

# MEASURING ROBOT

Nguyen Thi Linh, Duong Quynh Nhat, Nguyen Trong Toan

Date of Submission: 03-05-2026

Date of Acceptance: 13-05-2026

## I. OVERVIEW OF ARTIFICIAL INTELLIGENCE

### 1.1 Problem statement

In the current era, computers can solve many difficult problems that were previously unsolvable. However, there are still a large number of very interesting problems that require a suitable algorithm to solve them. Among these, problems using artificial intelligence are commonly encountered in practical applications. For example: applying steelmaking simulation methods to find the shortest path for a fire truck, chess problems, or autonomous robots...

An autonomous robot, or mobile robot (often abbreviated as robot), is defined as a robotic vehicle capable of self-movement and autonomous operation (and can be reprogrammed) under automatic control to complete assigned tasks. In theory, the operating environment of an autonomous robot may be on land, in water, in the air, in outer space, or a combination of these. The surface terrain on which the robot moves may be flat or uneven. This project focuses on autonomous maze-solving robots. Mobile robots are an important component of robotics, a field that is indispensable in automation. Together with the rapid development of automation systems, autonomous robots have become increasingly refined and have demonstrated growing benefits in industry and daily life. One issue of great interest in autonomous robot research is how a robot can determine its current position and move to a specified destination while automatically avoiding obstacles along the path. Therefore, successful implementation of this project will open a new approach and contribute to promoting the application of robots in everyday life and in research and development.

### 1.2 Overview of artificial intelligence

What is artificial intelligence?

Artificial intelligence is the scientific field that studies methods for creating machine intelligence that resembles human intelligence.

Some representative definitions of artificial intelligence are:

- A system that can think like humans.

- A system that can act like humans.

Để hệ thống mà biết suy nghĩ và hành động như con người thì hệ thống đó phải

be equipped with perception, knowledge, automated reasoning, learning, vision, and movement capabilities similar to those of humans.

In general, the human problem-solving process is expressed through four basic operations:

- Determine the set of goals.
- Collect facts and inference rules.
- Focusing mechanism.
- Inference engine.

Thus, machine intelligence is the machine's problem-solving capability with the following characteristics:

- Acting like humans.
- Thinking like humans.
- Learning like humans.
- Processing information like humans.
- Acting and thinking on a logical and accurate basis.

### 1.3 Overview of Arduino

Arduino is an open-source logic-programming platform consisting of both hardware and software. It can interact with hardware devices such as sensors, motors, lights, and other devices. Arduino boards are diverse and easy to use for many different purposes. In addition, Arduino is compatible with many expansion boards, making assembly and use easy. The working environment is based on C/C++, and the structure of a simple program consists of two basic functions: `setup()` and `loop()`. Its rich and easy-to-use library system has made Arduino increasingly popular. By using Arduino boards, users can create electronic devices that serve practical life needs.

Due to its compact size and strong processing capability, Arduino is chosen as the central processor of many types of robots, especially mobile robots.

### 1.4 Overview of STEM robotics

#### What is STEM robotics?

STEM is the key to innovation and economic growth in a connected world where people are surrounded by technology and creativity.

So, what is STEM robotics? In this project, I would like to share some content related to STEM robotics.

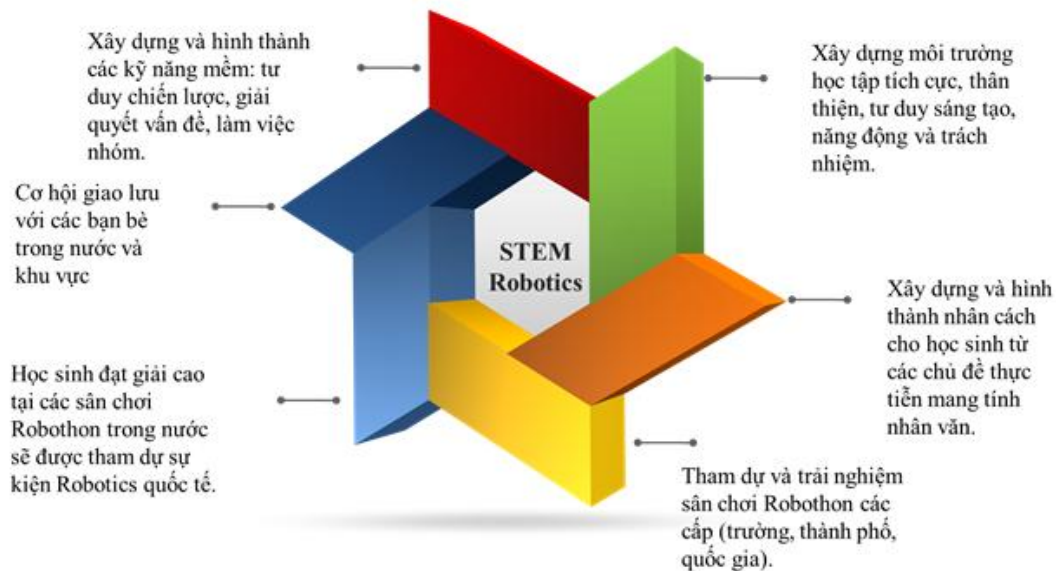
The STEM-Robotics program was developed by Robomatter under the Robotics Academy of Carnegie Mellon University (USA). Robomatter is a pioneering organization in researching the application of robots in learning environments to motivate students and teachers in teaching and learning the fields of technology, engineering, and mathematics. The lessons in the STEM-Robotics program are built around practical problem-based themes. Students then use robots and programming tools for

simulation. This allows students to learn integrated knowledge from the four fields of Technology - Engineering - Science - Mathematics.

With the 'Hands-on, minds-on' approach, STEM-Robotics learning activities stimulate students' imagination and creativity. This approach helps students learn in an enjoyable way by designing, building, and programming robots to operate, while also relating them to các hoạt động trong thế giới tự nhiên với học sinh.

The program is also designed to suit students with little or no programming experience who wish to learn basic programming knowledge.

### 1.5 Course benefits



Hình 1.1: Lợi ích của STEM robotics

#### 1.5.1 Educational values provided by STEM Robotics

Build and develop soft skills: strategic thinking, problem solving, and teamwork. Create opportunities for students to interact with peers inside and outside the country within the STEM learning community in general.

Build a positive and friendly learning environment that fosters creative thinking, dynamism, and responsibility. Develop science and engineering skills including asking questions and solving problems, investigating, analyzing and processing data, reasoning based on known information, and collecting, evaluating, and exchanging information, all through designing, creating, and assembling robot models.

Encourage students to explore science, computer skills, and programming through simple and intuitive programming software, especially the

skills to use engineering tools through accompanying technical software.

Participate in and experience major competitions such as Robothon and WRO. Create competitiveness and opportunities for exchange with domestic and international peers in the STEM learning community through international Robothon and WRO competitions.

### 1.6 Introduction to the line-following robot

#### 1.6.1 General introduction

Today, automatic control has become an indispensable human need. With the emergence of many theories and practical applications in daily life, automatic control is gradually shaping our lives. People are trying to create robots capable of working in place of humans, and we often encounter robots in automated industrial production lines or household service robots. In addition, Robocon competitions are held annually both domestically

and internationally to stimulate creative ideas, and Thai Nguyen University of Technology has been developing many excellent and useful robot competitions for students.

That is the premise for creating robots capable of practical application. For that very reason, we carried out this line-following robot control project, which is also the initial step in robot programming so that robots can perform more advanced tasks later.

Our automatic line-following and obstacle-avoiding robot vehicle uses Arduino programming to operate in places where it is difficult for humans to work, and it can be controlled automatically or via Bluetooth connection.

### 1.6.2 Functions and tasks

It can search for paths in narrow spaces where humans find it difficult to move and work, travel along a specific route predefined by the programmer, automatically avoid obstacles, and return to the correct previous path when an error occurs.

### 1.6.3 Conclusion

STEM Robotics is a highly practical and application-oriented subject that is very important in building foundational knowledge for school and engineering students. Even with little programming experience, students can still study and implement it; ultimately, they can understand operating principles, apply sensor systems, and build an intelligent line-following robot capable of recognizing and following colors.

## II. METHODS FOR CONTROLLING WIRELESS REMOTE ROBOTS

In this chapter, the authors present remote control methods for robots. Today, the robotics field is developing strongly with wireless technology and remote monitoring and control.

Robot control methods mainly include three methods:

- Control via Bluetooth.
- Control via RF.
- Controlled via Wi-Fi.

### 2.1 Bluetooth-based control

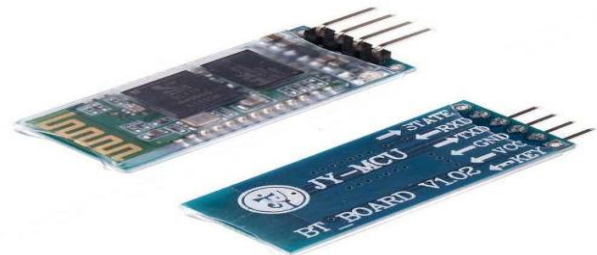
#### 2.1.1 Introduction to Bluetooth technology

Bluetooth is a technology that allows wireless communication between devices. It's an

electronic standard, meaning that manufacturers wanting this feature in their products must adhere to the standard's requirements. These technical standards ensure that devices can recognize and interact with each other using Bluetooth technology.

Today, most manufacturers produce devices that use Bluetooth technology. These devices include mobile phones, computers, and personal digital assistants (PDAs).

#### 2.1.2 Characteristics of Bluetooth technology



Hình 2.1: Module giao tiếp bluetooth

#### Advantages:

- Low power consumption.
- Applicable to many types of devices including handheld devices and mobile phones.
- Cost is continuously decreasing.
- Communication distance between two connected devices can reach up to 100 m.
- Bluetooth uses the 2.4 GHz band, and the data rate can reach up to 1 Mbps without devices needing direct line of sight.
- Easy application development: Bluetooth connects one application to another via the Bluetooth standard, making it independent of hardware and operating system platforms.
- High compatibility, supported by many hardware and software manufacturers.

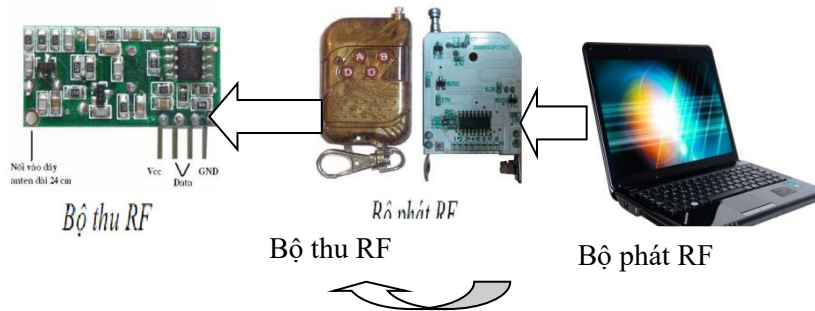
#### Disadvantages:

- Shorter connection distance than some other wireless technologies.
- Security is still relatively low.

### 2.2 RF-based control

This is one of the earliest forms of remote control and still plays an important role in life today. It transmits radio waves corresponding to binary commands. The radio receiver on the controlled device receives the signal and decodes it.

### 2.2.1 Robot control diagram



Hình 2.2: Sơ đồ điều khiển robot qua sóng RF thông thường

### 2.2.2 Operating principle

#### Transmitter

- Uses PT2262 Chip
- Signal frequency: 315MHz
- Transmission range: 50-100m

#### Receiver:

Uses PT2272 Chip

- Operating voltage: 5V
  - Current consumption: 2.5mA
  - Reception range: 50-150m
  - The RF transmitter module has 4 push buttons.
- When a button is pressed, one signal is transmitted at one time.

- Receiver module

- The RF transmitter module has 4 push buttons.
- When a button is pressed, one signal is transmitted at one time.
- Modul nhận tín hiệu RF là một bộ thu chỉ nhận được duy nhất 1 tín hiệu tại 1 thời điểm.
- => **Therefore, it cannot output signals**

**simultaneously or toggle one output on/off using the same button.**

### 2.2.3 Conclusion

#### \* Advantages:

- Can transmit over long distances.
- Can transmit through walls and glass.
- Relatively low cost.

#### \* Disadvantages:

- Susceptible to external interference when many devices use different frequencies.
- Relatively low transmitted and received data volume, limited mostly to bits.
- Low security.

### 2.3 Control via Wi-Fi and Ethernet network

Wi-Fi (Wireless Fidelity) is a wireless Internet access system using radio waves, similar to mobile phones, television, and radio, and requires no cables. Wi-Fi waves use frequencies of 2.4 GHz or up to 5 GHz, higher than conventional radio standards, so they can transmit high-capacity data such as video and images.



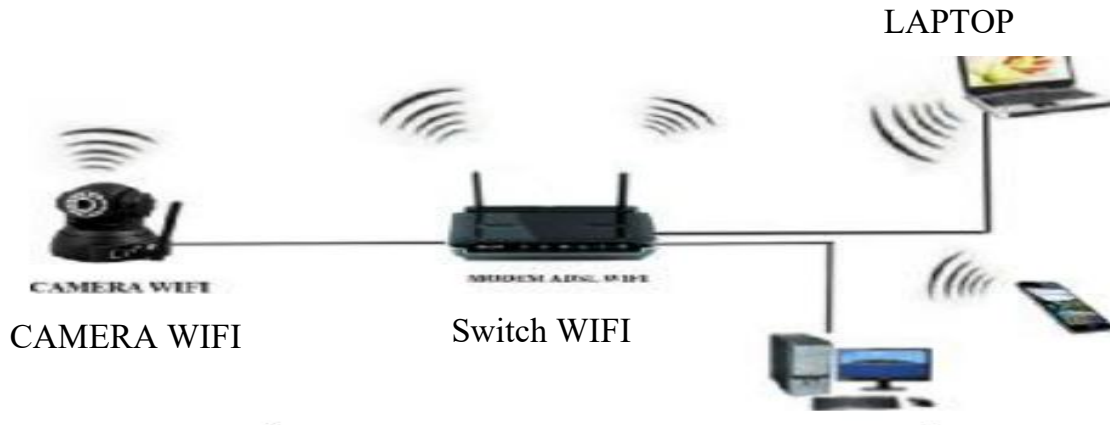
Hình 2.3: Module wifi esp01



Hình 2.4: Router Wifi

There are currently five common Wi-Fi standards: 802.11a/b/g/n/ac.

### 2.3.1 Cấu tạo và nguyên lý hoạt động của hệ thống



Hình 2.5: Sơ đồ quan sát hình ảnh từ xa thu về từ Robot

#### Nguyên lý hoạt động

According to the above diagram, these devices must be connected to each other via Wi-Fi and be in the same LAN (Local Area Network). The microcontroller is connected to the Wi-Fi modem via RJ45.

In which:

- IP camera is a camera using the TCP/IP protocol. Each camera is assigned an IP address, which must match the LAN in use. During operation, the camera captures images from outside and then transmits this image data back to the user for viewing on a computer. The camera uses an RJ45 port and Ethernet standard for data transmission.

- The Wi-Fi router is used to exchange data with the camera and monitoring devices such as laptops, desktop computers, and mobile phones.

- Users on laptops and similar devices install software compatible with the camera in use, connect those devices to the Wi-Fi network used by the camera, and then monitor the images from the camera.

#### Robot control section:

- A computer connected to the Wi-Fi switch is programmed with a control interface that sends commands to an IP address in the switch.

- The ENC28J60 Ethernet module is connected to the switch via the local area network port through an RJ45 connector. The microcontroller is programmed with an IP address suitable for the Wi-Fi switch.

- When the computer sends commands, the microcontroller receives and analyzes them and then

controls the robot.

### 2.3.2 Advantages and disadvantages of control via Wi-Fi and Ethernet network

#### Advantages:

- Widely available in cafes, homes, schools, and public places.
- Wide coverage range.
- Stable transmission signal, not easily interfered with by other devices.
- One of the most common technologies today for data transmission and reception.
- Large data throughput and high speed.
- High security and flexibility.

Ethernet is also highly rated for its security and strong control capability on each transmission line, making unauthorized access difficult. The system can maintain control rights, allowing only selected machines to connect to the network.

As a result, unauthorized access becomes difficult. The system can maintain control over which machines are allowed to connect to the network, thereby preventing unwanted security violations. With Wi-Fi, on the other hand, access is much easier, and therefore the risks are also greater.

#### Disadvantages:

- High initial investment cost.
- Programming is complex, with packet-based data processing.
- **Mobility is limited because it cannot easily move to many locations.**

## 2.4 Conclusion

Among the current wireless remote robot control methods, the authors chose Bluetooth-based control. With its low cost and easy replacement when failures occur, Bluetooth technology, being radio-frequency-based, allows any device with this integrated technology to communicate with other devices within a certain range. Therefore, Bluetooth is a reasonable solution for the proposed problem.

## III. GENERAL INTRODUCTION TO THE LINE-FOLLOWING ROBOT

### 3.1 Introduction to the line-following robot vehicle

### 3.2 Structure of the line-following robot

#### 3.2.1 Arduino Mega 2560

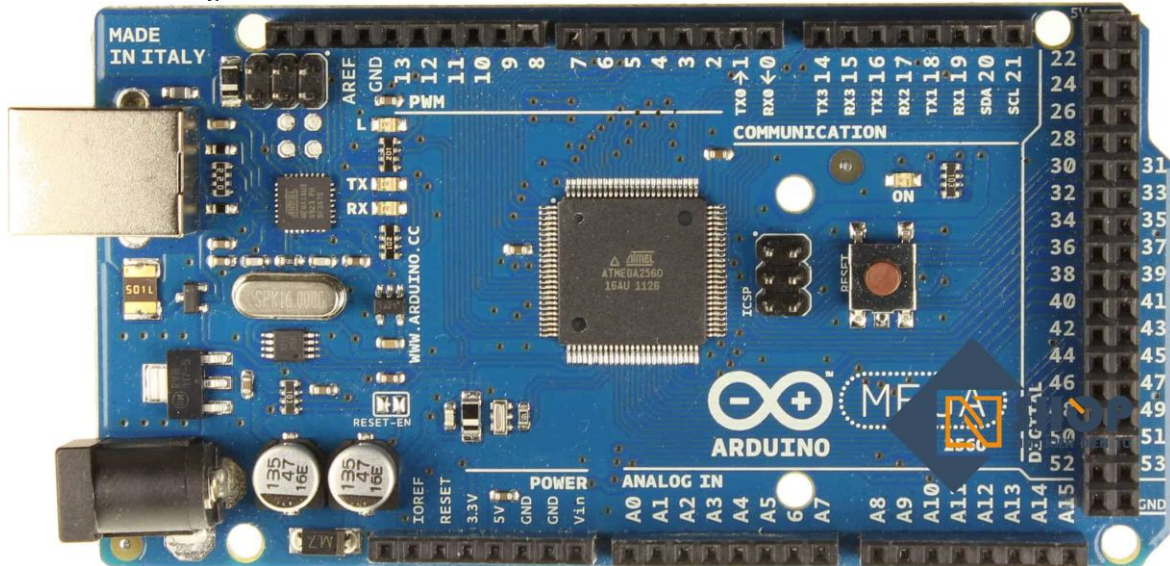


Figure 3.1: Arduino Mega 2560

\*Why use the Arduino Mega 2560:

### 3.1.1 Introduction

This is a robot that automatically follows a line and avoids obstacles, using Arduino programming to operate in places where humans find it difficult to work, and it can be controlled automatically or via Bluetooth connection.

### 3.1.2 Functions and tasks.

It can navigate narrow spaces where it's difficult for humans to move and work, following a specific route programmed by the user, automatically avoiding obstacles, and returning to the original path if it makes a mistake.

When cheaper boards are available, why use the Arduino Mega? The main reason behind this is the additional features built into this board. The first is its large I/O system design with 16 analog probes and 54 digital probes available, supporting USART and other communication modes. Second, it has RTC and other features such as an analog comparator, an advanced timer, interrupts for the controller wake-up mechanism for greater power savings, and a fast 16 MHz crystal clock for 16 MIBS. It has more than 5 pins for Vcc and Gnd for connecting other devices to the Arduino Mega.

Other features include JTAG support for programming, debugging, and troubleshooting. With large FLASH memory and SRAM, this board can handle large system programs with ease. It is also compatible with various board types such as high-level (5V) or low-level (3.3V) signals with I/O ref. Brownout and the monitoring mechanism help make the system more reliable and robust. It supports ICSP as well as USB microcontroller programming with a PC.

The Arduino Mega 2560 is a replacement for the older Arduino Mega and therefore, generally, it will be referred to without the '2560' extension. Due to its many pins, it is not usually used for routine projects but you can find them in more complex ones such as Radon detectors, 3D printers, temperature sensors, IoT applications, real-time data monitoring applications.

\*Basic characteristics:

The Arduino Mega2560 is a board with many outstanding features. The first feature is its large I/O system design with 16 analog converters and 54 digital converters supporting UART and other communication modes. Secondly, the Arduino Mega 2560 has built-in RTC and other features such as comparators, timers, and interrupts for operation control, power saving, and faster speeds with a 16 MHz crystal oscillator.

Other features include JTAG support for programming, debugging, and troubleshooting. With large FLASH memory and SRAM, this board can easily handle large system programs. It is also compatible with various board types such as high-level (5V) or low-level (3.3V) signals with I/O pins. Brownout and watchdog make the system more reliable and robust. It supports ICSP as well as USB microcontroller programming with a PC. The Arduino Mega 2560 is a replacement for the older Arduino Mega.

The Arduino Mega 2560 also includes additional features such as a comparator, interrupts, power saving mode, temperature sensor, RTC, and many more.

**3.2.2 Pin diagram of Arduino Mega 2560:**

Arduino Mega	Features and specifications
Microcontroller	AVR ATmega 2560 (8bit)
Supply voltage	7-12V
Number of digital I/O pins	54
Number of analog I/O pins	16
Clock frequency	16 MHz ( nhà sản xuất cài đặt là 1MHz)
Flash memory	128 KB
SRAM	8 KB
Communication	USB (Lập trình với ATmega 8), ICSP (lập trình), SPI, I2C và USART
Timers	2 (8bit) + 4 (16bit) = 6 Timer
PWM	12 (2-16 bit)
ADC	16 (10 bit)
USART	4
VIN	Supply voltage (7-12 V)

GND	Ground pin
5 V supply	For powering external hardware devices
3.3 V supply	For external low-voltage hardware devices

Arduino Mega 2560 cũng được đóng gói với các tính năng bổ sung như Bộ so sánh tương tự, Ngắt ngoài & Ngắt phần mềm, Chế độ tiết kiệm điện, Cảm biến nhiệt độ sẵn có, RTC và hơn thế nữa.

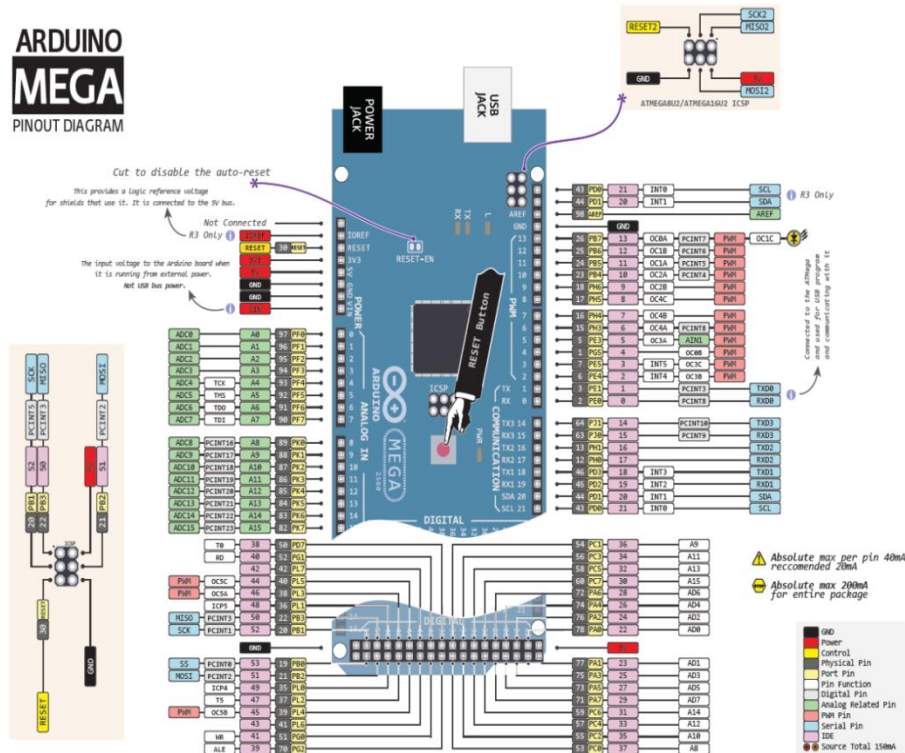


Figure 3.2: Arduino Mega 2560 pin diagram

### 3.2.3 Robot frame design

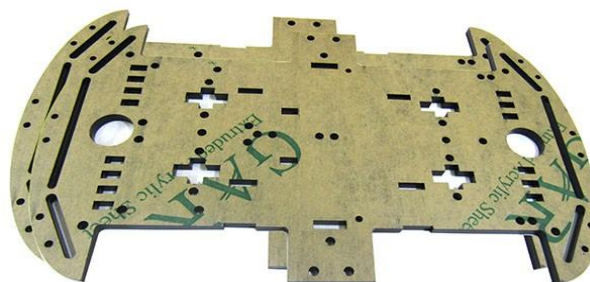


Figure 3.3: Robot frame

- The robot frame is manufactured with dimensions of 300 x 200 mm.
- The main frame is made of mica sheet using a 3D-printing method.

\* Advantages

- Light weight
- Easy to manufacture
- Low cost

\* Disadvantages

- Limited durability, easily broken during operation

### 3.2.4 Wheels



*Figure 3.4: Front and rear wheels of the robot*

- The two rear wheels use ABS plastic wheels coated with rubber to ensure friction on smooth surfaces.
- The front wheel is a self-guiding omnidirectional wheel.
- The robot's speed and direction are controlled through the two rear wheels.

### 3.2.5 Selecting the robot motor



*Figure 3.5: Robot motor*

*To satisfy the motion requirements during robot operation, the authors selected a motor with the following technical specifications:*

- Model: JGA25-370
- Operating voltage: 3-24 V
- Best operating voltage: 6-9 V
- Power: 2 W
- Weight: 100 g

### 3.2.6 Robot power supply

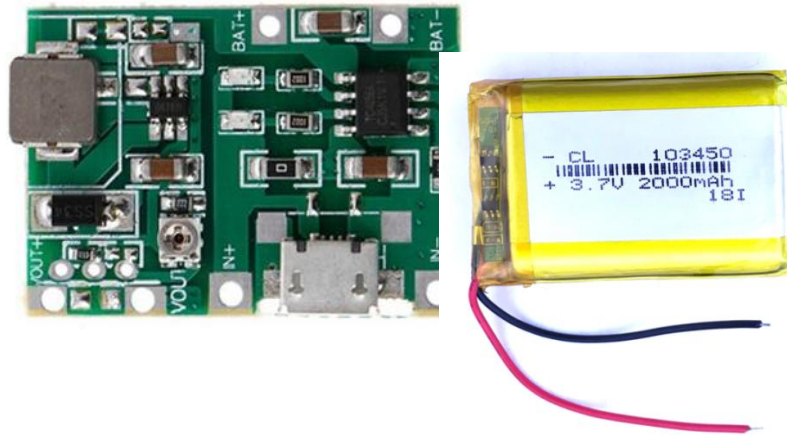


Figure 3.6: Robot power supply

A 3.7 V 3 A lithium battery is used together with a charging circuit and a boost circuit to raise the voltage to 9 V.

### 3.2.7 L298 motor driver circuit

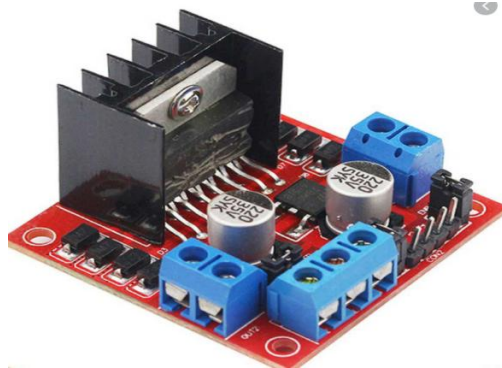


Figure 3.7: Control circuit

The L298 motor control circuit allows easy control of the speed and direction of a DC motor. In addition, the L298 module can also control one bipolar stepper motor. The L298 H-bridge motor circuit supports voltages from 5 V to 35 V.

The L298 motor control circuit integrates a 7805 voltage regulator IC to create a 5 V supply for other devices.

The L298 DC motor driver can control 2 DC motors with a maximum current of 2 A per motor. The circuit integrates protection diodes and a 7805 regulator IC to provide 5 VDC for other modules (this 5 V should only be used when the supply voltage is below 12 VDC).

- Dimensions: 43 x 43 x 27 mm

The L298 DC motor driver is easy to use, low-cost, easy to install, and is an optimal choice in its price range.

#### Technical specifications

- Main IC: L298 - Dual Full Bridge Driver
- Input voltage: 5~30 VDC
- Maximum power: 25 W per bridge (note: power = current x voltage, so the higher the supply voltage, the lower the current; rated power is 25 W).
- Maximum current for each H-bridge: 2 A
- Logic voltage level: Low -0.3 V~1.5 V, High: 2.3 V~Vss

### 3.2.8 Infrared sensor

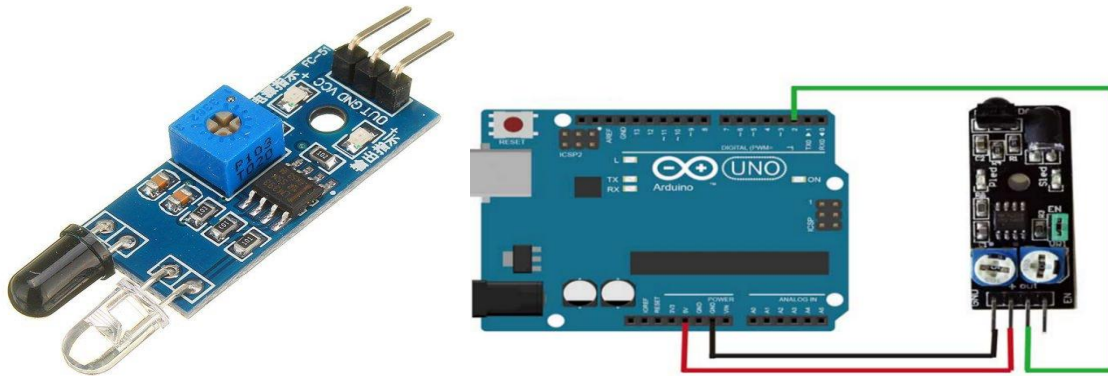


Figure 3.8: Infrared sensor

The IR Infrared Obstacle Avoidance sensor is used to detect obstacles using infrared light. It is simple to use, with a potentiometer to adjust the sensing distance. Its digital output makes interfacing and programming with microcontrollers easy, and it is suitable for obstacle-avoiding robots, anti-theft alarms, automatic door models, and similar applications.

#### Technical specifications:

- Operating voltage: 3.3~5 VDC
- Obstacle detection using infrared light.
- Output: Digital TTL
- Integrated potentiometer for adjusting sensing distance.
- Dimensions: 3.2 x 1.4 cm

### 3.2.9 Wireless PS2 remote receiver module



Figure 3.9: PS2-to-UART conversion circuit

#### Technical specifications of the conversion circuit:

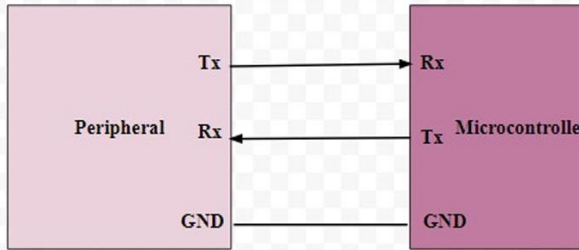
- Receives data from the PS2 controller
- PS2-to-UART conversion circuit
- Communication standard: UART
- UART connection to microcontroller
- Supply voltage: 5 VDC
- 3.3 VDC interface; do not connect power in reverse polarity.

#### What is UART communication?

- UART stands for Universal Asynchronous Receiver-Transmitter. It is an integrated circuit used for serial data transmission between computers and peripheral devices.

- UART's main function is serial data transmission. In UART, communication between

two devices can be carried out by serial data communication and parallel data communication.



UART Communication

- Serial data communication means data can be transmitted over a cable or wire in bit-by-bit form and requires only two wires. It requires very few circuits or wires. This communication method is very useful in interfacing circuits compared with parallel communication.

- Advantages and disadvantages of UART include:

- It only needs two wires for data transmission
- A CLK signal is not required.
- It includes a parity bit for error checking
- The data packet arrangement can be modified because both sides are arranged accordingly
- The maximum data frame size is 9 bits

- It does not include certain subsystems (or)
- UART baud rates must be within 10% of each other

- Applications of UART:

UART thường được sử dụng trong các bộ vi điều khiển cho các yêu cầu chính xác và chúng cũng có sẵn trong các thiết bị liên lạc khác nhau như giao tiếp không dây, thiết bị GPS, mô-đun Bluetooth và nhiều ứng dụng khác.

Các tiêu chuẩn truyền thông như RS422 & TIA được sử dụng trong UART ngoại trừ RS232. Thông thường, UART là một IC riêng được sử dụng trong giao tiếp nối tiếp UART.

#### IV. PROGRAMMING THE AUTOMATIC LINE-FOLLOWING ROBOT VEHICLE

##### 4.1 Introduction to the line-following robot

A line-following robot is a type of mobile robot capable of following a line whose color is distinct from the environment. The applications of this type of robot are mainly in automatic goods transportation in factories, seaports, defense and security, and aerospace. In practice, there are many forms of line-following vehicles. The simplest type commonly known to students is a vehicle with two drive wheels, one omnidirectional wheel, and one line-sensing circuit.

##### 4.1.1 Software design

###### Algorithm flowchart

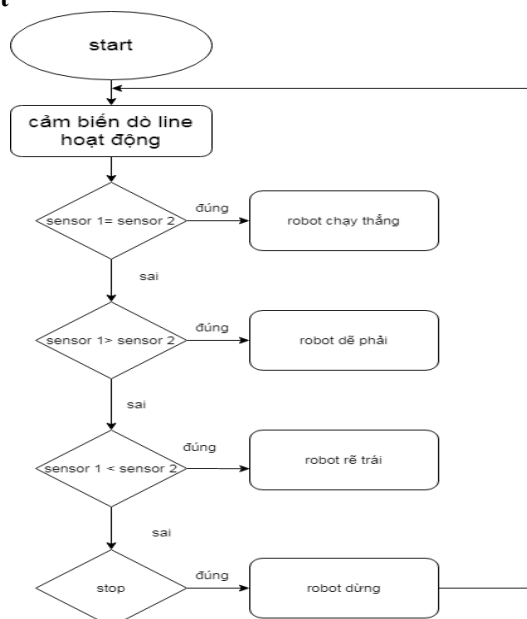


Figure 4.1: Algorithm flowchart

Explanation of the algorithm flowchart

- Initially, when the robot starts, if sensor 1 and sensor 2 are equal, the robot moves straight ahead.
- If sensor 1 > sensor 2, the robot turns right to return to the original lane.

- If sensor 1 < sensor 2, the robot turns left to return to the original lane.
- If neither of the above cases applies, the robot stops.

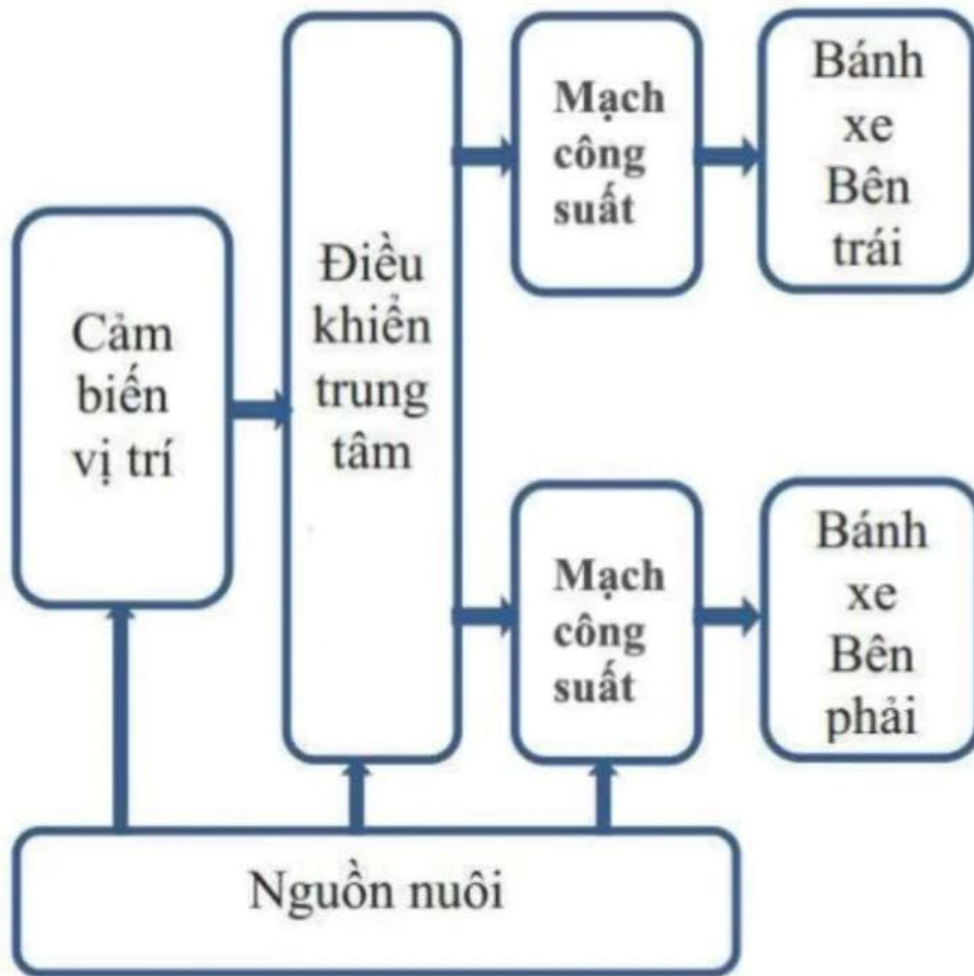


Figure 4.1: Basic block diagram of the line-following vehicle

### 4.1.2 Principle circuit

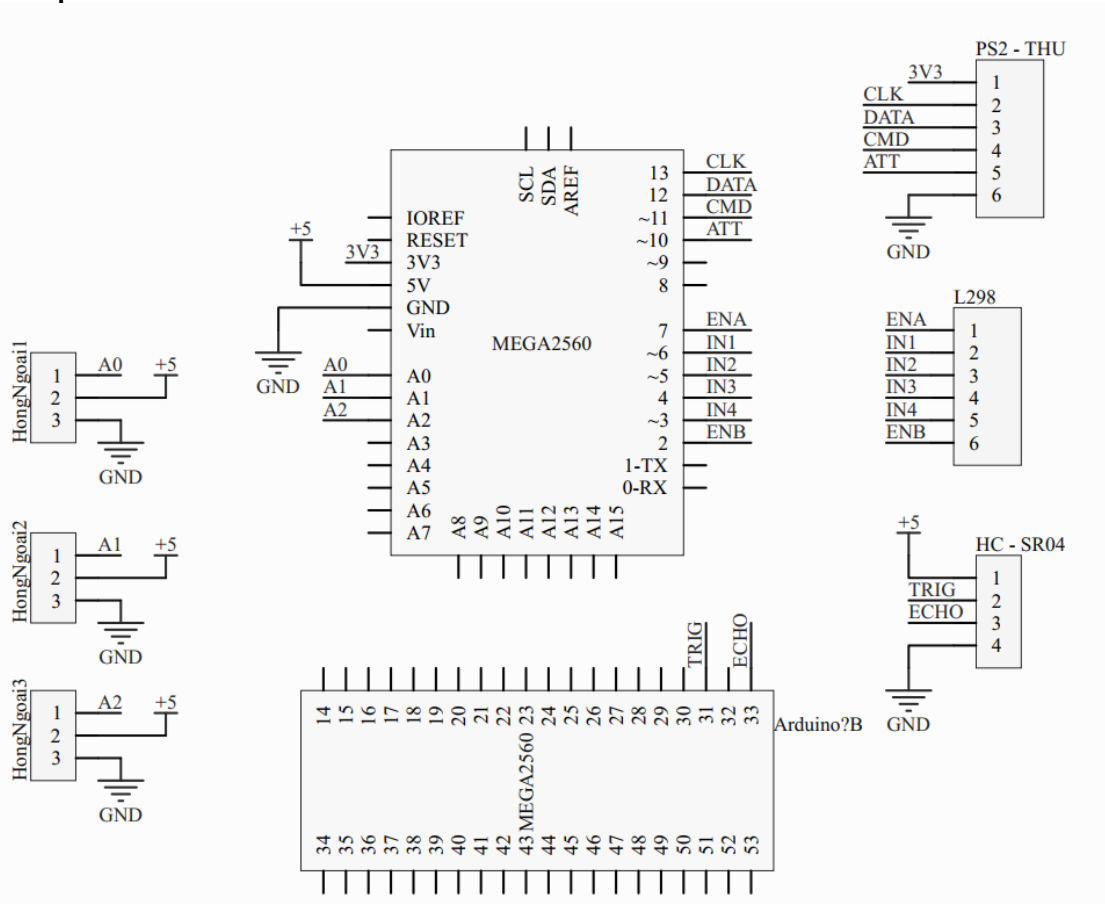


Figure 4.2: Principle diagram

### 4.1.3 Operating Principle

The control circuit uses a 12 VDC supply to power the relay. When voltage is applied, the power circuit stabilizes it to 5 VDC to supply the Arduino Mega2560 and provides 3.3 V for the controller signal.

➤ When control signals are received from the PS2 controller, the Arduino Mega2560 receives them.

Each button on the PS2 controller corresponds to specific encoded and decoded data. Based on the received data, the Arduino Mega2560 sends the corresponding control commands to the driver circuit.

➤ The received data are controlled and programmed according to the following table:

No.	PS2 controller	Arduino Uno R3
1	PSB PAD UP	Control motor forward motion
2	PSB PAD DOWN	Control motor reverse motion
3	PSB PAD LEFT	Control motor to the left
4	PSB PAD RIGHT	Control motor to the right
5	PSB GREEN	Automatic control
7	PSB BLUE	Manual control
8	PSB PINK	Control motor forward-left motion
10	PSB L2	Decrease motor speed
11	PSB_L1	Điều khiển tăng tốc độ động cơ

12	PSB_R1	Điều khiển giảm tốc độ động cơ
13	PSB_R2	Auto mode
14	PSB_L2	Manual control mode

Figure 4.3: PS2 data and Arduino

#### 4.2 Wireless PS2 controller

The wireless PS2 controller is a remote control device for robots, smart vehicles, and various remotely controlled devices. It interfaces with peripheral microcontrollers and is easy to use.



• Figure 4.4: Wireless PS2 controller

- Receiver voltage: 3.3 VDC
- PS2 controller made of high-quality ABS material.
- 2.4 GHz wireless receiving technology, with range up to 10 m.
- Digital / analog operating mode, switchable at any time, with indicator light.
- 4-axis 12-button design to meet gaming button needs (digital mode: 2-axis 10 buttons; analog mode: 4-axis 12 buttons).
- Buttons support continuous trigger function for more convenient operation.
- Two high-precision analog joysticks for 360-degree control.
- The left and right sides each have an integrated vibration motor capable of fine vibration effects.
- Quick connection, fast response, no lag.
- Supports USB 1.1 / 2.0 (requires an additional USB-to-PS2 adapter)
- Supports WIN 98 / ME / 2000 / XP / Vista
- Power supply: 2 AAA batteries

- The product includes one wireless controller and one receiver unit.

#### 4.3 Developing the algorithm for the line-following robot

##### Algorithm

##### Algorithm idea

When the vehicle operates, the sensors collect information, which is converted through the ADC into electrical signals and sent to the microcontroller. The received value is compared with a preset value (threshold). If  $ADC[i] < \text{threshold value}$ , then the  $i$ -th receiving LED is over the floor area; otherwise, it is over the line area.

The sensor circuit uses 8 emitter-receiver LED pairs corresponding to 8 ADC values sent to the microcontroller. Based on the 8 received ADC values, the relative deviation between the robot trajectory and the desired trajectory is determined, and then this deviation is classified into levels.

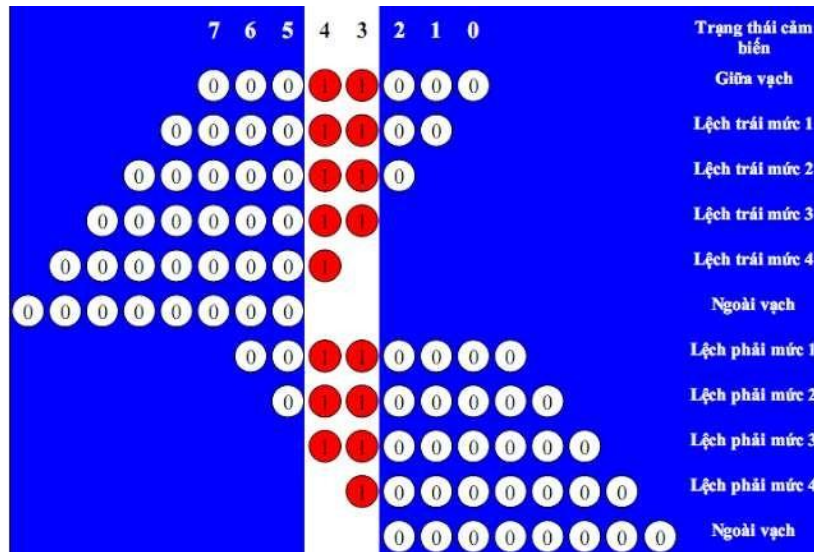


Figure 4.5: Levels of deviation from the robot trajectory according to the line-following sensors.

- Based on the deviation levels, adjust the speeds of the left and right wheels to bring the robot back to the trajectory.
- Specifically, to turn left, the speed of the right wheel is made greater than that of the left wheel by a value corresponding to the deviation levels (which must be determined experimentally), and vice versa. This method is simple, but the robot runs unstably, sometimes very fast and sometimes slow, and the stability depends greatly on the motor and the robot's mechanical structure.
- To overcome this drawback, a PID controller is applied to control the robot position.

#### PID control algorithm

PID is short for Proportional, Integral, and Derivative. Although it has existed for a long time, PID remains the most widely used control algorithm in automatic control applications. To better understand the nature of the PID algorithm, consider the example of position control of a car on a straight line. Suppose you have a vehicle (for example, a toy car) equipped with a DC motor. The motor generates a force to drive the vehicle forward or backward along a straight line as shown in Figure 12.

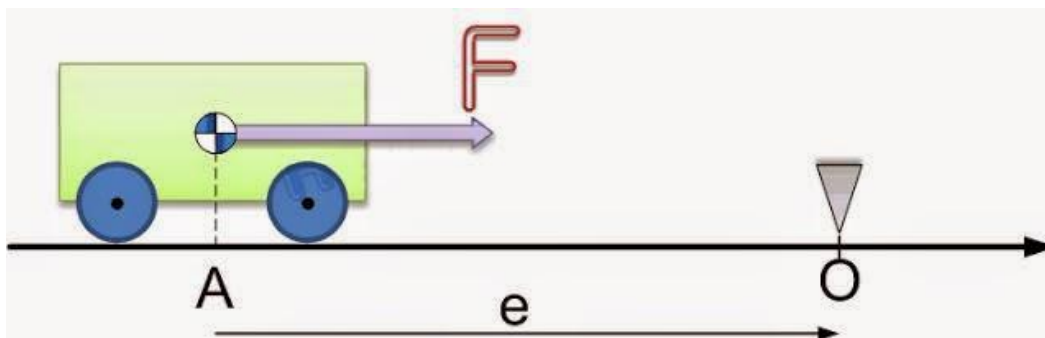


Figure 4.6: Example of vehicle position control on a straight line

- Let  $F$  be the force generated by the motor to control the vehicle. Initially, the vehicle is at position  $A$ . The task is to automatically control force  $F$  so that the vehicle reaches position  $O$  with the requirements of accuracy, fast response, and stability (small overshoot).
- Naturally, if the current position of the vehicle is far from the desired position (point  $O$ ), or in other words if the error is large, a large force  $F$  is needed to quickly

bring the vehicle to  $O$ . A simple way to formulate this idea is to use the linear relationship:

$$F = K_p * e \quad (1)$$

Here,  $K_p$  is a positive constant called the proportional gain, and  $e$  is the control error, that is, the distance from point  $O$  to the current vehicle position. The control goal is to drive  $e$  to 0 as quickly as possible. Clearly, if  $K_p$  is large, then  $F$  is

also large and the vehicle moves very quickly toward position O.

However, if force F is too large, it accelerates the vehicle very quickly (Newton's second law:  $F = ma$ ). When the vehicle reaches position O (that is,  $e = 0$ ), force F becomes 0, but due to inertia the vehicle continues moving to the right and overshoots point O. At this moment, the error e becomes nonzero again; this error is called overshoot. Then e is negative, and force F appears again in the opposite direction to pull the vehicle back toward O. But once again, because  $K_p$  is large, force F is also large and may pull the vehicle too far to the left of point O.

This process continues, and the vehicle keeps oscillating around point O. In some cases, the oscillation may move farther and farther away from O. The controller is then considered unstable. One proposal to reduce overshoot is to use a 'braking' component in the controller. Ideally, when the vehicle is far from O, the controller produces a large force F, but when the vehicle approaches O, the 'braking' component reduces the vehicle speed. We know that when an object oscillates around a point, its speed is greatest at the center of oscillation (point O). In other words, near point O, the error e changes most rapidly. The rate of change of e can be calculated by the derivative with respect to time. Thus, when the vehicle moves from A toward O, the derivative of the error increases in magnitude but acts opposite to force F because e is decreasing rapidly.

If the derivative is used as the 'braking' component, the vehicle overshoot can be reduced. This 'braking' component is exactly the D (Derivative) term in the PID controller under study. Adding this D term to the current P controller gives the following PD controller:

$$F = K_p * e + K_d * (de/dt)$$

Here,  $de/dt$  is the rate of change of the error e, and  $K_d$  is a non-negative constant called the derivative gain.

The presence of the D term reduces overshoot. When the vehicle approaches O, force F consists of two components:  $K_p * e \geq 0$  (P) and  $K_d * (de/dt) \leq 0$  (D). In some cases, the D component becomes greater than the P component, and force F reverses direction, effectively braking the vehicle so that its speed decreases sharply near

point O. A problem arises when the D component is too large compared with the P component, or when the P component itself is small. Then, as the vehicle nears O (but has not actually reached O), it may stop completely. At that moment, the D component becomes 0 because the error no longer changes, so  $F = K_p * e$ . Since both  $K_p$  and e are then small, F may be too small to overcome static friction. Imagine trying to push a truck weighing tens of tons: although a pushing force exists, the truck does not move. As a result, the vehicle would remain stationary even though the error e is still not zero. This situation is called steady-state error. To avoid steady-state error, an additional controller component is introduced to accumulate the error. When steady-state error occurs, the P and D components lose their effect, and the new control component accumulates the error over time, thereby increasing force F over time until it becomes large enough to overcome static friction and continue pushing the vehicle toward point O.

This 'accumulating' component is the I (Integral) term in the PID controller. As we know, the integral of a quantity over time is the accumulation of that quantity over time. The complete PID controller is therefore:

$$F = K_p * e + K_d * (de/dt) + K_i * \int e dt \quad (3)$$

(note:  $\int e dt$  is the integral of variable e with respect to t)

Thus, the function of each term in the PID controller is now clear. Depending on the purpose and the controlled object, the PID controller may be reduced to a P, PI, or PD controller. The main task of a PID controller designer is to choose  $K_p$ ,  $K_d$ , and  $K_i$  so that the controller performs well and stably (this process is called PID gain tuning). This is not an easy task because it depends on many factors.

- Practical experience in choosing  $K_p$ ,  $K_d$ , and  $K_i$ :
- Choose  $K_p$  first: test the P controller on the real object (or a simulation), and adjust  $K_p$  so that the response time is sufficiently fast while accepting a small overshoot.
- Add the D component to eliminate overshoot, increase  $K_d$  gradually, test, and choose an appropriate value. Steady-state error may appear.
- Add the I component to reduce steady-state error.  $K_i$  should be increased gradually from small to large in order to
- reduce steady-state error while preventing overshoot from reappearing. Algorithm flowchart:

4.4 Arduino programming: (Code appendix)

4.5 Some images of the completed robot

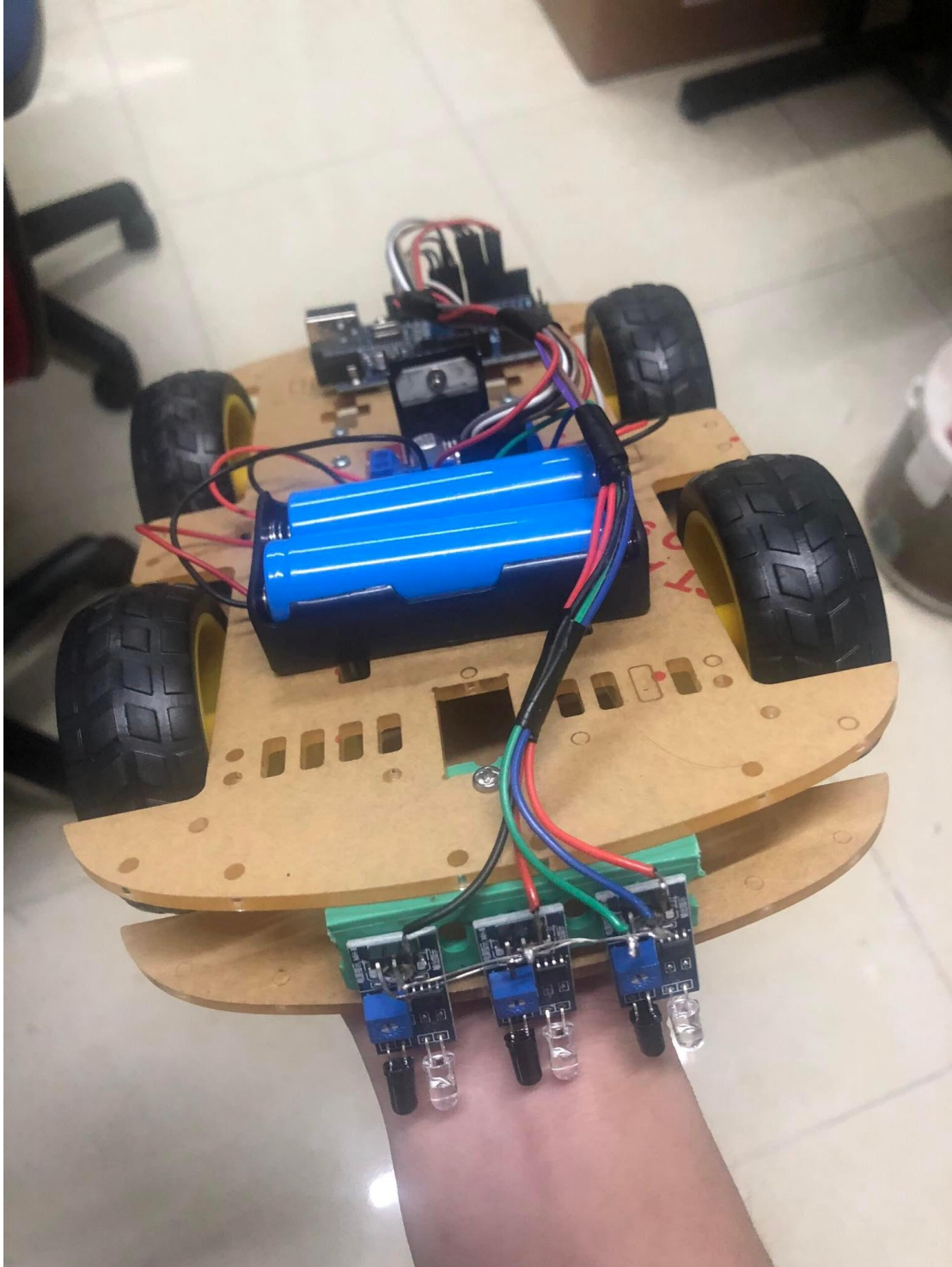


Figure 4.7: Completed robot image 1

Figure 4.8: Completed robot image 2

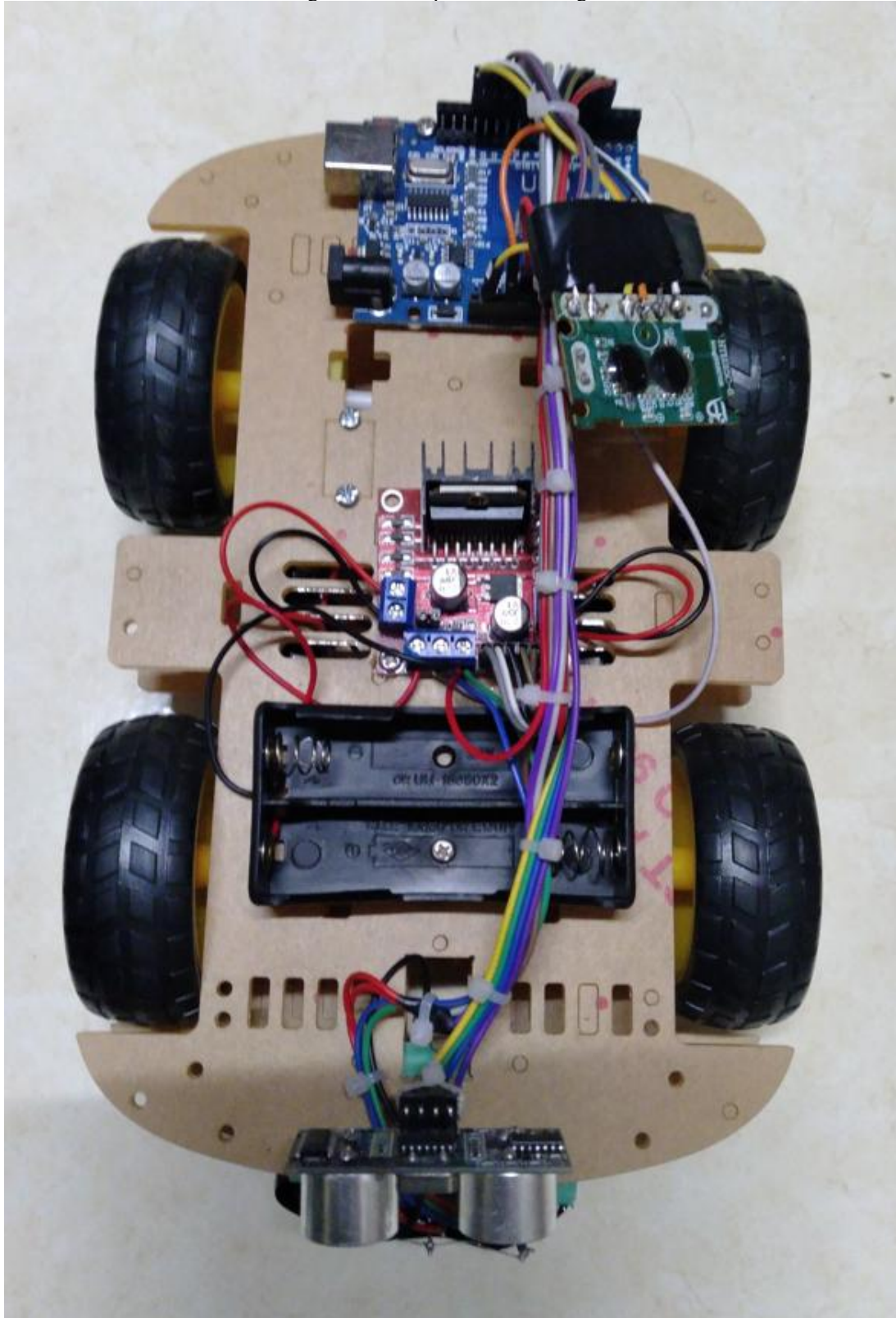


Figure 4.9: Completed robot image 3

The robot operates well, is stably controlled, and satisfies the initial requirements.

#### 4.6 Conclusion

This is only a small task in robot control programming, but through the process of designing and implementing the project, we have gained a great deal of valuable practical experience. The aim of the project is to create an initial platform for students to experiment with practical microcontroller applications, thereby enabling them to explore and develop many other applications needed in everyday life.

Through the project 'Design of a line-following control system for a robot', we found that this project is well suited to the knowledge we have learned in control engineering. Although we made our utmost effort to complete this course project using the knowledge we learned, together with reference books and other sources, shortcomings were unavoidable. Therefore, we sincerely hope to receive valuable comments from lecturers and fellow students so that the project can be improved to the highest level.

We hope for the lecturers' understanding and dedicated guidance so that we can continue to learn and approach new technologies and further improve this project.

We would like to take this opportunity to sincerely thank our two supervisors, M.Sc. Duong Quynh Nhat and M.Sc. Nguyen Trong Toan, who guided and helped us complete this project. We also thank the lecturers in the department for providing us with the knowledge necessary to complete this project, and for the valuable lessons and experiences they shared during teaching, which will certainly serve as a foundation for future success in our careers.

#### 4.7 Future development directions

Due to limited implementation time and limited knowledge, the project currently stops at controlling the robot via Bluetooth and the PS2 controller, and studying the PID-based line-following algorithm for the robot. Therefore, to improve the practical applicability of the project, we propose adding the following requirements:

Use more types of sensors, such as air concentration sensors, to measure air quality and then collect and transmit signals to a computer to detect highly polluted areas.

- Modify the mechanical structure so that the robot can move into complex terrains.
- The hardware of each module should be separated to facilitate modification.

- Develop a robot system for cargo transport and sorting at airports.

It is hoped that, with the above development directions together with further ideas and comments from lecturers, this project will continue to develop and become more complete and more practically applicable in real life.

*Sincerely thank you!*

#### CODE APPENDIX

```
#include <PS2X_lib.h> //for v1.6
#define ECHO 33
#define TRIG 31

#define hongngoai1 A2
#define hongngoai2 A1
#define hongngoai3 A0
#define ENA 2
#define inA1 3
#define inA2 4
#define inB1 5
#define inB2 6
#define ENB 7

PS2X ps2x;
int dem = 0;
int error = 0;
byte type = 0;
byte vibrate = 0;
int Sp = 200;
long duration;
float distance;

void lui() {
  analogWrite(ENA, Sp);
  analogWrite(ENB, Sp); //đi chậm cho khỏi trượt
  Line
  digitalWrite(inA1, HIGH);
  digitalWrite(inA2, LOW);
  digitalWrite(inB1, HIGH);
  digitalWrite(inB2, LOW);
}

void tien() {
  analogWrite(ENA, Sp);
  analogWrite(ENB, Sp); //đi chậm cho khỏi trượt
  Line
  digitalWrite(inA1, LOW);
  digitalWrite(inA2, HIGH);
  digitalWrite(inB1, LOW);
  digitalWrite(inB2, HIGH);
}

void trai() {
  digitalWrite(inA1, LOW);
```

```
digitalWrite(inA2, HIGH);
digitalWrite(inB1, LOW);
digitalWrite(inB2, LOW);
}
void phai() {
analogWrite(ENA, Sp);
analogWrite(ENB, Sp); //đi chậm cho khỏi trượt
Line
digitalWrite(inA1, LOW);
digitalWrite(inA2, LOW);
digitalWrite(inB1, LOW);
digitalWrite(inB2, HIGH);
}
void dung() {
analogWrite(ENA, Sp);
analogWrite(ENB, Sp); //đi chậm cho khỏi trượt
Line
digitalWrite(inA1, LOW);
digitalWrite(inA2, LOW);
digitalWrite(inB1, LOW);
digitalWrite(inB2, LOW);
}
void SR04() {
digitalWrite(TRIG, LOW);
delayMicroseconds(2);
digitalWrite(TRIG, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG, LOW);
duration = pulseIn(ECHO, HIGH); //Đọc xung
độ dài
distance = duration * 0.034 / 2; //cm // nhân giá
trị tốc độ truyền sóng 340m/s. chia 2: truyền vat
can và chuyển về.
Serial.println(distance);
}
void autoSetup() {
if (distance <= 8) {
traoi();
}
else {
tien();
}
}
void setup() {
Serial.begin(57600);
error = ps2x.config_gamepad(13, 11, 10, 12, true,
true); //setup pins and settings: GamePad(clock,
command, attention, data, Pressures?, Rumble?)
check for error
if (error == 0) {
Serial.println("Found Controller, configured
successful");
Serial.println("Try out all the buttons, X will
vibrate the controller, faster as you press harder;");
Serial.println("holding L1 or R1 will print out
```

```
the analog stick values.");
Serial.println("Go to www.billporter.info for
updates and to report bugs.");
}
else if (error == 1)
Serial.println("No controller found, check
wiring, see readme.txt to enable debug. visit
www.billporter.info for troubleshooting tips");
else if (error == 2)
Serial.println("Controller found but not
accepting commands. see readme.txt to enable
debug. Visit www.billporter.info for
troubleshooting tips");
else if (error == 3)
Serial.println("Controller refusing to enter
Pressures mode, may not support it. ");
//Serial.print(ps2x.Analog(1), HEX);
type = ps2x.readType();
switch (type) {
case 0:
Serial.println("Unknown Controller type");
break;
case 1:
Serial.println("DualShock Controller Found");
break;
case 2:
Serial.println("GuitarHero Controller Found");
break;
}
}
void loop() {
if ( dem == 1 ) {
SR04();
autoSetup();
}
if (error == 1) //skip loop if no controller found
return;
if (type == 2) { //Guitar Hero Controller
ps2x.read_gamepad(); //read controller
if (ps2x.Button(PSB_START)) //will
be TRUE as long as button is pressed
if (ps2x.Button(PSB_SELECT))
if (ps2x.Button(ORANGE_FRET)); // print
stick value IF TRUE
}
else { //DualShock Controller
ps2x.read_gamepad(false, vibrate); //read
controller and set large motor to spin at 'vibrate'
speed
if (ps2x.Button(PSB_START)) //will
be TRUE as long as button is pressed
if (ps2x.Button(PSB_SELECT))
if (ps2x.Button(PSB_RED)) {
if ( digitalRead(hongngoai1) == HIGH)
traoi();
else if ( digitalRead(hongngoai3) == HIGH)
```

```

tien();
    else if ( digitalRead(hongngoai2) == HIGH)
    {
        delay(50);}
    }
    phai();
    }
    if (ps2x.Button(PSB_PAD_UP)) { //will be
    TRUE as long as button is pressed
        Serial.print("Up held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_UP),
    DEC);
    tien();
    }
    if (ps2x.Button(PSB_PAD_RIGHT)) {
        Serial.print("Right held this hard: ");

    Serial.println(ps2x.Analog(PSAB_PAD_RIGHT),
    DEC);
    trai();
    }
    if (ps2x.Button(PSB_PAD_LEFT)) {
        Serial.print("LEFT held this hard: ");

    Serial.println(ps2x.Analog(PSAB_PAD_LEFT),
    DEC);
    phai();
    }
    if (ps2x.Button(PSB_PAD_DOWN)) {
        Serial.print("DOWN held this hard: ");

    Serial.println(ps2x.Analog(PSAB_PAD_DOWN),
    DEC);
    lui();
    }
    vibrate = ps2x.Analog(PSAB_BLUE); //this
    will set the large motor vibrate speed based on
    if (ps2x.NewButtonState()) //will be
    TRUE if any button changes state (on to off, or off
    to on)
    {
        if (ps2x.Button(PSB_L3))
            Serial.println("L3 pressed");
        if (ps2x.Button(PSB_R3))
            Serial.println("R3 pressed");
        if (ps2x.Button(PSB_L2))
            dem = 0;
        if (ps2x.Button(PSB_R2))
            dem = 1;
        if (ps2x.Button(PSB_GREEN));
    }
    if (ps2x.NewButtonState(PSB_BLUE))
    dung();
    if (ps2x.Button(PSB_L1)) {
        Sp--;
    }
    if (ps2x.Button(PSB_R1)) {
        Sp++;
    }
}

```

## REFERENCES

### ❖ Tài liệu tiếng việt

- [1] <https://xemtailieu.net/tai-lieu/arduino-robotics-489412.html>
- [2] <https://xemtailieu.net/tai-lieu/nghien-cuu-va-viet-ung-dung-ve-arduino-334155.html>
- [3] <https://www.arduino.cc/>
- [4] <https://xemtailieu.net/tai-lieu/tu-hoc-nhanh-arduino-cho-nguoi-moi-bat-dau-1572339.html>
- [5] Vi Điều Khiển Và Ứng Dụng Arduino Dành Cho Người Tự Học
- [6]. <https://htpro.vn/news/thu-vien-driver/huong-dan-su-dung-module-kq-130f-voi-arduino-12.html>
- [7] <https://hacktronics.co.in/ethernet-modules/kq-130f-power-line-carrier-data-transceiver-module>
- [8]. <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/>
- [9]. <https://randomnerdtutorials.com/nextion-display-with-arduino-getting-started/>
- [10]. <https://randomnerdtutorials.com/21-arduino-modules-you-can-buy-for-less-than-2/>
- [11]. <https://www.robotshop.com/community/robots/show/my-arduino-cmucam-indoor-rover>