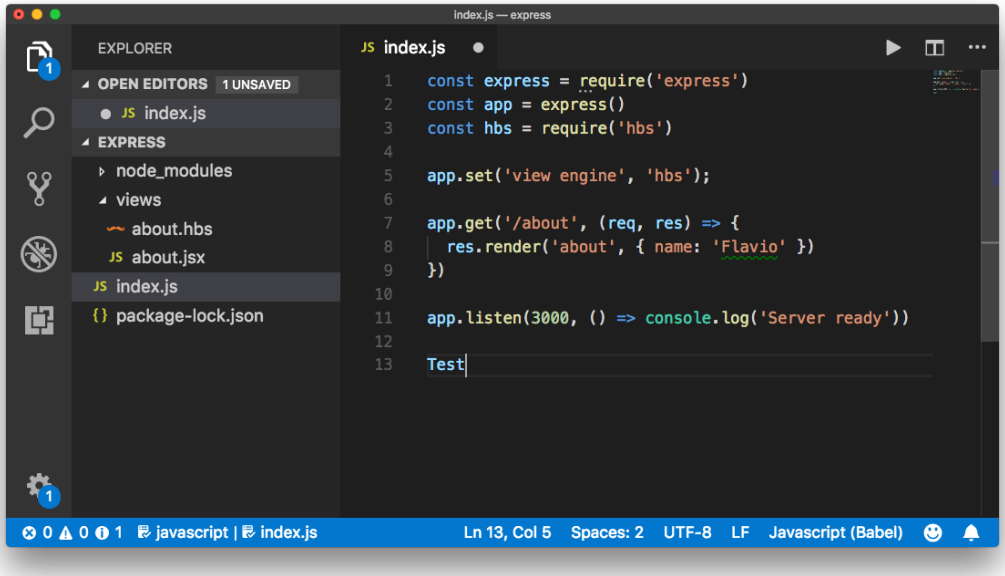


Figure III. 16: Software vs code (visual studio code).

III.4 Conclusion

This chapter provided a comprehensive overview of the hardware and software components used in the robotic system. The Arduino nano and ESP32-CAM enable wireless connectivity and vision-based navigation, while ultrasonic sensors enhance obstacle detection.

The L298N motor driver ensures precise control of movement, supported by a reliable power supply. On the software side, Arduino IDE and VS Code facilitate efficient programming and system integration. These tools collectively empower the robot to perform autonomous tasks with accuracy and adaptability. Future work could focus on optimizing sensor fusion and computational efficiency. This foundation sets the stage for advanced robotic applications.

CHAPITRE IV

TEST AND RESULTS

IV.1 Introduction

This chapter presents the practical implementation, testing, and performance evaluation of the autonomous mobile robot developed for this thesis. The primary goal of the robot is to autonomously track and follow a green-colored object in real-time, using visual input from an onboard ESP32-CAM and control logic transmitted via Wi-Fi. This behavior is achieved through monocular vision processing and embedded control communication over a wireless UDP protocol.

Initially, the project aimed to combine vision-based tracking with GPS-assisted localization for broader navigation capabilities. The hardware plan included the integration of an LS20031 GPS module, enabling the robot to determine its absolute position in open environments. However, during development and testing, the GPS module proved unreliable due to inconsistent signal acquisition, hardware instability, and serial communication failures. Despite multiple attempts to troubleshoot and stabilize the module, it was ultimately removed from the system to maintain the project's focus on robust and functional vision-based tracking.

As a result, all tests and experiments documented in this chapter are based solely on relative localization using real-time video analysis and motion response. The robot's behavior is guided entirely by the spatial location of a green object within the camera feed. The sections that follow detail the hardware integration, communication framework, decision-making logic, and a series of test cases used to evaluate system performance. Observations on the robot's movement, tracking capabilities, and encountered limitations are also discussed, along with proposed enhancements for future work.

IV.2 Robot Assembly

The mobile robot platform was assembled using modular components, selected for compatibility and cost-efficiency. The structural and electrical assembly includes:

- ✓ A 4-wheel drive chassis equipped with DC motors for traction.
- ✓ An ESP32-WROOM-32 board responsible for interpreting commands and controlling motors.
- ✓ An ESP32-CAM module, mounted to face forward, handling real-time image capture.
- ✓ A L298N motor driver, connected to the ESP32 for bidirectional motor control.
- ✓ Power sources**:**

-2 × 18500 batteries supply the motors.

-5V/2A power bank supplies the ESP32 boards.

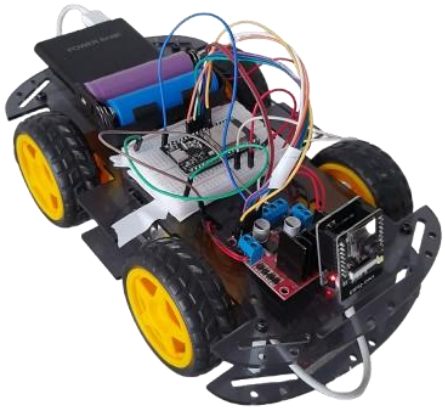


Figure IV. 1: Robot Prototype.

The assembly was designed for easy debugging and reconfiguration during development and testing.

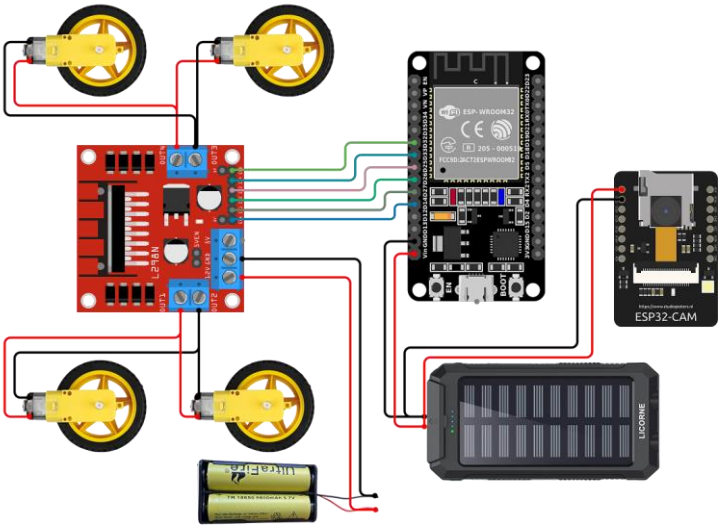


Figure IV. 2: Circuit diagram

IV.2 Communication Between Components

The system uses a wireless UDP-based protocol over Wi-Fi for communication between the vision processor (PC running Python) and the robot (ESP32 board). The communication flow is as follows:

1-ESP32-CAM streams video to a local IP address.

2-The Python script (`color_chasing_robot.py`) running on a PC captures and processes the video.

3-Based on object location, the script sends UDP commands to the ESP32-WROOM-32, such as:

`"f,40"` – move forward at speed 40

`"l,70"` – turn left at speed 70

`"s,0"` – stop

The ESP32 interprets these commands and controls motor movement through the L298N driver.

This communication is unidirectional, simple, and fast, though prone to occasional packet loss, as discussed later.

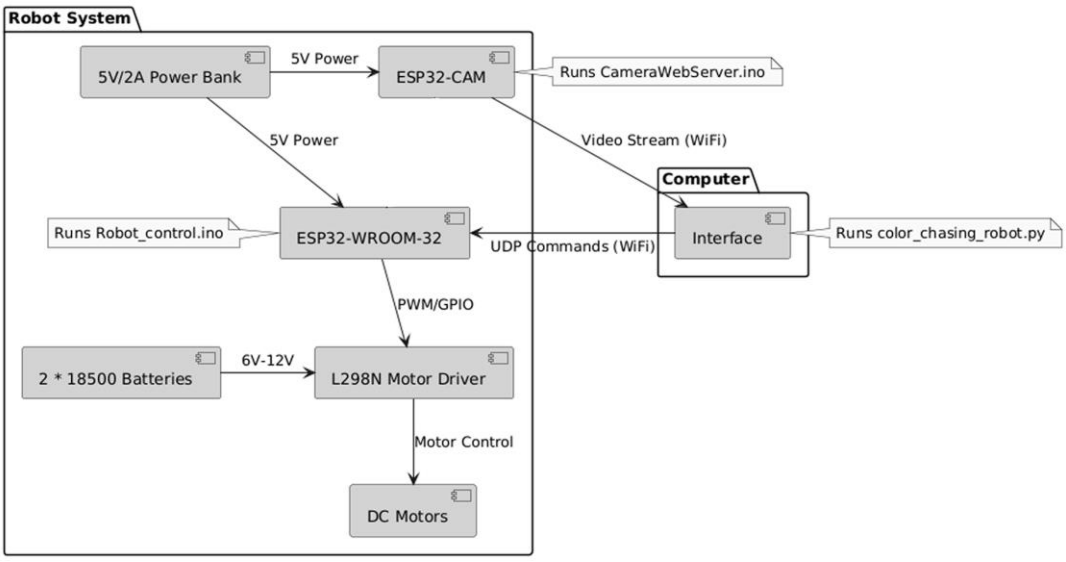


Figure IV. 3: Communication diagram

IV.3 Functional Logic Diagram

The robot's behavior can be summarized in a high-level activity diagram, representing the real-time loop executed by the vision-processing script and the control logic running on the ESP32 board as given in **Figure IV. 4 .**

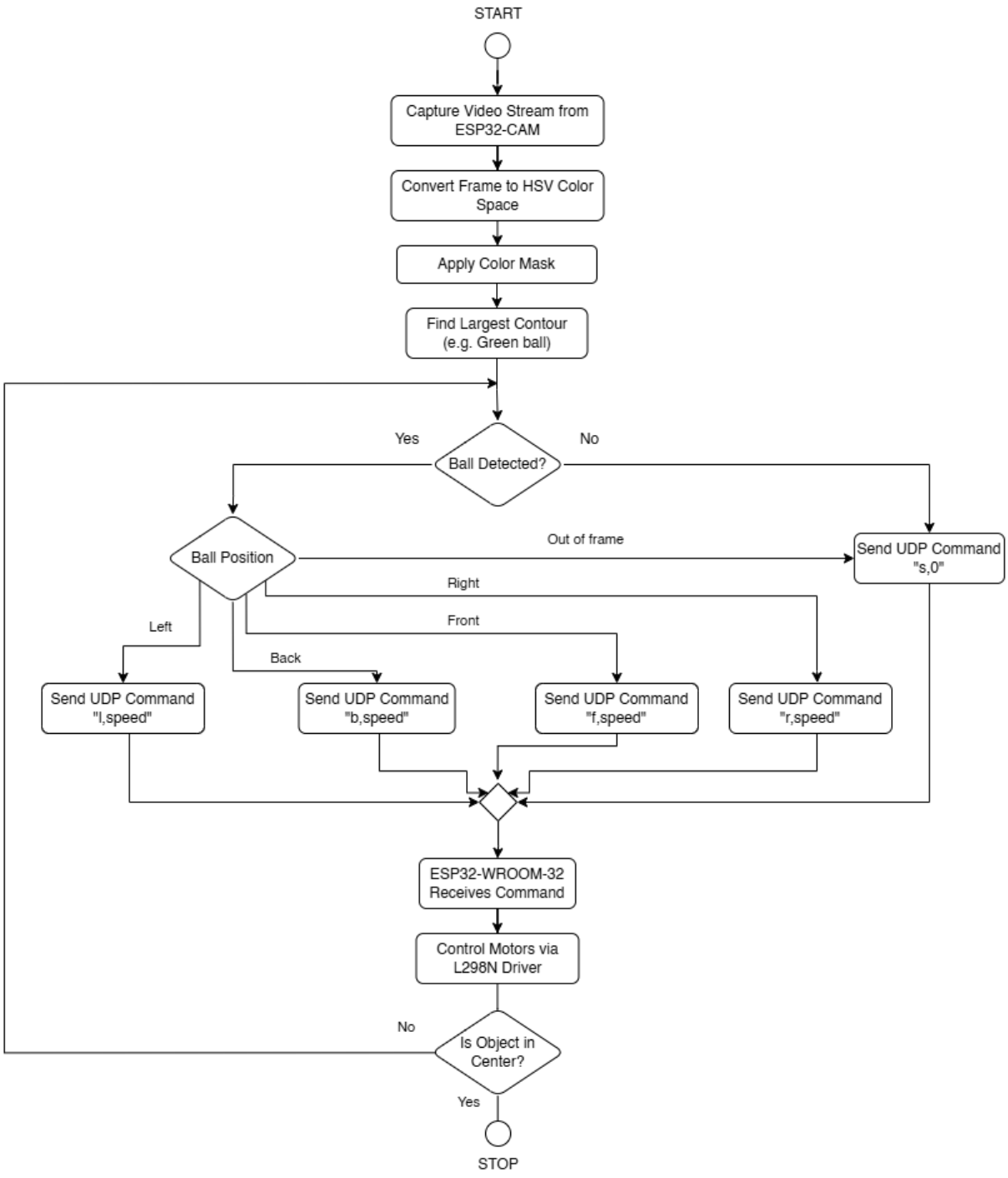


Figure IV. 4: Functional Logic Diagram

IV.3 Test Setup Description

To evaluate the performance of the autonomous robot, a series of indoor tests were conducted in a controlled environment. The test environment was a flat, open space with sufficient lighting. A green ball was used as the visual target (**Figure IV. 5) .**

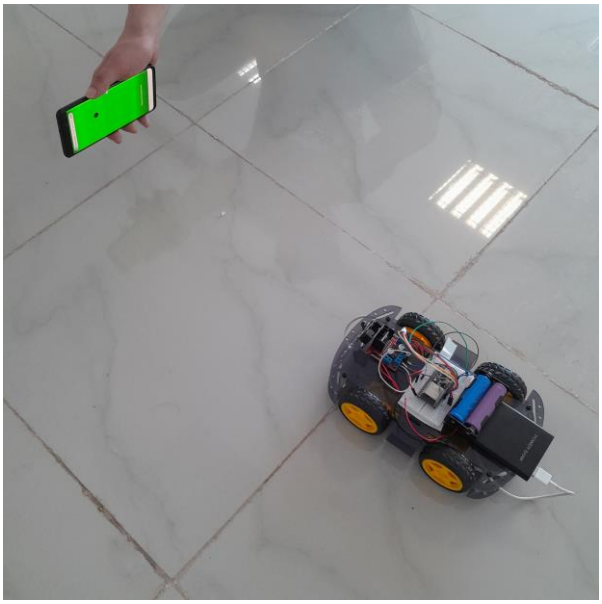


Figure IV. 5: Testing the robot.

First we developed a web interface using Html, Css and Javascript. This web interface was integrated in ESP32 CAM code **(Figure IV. 6) .**

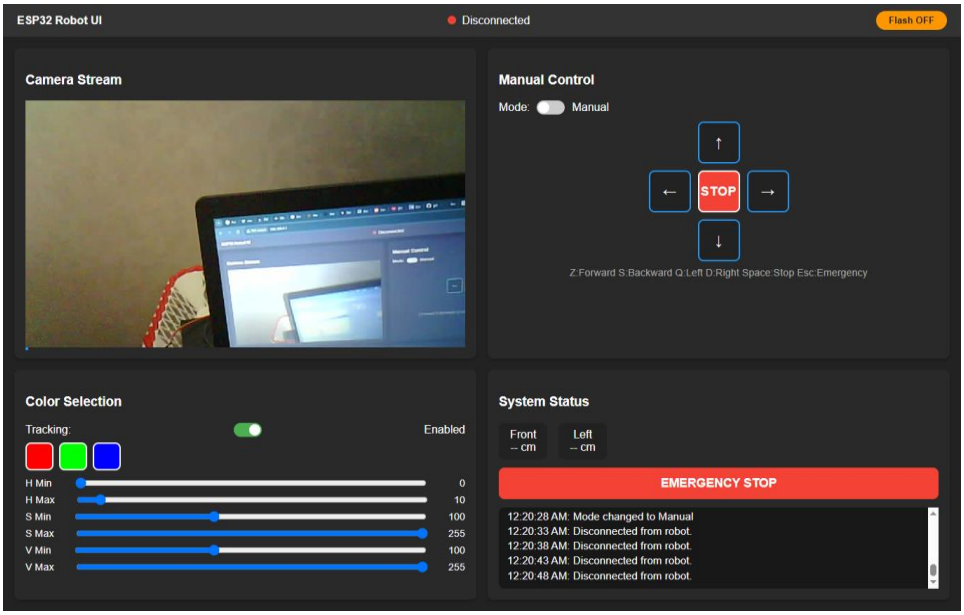


Figure IV. 7: Web interface

Due to the limited memory of the ESP32-CAM (520 KB) we had to use a Python script on a host computer processed the video stream from the ESP32-CAM and issued UDP commands to the ESP32 board for motion control.

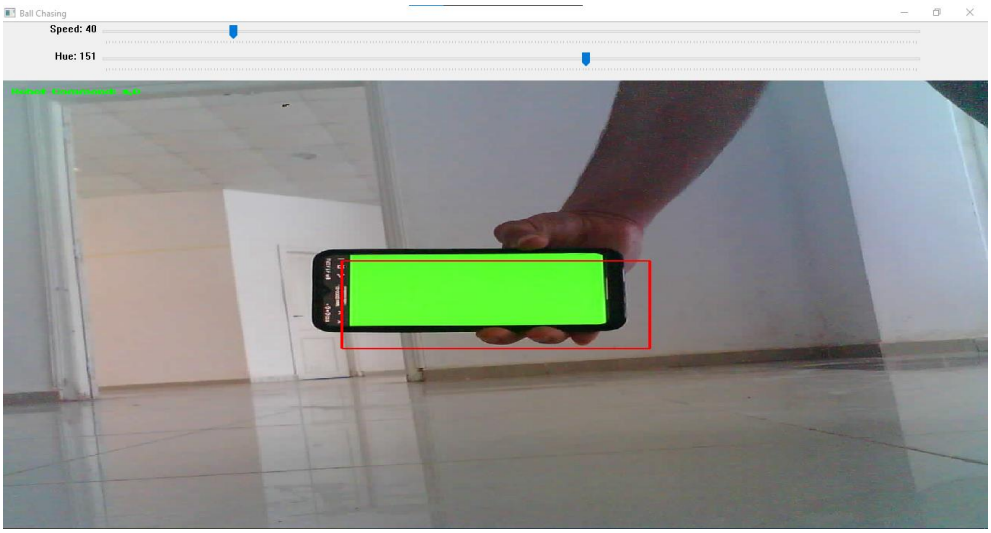


Figure IV. 8: Intrface of python video stream.

IV.3.1 Object Tracking

Object tracking was performed using real-time color segmentation in the HSV (Hue, Saturation, Value) color space. The Python script uses a predefined green range (`lower_green = [40, 40, 40]`, `upper_green = [70, 255, 255]`) to isolate green objects in the frame. The largest detected green contour is assumed to be the target.

Once detected:

- The centroid of the green object is computed using image moments.
- This (x, y) coordinate is used to determine directional control relative to the rectangle's position in the frame.
- A green circle is drawn around the object and its coordinates are displayed in real-time.
- The object's motion trail is visualized using a buffer of the last 10 centroid positions connected by green lines.



Figure IV. 9: Video stream with color mask applied

IV.3.2 Movement Response

The robot's mobility system was tested across multiple scenarios to evaluate its responsiveness to control commands derived from object position. The movement control strategy is based on a central red rectangle (detection zone) superimposed on the video stream. Depending on the detected position of the green object (ball) in relation to this rectangle, one of four possible movement commands is triggered:

a.Move Forward

When the ball is detected above the rectangle.



Figure IV. 10: Tracking the object above the rectangle.

b.Move Left

When the ball is left of the center.

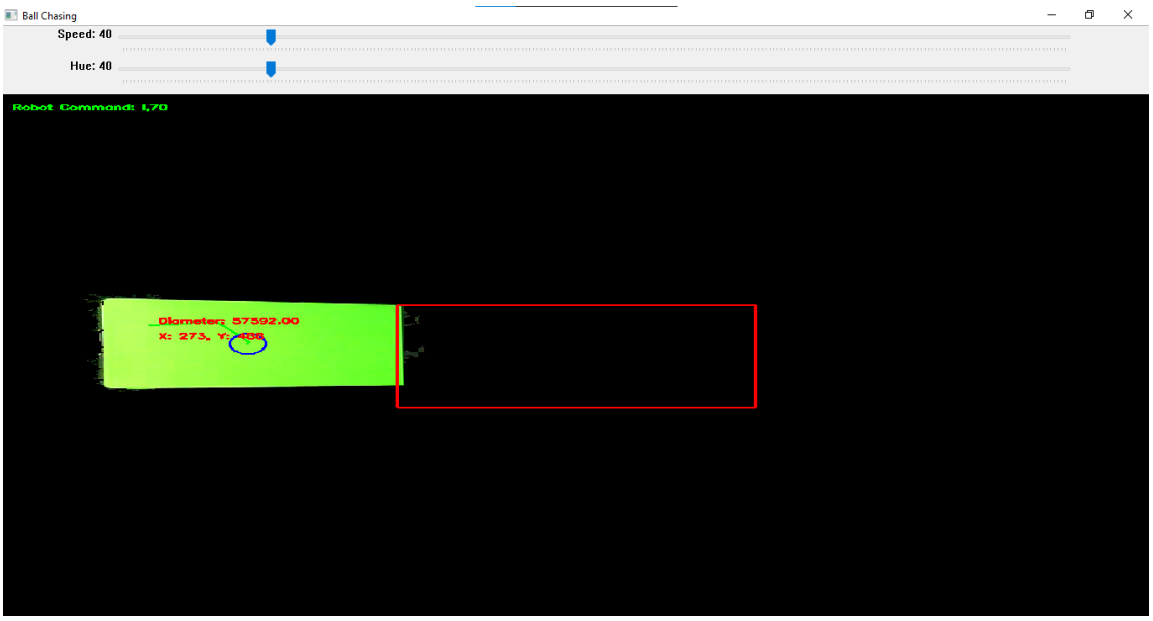


Figure IV. 11: Tracking the object left the center.

c.Move Right

When the ball is right of the center.

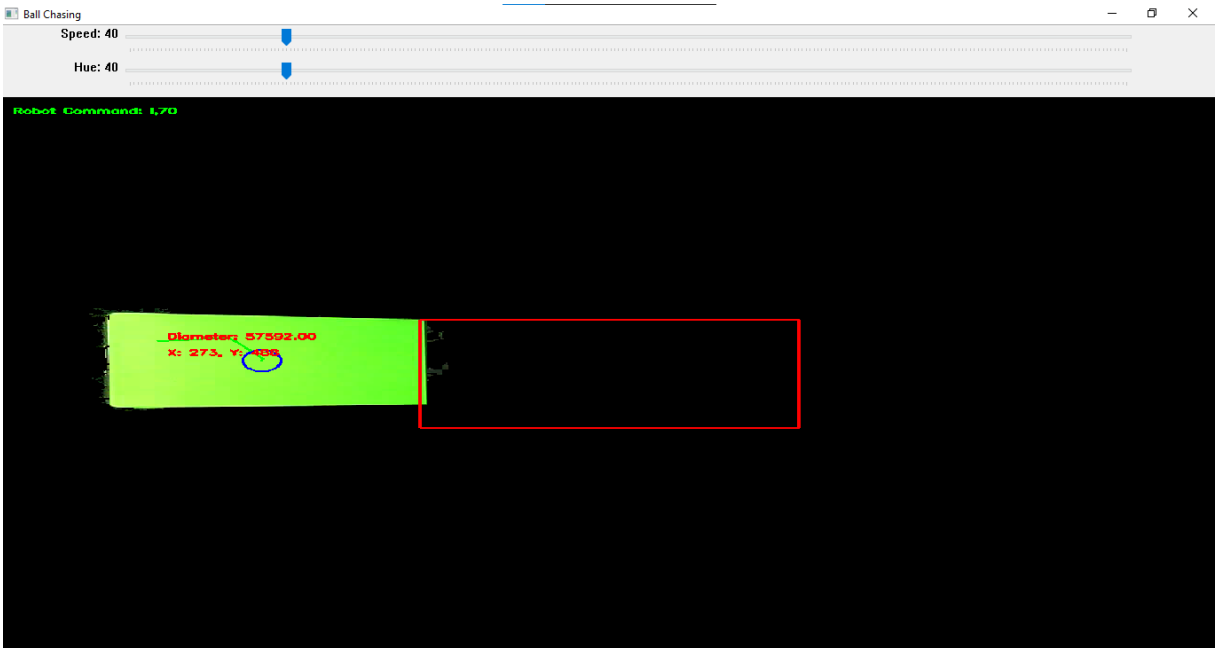


Figure IV. 12: Tracking the object right of the center.

d. Move Backward

When the ball is detected under the rectangle.

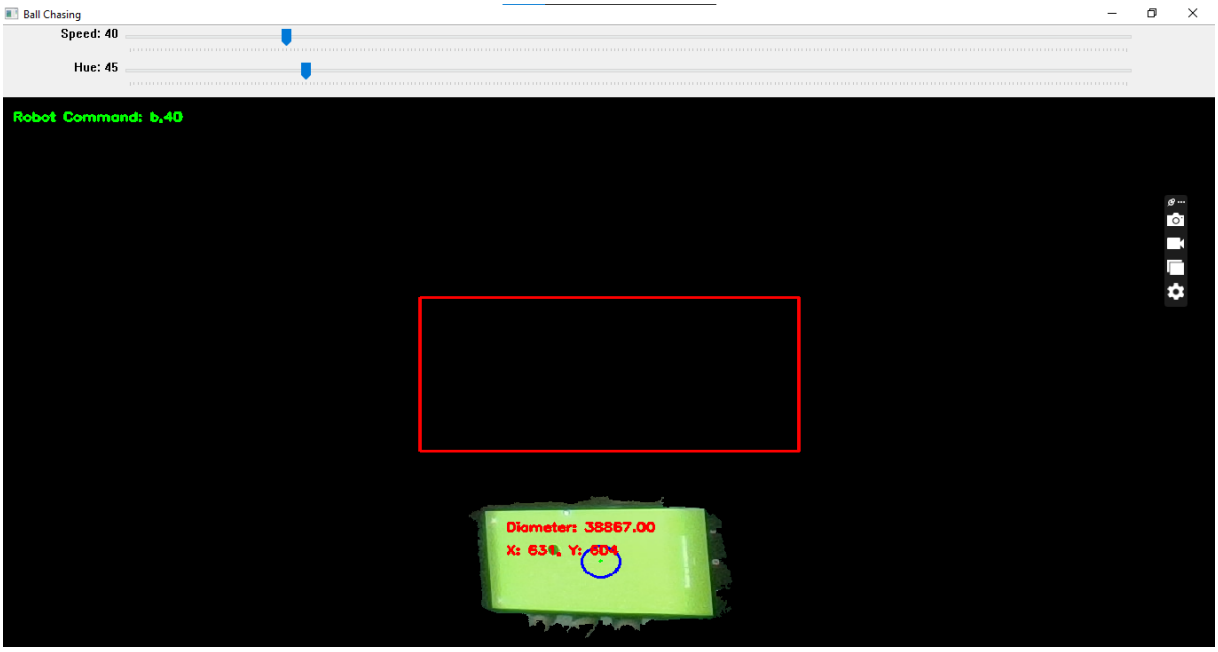


Figure IV. 13: Tracking the object under the rectangle.

The control loop runs in real-time, with a detection-check and command-sending frequency determined by the frame rate (approx. 15–20 FPS). The robot's movement in response to command input was consistent and precise under good Wi-Fi conditions. During testing:

- There was a noticeable improvement in agility with higher speed values (above 50), though this came at the cost of reduced accuracy during direction changes.
- Turning responsiveness (left/right) was optimal when the ball was closer to the edges of the frame, prompting sharper correction.
- The stop condition was accurate when the ball stabilized in the center zone.

Occasional delay (150–200 ms) was observed between command issuance and actuation, mainly due to UDP transmission latency and video processing load.

I.3.3 Observations

Key observations during tracking:

- The robot successfully followed the green ball in various directions and speeds.
- The object trail visualization helped verify tracking consistency.
- Tracking remained stable as long as:
    - The object was fully visible.
    - Ambient lighting conditions remained stable.
- Tracking failure occurred when:
    - The object moved too quickly out of frame.
    - The object color was altered by shadows or reflections.
    - Another green object or background interference entered the frame.

The simplicity of this tracking method made it efficient but vulnerable to color-based interference, making environmental control (consistent lighting and background) essential for reliable operation.

IV.4 Problems and Limitations

1. **ESP32-CAM Memory Constraints**
    - Limited RAM restricted use of advanced vision models.
    - Stream resolution had to be lowered to prevent frame drops.
2. **GPS Module Removal**

- The originally planned GPS module (LS20030) failed during testing and was excluded.
- Absolute localization was abandoned in favor of relative, vision-based tracking.

3. **Lighting Sensitivity**

✓ Color-based tracking was highly sensitive to environmental lighting.
✓ Shadows, reflections, or dim areas sometimes caused false positives or loss of tracking.

4. **UDP Communication Lag**

✓ Occasional packet loss or delay affected robot responsiveness.
✓ Adding packet acknowledgment or retries could improve reliability.

IV.6 Discussion and Analysis

The system successfully demonstrated autonomous tracking behavior using a low-cost platform. Despite hardware limitations, the robot maintained reliable performance in real-time object following using monocular vision and color segmentation.

Key insights

- Simple vision techniques (HSV filtering + centroid tracking) are effective for controlled environments.
- Using UDP allowed quick communication but at the cost of reliability under noisy network conditions.
- Manual tuning (via OpenCV trackbars) gave flexibility in adjusting color thresholds during operation.

Possible Enhancements

- Incorporating PID control for smoother movement transitions.
- Adding fallback behavior when the target is lost (e.g., rotate in place to search).
- Testing with different colored objects to evaluate generalization.

IV.5 Conclusion

The tests validated that a low-cost robot equipped with ESP32 and basic computer vision can perform real-time tracking of a colored object. While limitations were observed, particularly in vision reliability and communication lag, the system performed its core function successfully. This opens up opportunities for expansion, including multiple object tracking or gesture-based control.

General Conclusion

This thesis presented the design, implementation, and testing of a vision-based autonomous mobile robot capable of detecting and following a colored object in real-time. By integrating computer vision with embedded control, the robot was able to interpret visual data from an ESP32-CAM module and respond with appropriate movement commands through an ESP32 microcontroller connected to a motor driver. A series of experimental tests validated the system’s ability to track a green object and respond accurately with directional movements, including forward, backward, left, right, and stop behaviors.

The results demonstrate that even with low-cost hardware and basic image processing algorithms, effective autonomous behavior can be achieved in controlled environments. The system's reliance on monocular vision and UDP communication proved sufficient for object tracking, albeit with some limitations related to lighting variability, object occlusion, and network-induced latency.

One major setback encountered during the project was the failure of the GPS module (LS20031), which was initially intended to provide absolute localization data. The module experienced hardware instability and failed to maintain reliable communication during integration. Despite attempts to troubleshoot the issue, it was determined that continuing with the GPS module would compromise the project timeline and system reliability. Therefore, the module was excluded from the final implementation, and the robot’s navigation strategy was revised to focus entirely on vision-based relative positioning.

In conclusion, the system met its core objective of demonstrating autonomous object tracking using a simple yet effective combination of vision and control technologies. This work serves as a foundation for future improvements, such as incorporating multi-object tracking, improved sensor fusion, or the reintroduction of a functional GPS system for outdoor navigation.

References

[1] LaValle, S. M. , « Planning Algorithms », Cambridge University Press, september 2020.

[2] Siciliano, B., & Khatib, O. . « *Springer Handbook of Robotics* (2nd ed.) », Springer Handbook of Robotics, mars 2016.

[3] Zhang, Z., « Embedded Computer Vision and Robotics: Real-Time Applications with Python », Wiley-IEEE Press, june 2020.

[4] Parra, C., & Gebhart, F. , « A Comparative Study of GPS and Visual Odometry for Mobile Robot Localization », IEEE Access, 9, 76543–76554, 2021.

[5**]** Singh, H., & Kumar, R. , « Low-Cost Autonomous Robots for Smart Applications: Challenges and Perspectives. », Journal of Intelligent & Robotic Systems, 95(3–4), 757–774 2019.

[6] SLIMANE Noureddine, « Localization system for mobile robots», Doctoral thesis in robotics, University of Batna. 2005.

[7] Roland Siegwart et al, « Introduction to Autonomous Mobile Robots », Massachusetts Institute of Technology, 2011.

[8] Bhagwati Charan Patel, G R Sinha and Naveen Goel, « Introduction to sensors », Book of Advances in Modern Sensors, November 2020.

[9] Cyril DROCOURT, « Localization and modeling of the environment of a mobile robot by cooperation of two omnidirectional sensors», doctoral thesis, University of Technology of Compiègne. 2002.

[10] J. Borenstein and L. Feng,′Gyrodometry : a new method for combining data from gyro ;and odometry in mobile robots′. Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnisota, April 1996.

[11] Omar AÏT-AIDER, « Model referenced localization of an indoor mobile robot», Doctoral thesis in robotics, University of Evry. 2002.

[12] Zoulikha BOUHAMATOU, « NAVIGATION VISUELLE D'UN ROBOT MOBILE», Master's thesis, University of Batna, 2015.

[13] R. Boutteau. , , doctoral thesis "Reconstruction tridimensionnelle de l'environnement d'un robot mobile, à partir d'informations de vision omnidirectionnelle, pour la préparation d'interventions", the University of ROUEN", 19 avril 2010.

[14] R. Klette, K. Schluns, A. Koschan, Computer Vision: Threedimensional Data from Images, Springer, Singapore (1998).

[15] P. Favaro and S. Soatto "A Geometric Approach to Shape from Defocus" IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pp. 406-417, March 2005.

[16] Smith, J., « The genesis of the Decca Navigator System », Journal of Navigation n° 68, pp. 219–237, Cambridge University Press, March 2015.

[17] Jean-Luc Cosandier, « General principles of satellite location», avril 2003.

[18] Differential GPS https://en.wikipedia.org/wiki/Differential_GPS, Consulted on 19 mars 2025.

[19] Kessaissia Mohamed Fares & Ferhat Mohamed,« Design of an Intelligent Building Management (IoT) System for Thermal Comfort», Master's thesis, 2023 .

[20] ESP32-CAM - Getting Started & Solving Common Problems https://dronebotworkshop.com/esp32-cam-intro/, Consulted on 21 April 2025.

[21] Datasheet of GPS smart antenna module, LS20030~3. LOCOSYS Technology Inc. https://datasheet4u.com/datasheet/LOCOSYS/LS20031-900733 Consulted on 21 April 2025.

[22] D.H.Titterton and J. L. Weston, Strapdown inertial navigation technology. USA : The Institution of Electrical Engineers, 2004.

[23] 18650 battery https://en.wikipedia.org/wiki/18650_battery#:~:text=An%2018650%20battery%20or%20186 5,giving%20them%20the%20name%2018650. Consulted on 2 mai 2025.

[24] Visualstudio https://code.visualstudio.com/docs/ Consulted on 10 mai.