THE UNIVERSITY OF DANANG
UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

CAPSTONE PROJECT

MAJOR: MECHATRONICS ENGINEERING

PROJECT TITLE:

DESIGN AND FABRICATION OF AGRICULTURAL ROBOTS

Supervisor : **Ph.D. THANH NGHI NGO**
Co-Supervisor : **Mr. BA KHANH TUAN VAN**
Approver : **Ph.D. PHUOC VINH DANG**
Students : **CHI THANH LE**
VAN NHUAN LE
Student ID : **101200288**
101200278
Class : **20CDT2**

Da Nang, 06/2025

SUMMARY

Topic: DESIGN AND FABRICATION OF AGRICULTURAL ROBOTS

Students: Chi Thanh Le Student ID: 101200288 Class: 20CDT2

Van Nhuan Le Student ID: 101200278 Class: 20CDT2

Supervisor: Dr.Thanh Nghi Ngo

Reviewer:

1. Actual demand:

Nowadays, weeds are a big challenge for many agricultural sectors in the world. The main reason is that weeds compete fiercely with crops or plants for space, sunlight and nutrients in the soil. In addition, weeds can also create conditions for insects to damage crops more easily. Therefore, weed control is always a top concern forfarmers. Currently, to eliminate stubborn weeds, farmers often use herbicides or manually pull them out. With the desire to apply scientific and technological advancements to agricultural production, enhance labor efficiency, and minimize negative impacts on the ecosystem, our team has decided to propose the idea of "**DESIGN AND FABRICATION OF AGRICULTURAL ROBOTS**." This innovation aims to replace traditional weed control methods, helping farmers eliminate weeds while ensuring environmental sustainability.

2. Scope of research topic:

- Research, design and manufacture of robots that can automatically kill weeds using Laser.
- Research and apply image processing technology integrated with AI to distinguish between crops and weeds.
- Research on calculating the structure to choose the right location
- Research and apply the use of clean energy such as solar panels as power sources for robots.
- Design and manufacture of motor control circuits and sensors for robots.
- Design user interfaces combined with predictions.

3. The content of the topic has been done:

Number of pages of description: Pages

Number of drawings: A0 drawings

Model: 1

4. Results:

- Research and apply Yolo v8 to image processing to distinguish plants.
- Successfully distinguish plants and weeds in the nursery.
- Transfer the actual coordinates of the plants to process so that the Robot can approach.
- Calculate and design the mechanism to find the location and drive the Robot.
- Design the robot using Solidwork software and draw the drawing using Autocad.
- Assemble the modules as well as draw the circuit principle on Autocad.
- Learn about solar energy sources and how to use them.
- Learn and use some types of sensors: Humidity, temperature.
- Research and apply IoT technology.

THE UNIVERSITY OF DANANG
University of Science and Technology
Faculty of Mechanical Engineering

SOCIALIST REPUBLIC OF VIETNAM
Independent – Freedom - Happiness

GRADUATION PROJECT TASK

No.	Student's name	Students ID	Class	Major
1	Chi Thanh Le	101200288	20CDT2	Mechanical Electronics
2	Van Nhuan Le	101200278	20CDT2	Mechanical Electronics

1. *Topic: DESIGN AND FABRICATION OF AGRICULTURAL ROBOTS*
2. *Topics covered by: ☐ Having signed an intellectual property agreement for the implementation results*
3. *Initial figures and data:*
4. *Contents of explanations and calculations:*

a. *General part:*

No.	Student's name	Contents
1	Chi Thanh Le	Research, study theory, study related documents on topics such as: parameter calculation, drive system, sensor, image processing theory.
2	Van Nhuan Le	

b. *Private part:*

No.	Student's name	Contents
1	Chi Thanh Le	+ Calculation, design and manufacture of robot mechanical structures. + Model assembly. + Design of robot control circuit system. + Building data transmission systems and web design. + Programming of kinematic control and sensors for Robots

2	Van Nhuan Le	+ Calculation, design and manufacture of robot mechanical structures. + Model assembly. + Build a robot control algorithm flowchart. + Image processing and actuator control. + Solar panel system installation.

5. *Drawings, graphs (specify types and sizes of drawings):*

a. *General part:*

No.	Student's name	Contents	
1	Chi Thanh Le	Robot general drawing	A0
2	Van Nhuan Le		

b. *Private part:*

No.	Student's name	Contents	
1	Chi Thanh Le	Control Circuit Block Diagram	A0
		Electrical Circuit Block Drawing	A0
2	Van Nhuan Le	Drawing of Dynamic Diagram	A0
		Detailed Drawing	A0

6. *Supervisor: Dr. Ngo Thanh Nghi*
7. *Project assignment date:*
8. *Project completion date:*

Da Nang, January.......,2025

Supervisor

Dr. Thanh Nghi Ngo

PREFACE

In the era of increasingly developing digital technology, robots have become an important component in many fields, from industrial manufacturers, healthcare, education to agriculture. With the ability to perform complex and dangerous tasks, tireless work, robots not only help improve work efficiency but also open up new potentials for the development of science and technology.

Realizing the importance and potential of robots, our group of students decided to delve deeper into the structure and operating principles of robots, and at the same time, we have chosen for ourselves the necessary skills to solve technical problems in practice. To enhance the applicability and expand the scope of robot operations, we have integrated image processing in agriculture to distinguish between crops and weeds. One of the most important formulas in agriculture today is weed control, environmental pollution, soil degradation and adverse impacts on human health.

Realizing this potential, our team has developed an AI-powered laser lawn robot that identifies and removes weeds with high precision and efficiency in specific crop stages. During the implementation of the project, the research team has consulted many documents, scientific studies as well as advanced systems in the world to build a solution suitable for practical conditions. The work of completing the graduation project encountered many difficulties and difficult problems. However, with the help and enthusiastic guidance of Mr. Ngo Thanh Nghi as well as other teachers, we were able to complete this project. However, with limited knowledge and no experience in calculation and practical construction, it is difficult to avoid mistakes. We respectfully hope to continue to receive guidance from teachers so that we can further improve our knowledge. Finally, I would like to sincerely thank the teachers in the Faculty of Mechanical Engineering - University of Science and Technology - University of Danang, especially Dr. Ngo Thanh Nghi for helping to complete this project.

Da Nang, 06/2025.

Students

Chi Thanh LE

Van Nhuan LE

ASSURE

Dear Faculty of Mechanical Engineering - University of Science and Technology - University of Danang.

We hereby guarantee that the project complies with the regulations on academic integrity.

Do not use documents and data without clear origins.

Do not create data during surveys, experiments, practices, and internships.

Do not use fraudulent forms in presenting or demonstrating academic activities or results from your academic process.

Do not use other people's words or expressions without permission, present, copy, translate paragraphs, or state other people's ideas without citation.

Our group understands that violating the principles not only affects the honesty of the project but also goes against ethics. Therefore, we pledge to take full responsibility.

Student implementation

Chi Thanh Le   Van Nhuan Le

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 TOPIC OVERVIEW

1.1 Current situation

Currently, weeds are a "nightmare" for the global agricultural industry, as they grow and spread rapidly, competing with crops for light, space, and nutrients. They also create favorable conditions for harmful insects to thrive, posing a serious threat to harvests. Therefore, weed control is always an urgent issue for farmers. To address this problem, the two most common methods used by farmers today are spraying herbicides and manual weeding. However, both methods have significant drawbacks: herbicides can pollute the soil and water, and negatively impact food safety—an issue that is increasingly emphasized in the selection of clean food. Meanwhile, manual weeding is too time-consuming and labor-intensive, especially in the context of an increasingly scarce agricultural workforce.

Figure 1–1 Growing vegetables and weeding [7]

With the desire to join the era of industrialization, modernization, application of technology and science in life and cultivation. The group came up with the idea of creating "***DESIGN AND FABRICATION OF AGRICULTURAL ROBOTS***" to improve work efficiency and quality for people, ensure food hygiene and safety on the table, and minimize the amount of chemicals left during the plant growth process.

1.2 Overview of the topic

1.2.1 Introduction to Automation in Agriculture

Automation in agriculture is the application of automation technology to agricultural activities to increase productivity and farming efficiency. Using automation equipment in agriculture helps save time, labor, increase production control and minimize errors, thereby improving product quality. Automation in agriculture can be applied in many activities such as planting, caring for crops, harvesting and transporting products.

The advantages of automation in agriculture include:

- Increased productivity: Automation systems in agriculture help increase productivity and production efficiency by reducing time and labor costs. At the same time, they increase accuracy and reduce dependence on human factors.
- Optimize resource use: Automation in agriculture can help optimize the use of resources such as water, fertilizers, pesticides, reduce waste and save costs.
- Ensure product quality: Automation equipment and systems help control product quality from the planting process to harvest. Thereby helping to ensure product quality meets standards and meets market requirements.
- Increased safety: Automation in agriculture reduces workers' exposure to hazardous chemicals and equipment. This reduces the risk of work-related accidents.
- Improved environment: Automation in agriculture can help reduce the amount of chemicals and fertilizers used. This reduces pollution and keeps the environment clean.
- Increased sustainability: Automation in agriculture helps increase the sustainability of the agricultural sector by minimizing negative environmental impacts, optimizing resource use, and increasing productivity.

APPLICATIONS OF AUTOMATION IN AGRICULTURE:

Integration of IoT sensors across agricultural operations. These smart devices and sensors are interconnected and remotely managed throughout the farming process, allowing environmental conditions to be monitored and greenhouse climates to be improved for optimal production. The use of LED technology in advanced farming systems to improve crop growth. This method is especially effective in areas with limited agricultural land or in urban areas, providing the right wavelengths of light to maximize crop growth.

Figure 1–2 Led technology [8]

Growing in greenhouses and net houses using hydroponics and aeroponics. These techniques isolate crops from external conditions, allowing for precision technology to be applied to achieve consistent and high-quality yields.

Figure 1–3 Greenhouse [8]

The use of photovoltaic cells (solar panels) to optimize land use and reduce energy costs. Most farm equipment and operations are solar powered, promoting sustainability.Moreover, the integration of solar panels into agricultural land can be thoughtfully designed to optimize space, for instance, by placing them over irrigation canals or underutilized areas, enabling the land to serve dual purposes energy production and crop cultivation. This forward-thinking approach contributes to environmental preservation by reducing greenhouse gas emissions and promotes the widespread adoption of renewable energy, aligning with global initiatives to address climate change.

Figure 1–4 Solar [7]

The use of robots is increasing to manage crops and livestock, especially in countries with aging populations or expanding agricultural operations. Robots streamline tasks such as planting, tending, and harvesting. The use of the internet, mobile devices, and cloud computing to improve farm management and financial services. By connecting all farm operations to external systems, these technologies provide efficient operational solutions. Furthermore, the ability to operate overnight in a variety of weather conditions improves operational efficiency, while integration with data analytics allows farmers to make informed decisions based on real-time insights. This technological shift also supports precision agriculture and is especially beneficial for fire farms to meet the growing global food demand.

Figure 1–5 Robots pick fruit in greenhouses [9]

The robot can see fruits and tell whether they are ripe or not, so it can determine which ones are ready to harvest, even in a messy and complex orchard. It can clamp down deep into vines to pick ripe fruit without disturbing the surrounding green fruit.

1.2.2 Introduction to the topic

Currently, weeds are the "nightmare" of the global agricultural industry, as they constantly compete for light, space and nutrients with crops, while creating conditions for harmful insects to thrive, seriously threatening crops. Therefore, weed eradication is always an urgent problem for farmers. To deal with this, the two most popular methods today are using chemical herbicides and manual weeding. However, both have significant disadvantages: herbicides can pollute soil and water and reduce plant diversity, while manual weeding is too hard, time-consuming and difficult to do in the context of increasingly scarce agricultural labor.

In order to exploit the power of science and technology to solve this problem, our group came up with a bold idea: "***DESIGN AND FABRICATION OF AGRICULTURAL ROBOTS*** ". This project not only aims to replace inefficient traditional methods, but also has the ambition to increase labor productivity, minimize environmental damage and shape a sustainable future for agriculture. By integrating advanced robotic technology with the ability to use precise laser beams, this initiative promises to bring a smart solution that helps farmers effectively remove weeds, protect the ecosystem and stay ahead of the current 4.0 technology trend.

1.2.3 Urgency of the topic

The urgency of the topic "Design and Fabrication of Agricultural Robots" is evident through the following aspects:

Growing Challenges Posed by Weeds: Weeds fiercely compete with crops for essential resources such as sunlight, water, and soil nutrients, leading to reduced agricultural yields and increased vulnerability to pests and diseases. Amid climate change and the rising prevalence of herbicide-resistant weeds, the need for an effective and sustainable weed control solution has become more pressing than ever.

Limitations of Traditional Methods: While chemical herbicides offer quick results, they come with significant drawbacks, including environmental pollution, soil degradation, and adverse effects on human health. On the other hand, manual weeding is labor-intensive, time-consuming, and increasingly impractical due to the widespread shortage of agricultural labor. This underscores the urgent need for a more advanced alternative.

Trend Toward Sustainable Agriculture: The global push for smart and eco-friendly agriculture aims to meet the rising food demands of an expanding population. Developing laser-based weed-control robots not only reduces reliance on harmful chemicals but also helps preserve ecosystems and enhances the quality of agricultural products, aligning with the stringent standards of modern markets.

Demand for High-Tech Integration: The advent of Industry 4.0 has paved the way for integrating robotics and artificial intelligence into agricultural production. The development of laser-equipped robots is a critical step toward optimizing labor efficiency and modernizing agriculture, thereby strengthening the competitive edge of nations in this sector.

1.2.4 Principle of operation

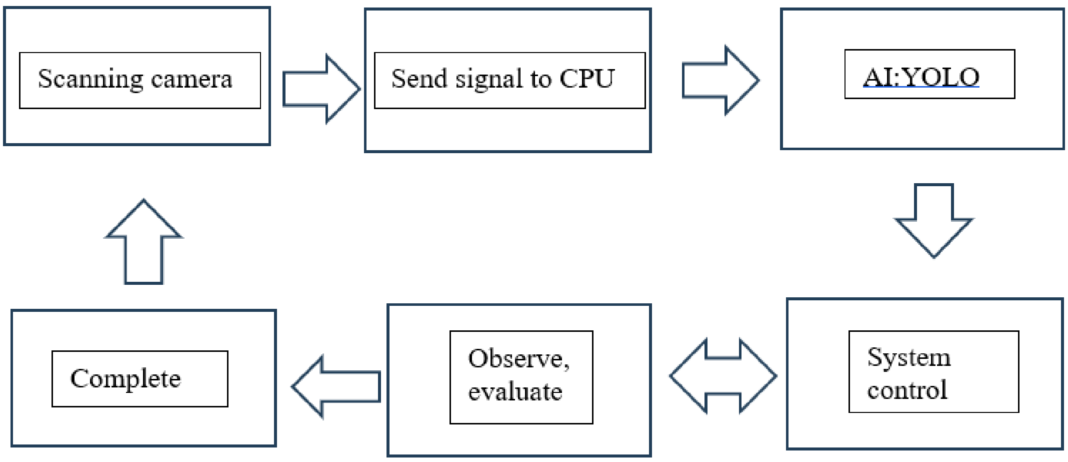

Figure 1–6 Principle

Scanning Camera: The process begins with a scanning camera, which is mounted on the robot. This camera continuously captures images of the field as the robot moves. Its primary role is to detect and identify weeds among the crops by scanning the surrounding environment in real-time.

Send Signal to CPU: Once the camera captures the images, the data (in the form of visual information) is transmitted to the Central Processing Unit (CPU). The CPU acts as the "brain" of the robot, processing the incoming data and preparing it for further analysis.

AI: YOLO (You Only Look Once) The processed data from the CPU is then fed into an AI system, specifically utilizing the YOLO algorithm. YOLO is a popular real-time object detection algorithm known for its speed and accuracy. In this context, YOLO analyzes the images to identify and classify objects, distinguishing weeds from crops. It uses pre-trained models to recognize patterns and features of weeds, ensuring precise targeting.

System Control: After the AI identifies the weeds, the system control unit receives the data and coordinates the robot's actions. This unit is responsible for directing the laser mechanism to target the identified weeds. It ensures that the laser is accurately aimed at the weeds while avoiding damage to the crops. The system control also manages the robot's movement and ensures efficient operation across the field.

Observe, Evaluation: Once the laser has been deployed to eliminate the weeds, the robot enters an observation and evaluation phase. The camera continues to monitor the area to assess the effectiveness of the weed removal. This step involves checking whether the targeted weeds have been successfully eradicated or if further action is needed. The evaluation data is sent back to the system control for adjustments, ensuring continuous improvement in performance.

Complete: If the evaluation confirms that the weeds have been successfully removed, the process is marked as complete for that specific area. The robot then moves on to the next section of the field, repeating the cycle until the entire area is weed-free.

Main controller

Central controller : ARDUINO MEGA 2560.

Image processing controller: COMPUTER LAPTOP.

Secondary control: ESP32

Communication between devices

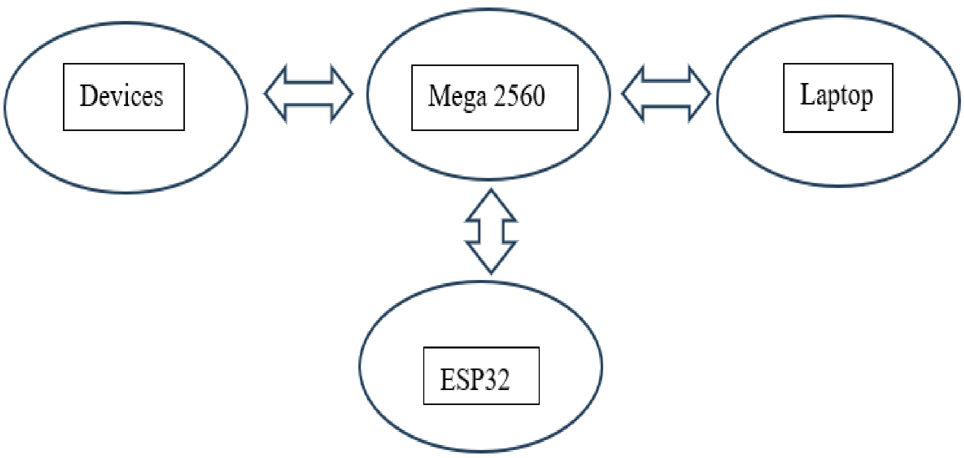

Figure 1–7 Communication

The system operates on a two-way communication model:

- Mega 2560 is the main control center, processing signals from devices and commanding them to operate.
- ESP32 acts as a wireless bridge, allowing communication between Mega 2560 and laptop.
- Laptop is the remotion control interface, image processing center.

1.3 Expected goals of the project

Table 1–1 Parameters for robot

Parameter	Describe
Performance	Stable operating system (90% - 95%)
Weeding Accuracy	Laser weed identification and removal with over 80% (Accuracy ±5%)
Herbicide Productivity	Kill 50 weeds in 1 minute with laser technology.
Environmental Monitoring	Provides temperature and humidity data on web interface.
Energy Sources and Storage	Continuous operation in sunlight

This is what the team plans to achieve. This is expected to be the first version of the team's development. There will be more advanced versions to come. During this phase, the team will prioritize completing key features, ensuring stability, ease of use, and future expansion. During the implementation, the team will also collect feedback, evaluate actual performance, and continuously improve towards more advanced versions.

The next versions will be developed with the goal of expanding functionality, improving performance, integrating more modern technologies, and providing better response. The team is committed to maintaining the spirit of continuous learning, innovation, and improvement throughout the product development process.

CHAPTER 2 ANALYSIS AND SELECTION OF DESIGN OPTIONS

2.1 Options for drive wheels

2.1.1 Option 1: Using stepper motor

A stepper motor is an electric motor that rotates in steps with a fixed angle, controlled by digital pulses. It is often used in applications that require precise position control(such as CNC machines, 3D printing).

Figure 2–1 Step Motor [10]

Advantages: Precise position control Stepper motors allow accurate control of wheel position and rotation angle without needing an encoder.

High holding torque: Provides significant torque when stationary, useful for preventing the robot from slipping on uneven agricultural soil.

No complex gearbox needed: Can achieve low speeds (75-100 RPM) through microstepping, reducing gearbox costs.

High durability: No brushes, resulting in less mechanical wear.

Disadvantages: Low efficiency: Consumes power continuously, even when stationary, reducing battery efficiency (~50-70%).

Torque drop at higher speeds: Torque decreases significantly above 100 RPM, limiting performance if higher speeds are needed later.

Noise and vibration: May produce noise or vibration at low speeds (75-100 RPM) without high-end microstepping.

Higher cost: Stepper motor + driver + optional gearbox is more expensive than equivalent DC motors.

2.1.2 Option 2: Using DC motor

DC motor (brushed or brushless) is a continuous rotation motor, widely used in mobile robots due to its low cost and ease of control.

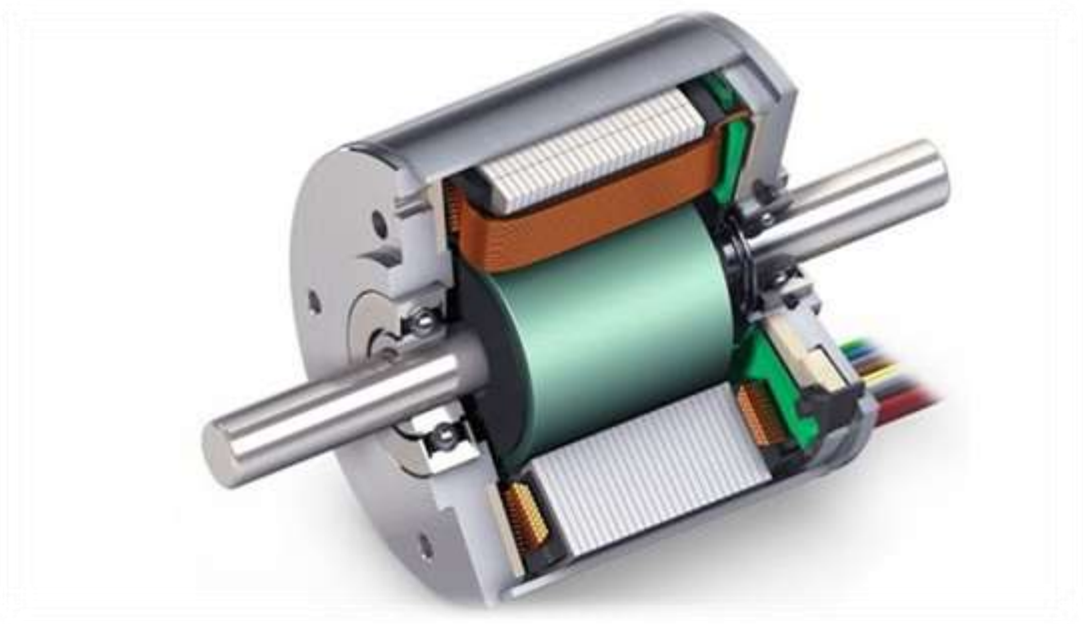

Figure 2–2 DC Motor [11]

Advantages:

High efficiency: Offers good efficiency (~80-90%), saving battery power, ideal for mobile robots.

Easy control: Requires only a PWM circuit (e.g., L298N, BTS7960) to adjust speed and direction, simpler than stepper motors.

Lower cost: DC motors are generally cheaper than stepper motor setups.

Stable torque: Provides consistent torque at low speeds (75-100 RPM), suitable for agricultural soil.

Terrain adaptability: Easily handles varying loads (e.g., soft or uneven soil), especially with a durable metal gearbox.

Disadvantages: Poor position control less precise than stepper motors, requiring an encoder for accurate positioning.

Maintenance: Brushed DC motors require periodic brush replacement, though brushless types are more expensive and complex.

Gearbox dependency: Requires a high-quality gearbox, which may increase costs.

Conclusion: DC motor options

The DC motor is the optimal choice, as you selected, due to its high efficiency, low cost, ease of control, and suitability for agricultural terrain. If precise positioning is needed later, an encoder can be added to the DC motor instead of switching to a stepper motor.

2.2 Options for select tranmission

2.2.1 Option 1: Belt transmission

Belt transmission a type of friction drive that uses pulleys and elastic belts to transmit power between two or more shafts. The main advantage of this system is its ability to operate efficiently under a wide range of speed and power requirements. It has higher tensile strength and can withstand sudden changes in load and reduces vibration. It operates smoothly and quietly.

Figure 2–3 Belt transmission [12]

Main components of a belt drive system:

A belt drive system typically consists of the following key components:

Pulley: The primary component of the system, pulleys can be made from various materials such as steel, aluminum, or plastic, depending on the application.

Belt: This serves as the connecting link between the pulleys. Belts can be crafted from materials like rubber, polyurethane, or leather, chosen based on the specific use case. The material of both the belt and pulley significantly impacts the system's performance. Rubber is commonly used due to its elasticity and durability, though specialized applications may call for materials like polyurethane or nylon to enhance strength and load capacity.

Frame: This structure supports and stabilizes the pulleys, ensuring consistent operation.

Shaft: The shaft supports the pulleys and facilitates the transfer of motion from the driving shaft to the driven shaft. It is typically made of metal or, in some cases, plastic.

Each component in a belt drive system plays a critical role in ensuring optimal performance.

Operating principle of a belt drive:

When one pulley rotates, it pulls the belt along, transferring motion and power to the other pulley. This movement depends on the rotational speed and torque of the driving pulley. A faster-spinning pulley increases the transmitted force, while a slower speed reduces it.

The efficiency of a belt drive system hinges on several factors, including the friction between the belt and pulley, the belt's tension, and the contact angle between the belt and pulley. All these elements must be carefully optimized to ensure the system operates effectively.

Advantages of belt drive**:**

One of the standout advantages of a belt drive is its ability to transmit power without requiring complex adjustments. This system operates reliably over extended periods with minimal maintenance. Moreover, it offers impressive flexibility in adjusting speed and torque, allowing users to easily tailor it to their specific needs.

Additionally, belt drives excel at reducing noise. Compared to other transmission systems like gears, they produce less vibration and sound, creating a quieter and more pleasant working environment.

Disadvantages of belt drive:

Despite its many benefits, belt drives come with certain limitations. Notably, their load-carrying capacity is relatively low compared to other transmission systems. For applications involving heavy loads, alternatives such as gear drives are often a better choice.

Furthermore, belts tend to wear out over time, necessitating regular inspections and replacements to maintain performance. Neglecting this can lead to a significant drop in transmission efficiency.

2.2.2 Option 2: Chain drive

Choose the chain drive because it has the following characteristics suitable for robots operating in agricultural environments:

High durability: Metal chains can withstand heavy loads and work well for a long time.

No Slip: The chain connects firmly to the sprocket, ensuring precise transmission, especially useful on farmland where stable traction is required.

High efficiency: Transmission efficiency is close to 98-99%, less energy loss than belt.

Good load capacity: Suitable for 50 kg load and soft, rough ground conditions ($\mu = 0.3$).

So, we see that the robot has slow speed, moves on dirt roads so it needs a large torque. The robot moves flexibly so it needs a compact, simple transmission that is easy to assemble and edit, so we choose a chain drive as appropriate.

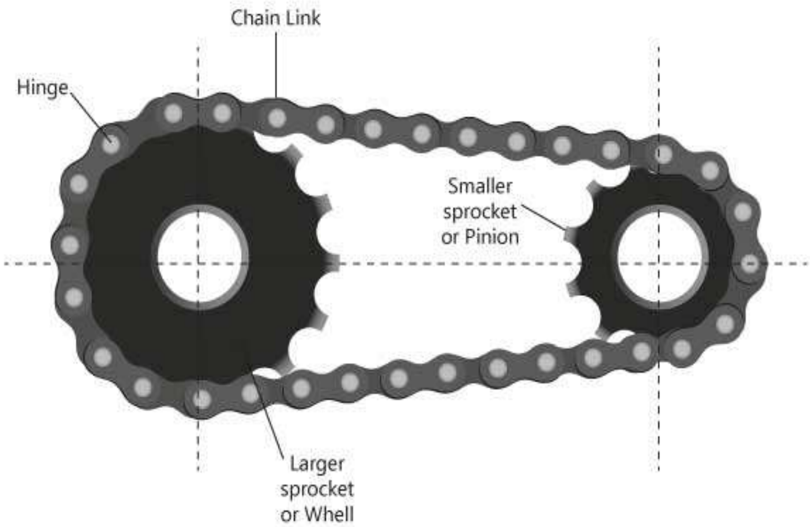

Figure 2–4 Construction of a chain drive [13]

Select chain type:

There are 3 types of chains: Bush chains, roller chains and toothed chains. We choose roller chain because the design has rollers so that the sliding friction between the Bush and the dental discs can be replaced by rolling friction. As a result, the durability of roller chain is higher than that of bush chain, the manufacturing is also not complicated and is widely used.

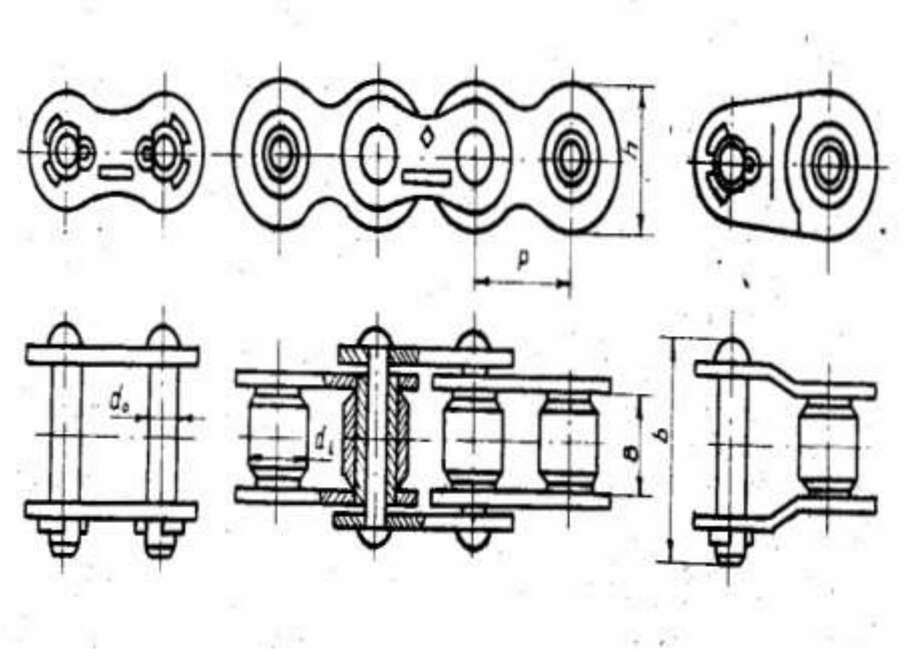

Figure 2–5 Roller chain structure [1]

2.3 Options for making actuators for robots.

2.3.1 Option 1: Robotic arm

Figure 2–6 Robotic arm [9]

Agricultural robots are defined as automated or semi-automated devices used to perform tasks in agriculture. There are many types of robots used in agriculture, including harvesting, sowing, irrigation, and soil analysis robots. Each type of robot has its own functions and advantages, contributing to the optimization of agricultural production processes.

The benefits of using robots in agriculture are clear. First, robots help reduce labor costs, an important factor in the context of agricultural labor shortages. Second, robots help increase productivity by working more continuously and accurately than humans. Tomato harvesting robots can harvest fruits with a high degree of uniformity in ripeness and size and with an accuracy of up to 86%. Finally, the use of robots also helps minimize negative impacts on the environment by using resources more efficiently and reducing the amount of pesticides and fertilizers needed. However, besides the benefits, the use of robots in agriculture still faces many challenges.This environmentally friendly approach contributes to soil health, biodiversity conservation and greenhouse gas emission reduction, in line with sustainable farming practices.

Table 2–1 Advantages and disadvantages of robot arms

Advantages	Disadvantages
Highly flexible, can reach any angle. Work in complex envriroments .without being affected, precise control.	Very high price on the market. Complicated, reversibledynamic equations. Requires many devices for precise control. Complex mechanical structure, difficult to manufacture and maintain.

2.3.2 Option 2: CNC mechanism.

Table 2–2 Advantages and disadvantages of CNC structure

Advantages	Disadvantages
Simple structure, easy to use, easy to install, easy to control. Simple calculation equation. High accuracy. Low energy consumption.	Limited working area compared to the arm.

Analysis and selection of mechanical transmission solutions. The transmission system controls the operation of the agricultural robot. The robot will perform translational movements including: Translational movement of the moving mechanism along the planting row, lifting and lowering movement of the agricultural tool storage structure, pushing and retracting movements of the harvesting or sowing arm.

Due to the characteristics of the agricultural environment, the transmission systems must ensure durability, the ability to operate in muddy, dusty environments and changing weather conditions.Specifically, the robot in question is designed to perform a number of translational movements necessary for its agricultural functions. These movements include the translational movement of the actuator along the crop rows, allowing the robot to efficiently navigate through the field while maintaining alignment with the crop rows. Additionally, the system facilitates the raising and lowering of the agricultural tool storage structure, which is important for adjusting the height of tools such as plows, seeders, or harvesters to suit different stages of operation or crop types. Furthermore, the drive system controls the movement and retraction of the strategic harvester or sowing arm, allowing the robot to precisely raise and retract its arm to sow, fertilize, or harvest crops without damaging the crop.

=> The group decided to choose the CNC actuator: Cost-effective, precise control, easy to calculate dynamics. Suitable for the job of bringing the Laser to the weed position on the plant bed.

2.3.3 Choosing mechanical solutions

For the X and Y axes, the team chose a toothed belt drive combined with a slider.

Advantages:

- Simple, compact structure, easy to install and maintain.
- High load capacity, ensuring reliability during operation.
- Large transmission ratio, helps to create large thrust force while requiring small impact force, suitable for agricultural environment.
- Good self-braking ability, minimizing the risk of slipping or moving incorrectly when operating in shaking conditions.

Disadvantages:

- Transmission efficiency is lower than that of toothed belt or rack transmission.
- Threads can wear quickly if operating at high speed for a long time.
- Regular maintenance is required to avoid dust and sand affecting operating efficiency.

2.3.4 Analysis and selection of electric drive solutions.

The agricultural robot system requires controlling the position of the robot arm when emitting a laser beam. In addition, the motor needs to have appropriate torque to be able to withstand the load when operating in outdoor conditions. The requirement is to choose a type of drive motor that can accurately control the position and maintain the torque when lifting and lowering the load.

The group uses stepper motors because:Stepper motors are an ideal choice for projects that offer outstanding advantages at low cost, suitable for student project budgets, position and speed control, simplifying system design. Stepper motors maintain good torque without the use of brakes or speed converters, ensuring stable performance. In addition, do not spend more money, the base motor has low maintenance costs, increased durability and reduced maintenance.

CHAPTER 3 COMPONENTS OF THE SYSTEM

3.1 Arduino Mega 2560

3.1.1 Introduction

The Arduino Mega 2560 is a powerful microcontroller designed for projects that are complex and require multiple input/output (I/O). Developed based on Microchip's ATmega2560 microcontroller, Arduino Mega 2560 provides powerful processing capabilities and high flexibility, suitable for many applications in fields such as automation, robotics, and remote control. far and more.

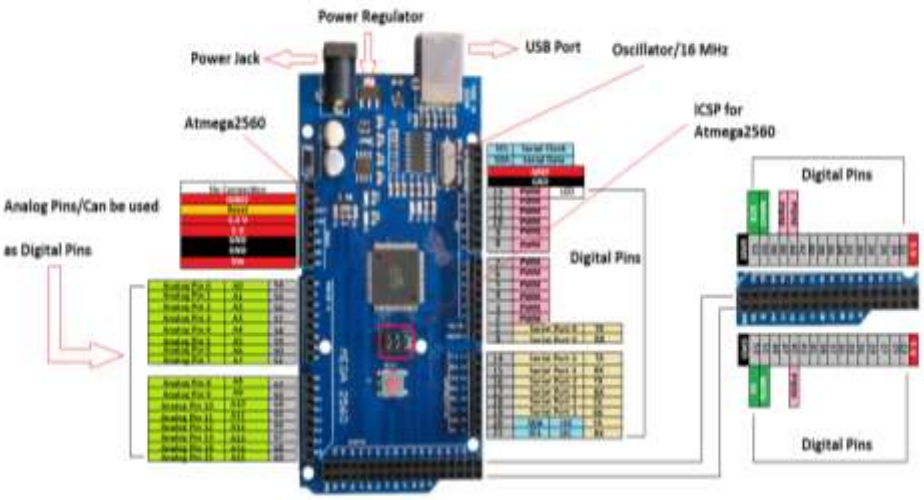

Figure 3–1 Arduino Mega 2560 [14]

3.1.2 Specification

- Microcontroller: ATmega2560
- Operating voltage: 5V
- Number of Digital I/O pins: 54
- Number of Analog Input pins: 16
- Current per I/O pin: 20 mA
- Current for 3.3V pin: 50 mA
- Flash memory: 256 KB (of which 8 KB is used for bootloader)
- SRAM: 8 KB
- EEPROM: 4 KB

ATmega2560 microcontroller, with 5V operating voltage, along with 54 digital I/O pins, 16 analog input pins, 20 mA current capacity for each I/O pin and 50 mA for 3.3V pin, and powerful memory configuration with 256 KB flash.

3.1.3 Outstanding features

Large number of I/O pins: With 54 Digital I/O pins and 16 Analog Input pins, the Arduino Mega 2560 is well suited for projects that require multiple connections to sensors, actuators, and external devices.

Powerful processing capabilities: ATmega2560 microcontroller with 16 MHz clock speed provides powerful enough performance to handle complex tasks and real-time requirements.

Easy programming: Arduino Mega 2560 is powered by the Arduino IDE programming environment, an easy-to-use and powerful tool that makes programming and uploading programs to the board simple.

High expandability: Arduino Mega 2560 can be easily combined with many different types of shields such as Ethernet shield, Motor shield, and many other shields, helping to expand the features and capabilities of the project.

Large Support Community: Arduino Mega 2560 is supported by a large and enthusiastic user community. This provides a rich source of resources from tutorials, source code libraries, to sample projects, making learning and implementing projects easier.

3.2 Ramsp 1.4

3.2.1 Introduction

Shield RAMPS 1.4 (Rep Rap Arduino Mega Pololu Shield) is an expansion board for Arduino Mega 2560, specifically designed for in-house applications and other stepper control projects. This shield is widely used in the RepRap community and DIY machine projects in 3D due to its performance, expandability and ease of use, applicable to a wide range of models. It provides a powerful platform for integrating multiple stepper motor controls, limited to the popular A4988, allowing precise control of up to five stepper motors, ideal for managing the X, Y, Z axes and additional in-house machines.

Figure 3–2 Ramsp1.4 [14]

3.2.2 Specification

Compatible with: Arduino Mega 2560

Control support: 5 stepper motors (X, Y, Z, E0, E1)

Stepper motor driver: Supports drivers such as A4988, DRV8825, and similar driver modules

Expandability: Supports connection to LCD monitor, SD card reader, and other modules via expansion pins

Power supply: Supports 12V-35V DC power for motors and peripheral devices

3.2.3 Outstanding features

Multiple stepper motor support: Shield RAMPS 1.4 is capable of controlling up to 5 stepper motors.

Integrated protection circuit: Shield is equipped with overcurrent and short circuit protection circuits, ensuring the safety of the system and components.

High expandability: With expansion pins, the RAMPS 1.4 shield can be easily connected to additional modules such as LCD screens, SD card readers, and temperature sensors, helping to expand functionality and 3D printer control capabilities.

Easy driver replacement: Stepper motor driver modules such as A4988 or DRV8825 can be easily replaced when needed, making system maintenance and upgrades simpler.

3.2.4 Connection

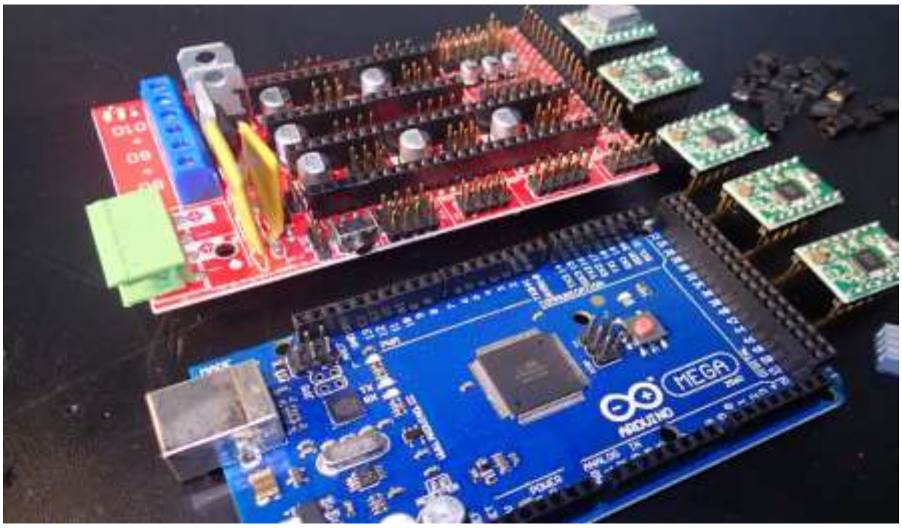

Figure 3–3 Attach RAMPS 1.4 to Arduino Mega 2560 [15]

Connecting the components in your setup, including the Arduino Mega 2560 and the RAMPS 1.4 shield along with the stepper motor drivers, is an important step for projects like 3D printers or agricultural robots. Start by aligning and securely installing the RAMPS 1.4 shield onto the Arduino Mega 2560, making sure all pins are connected properly. Next, install the stepper driver modules into the designated X, Y, Z, E0, and E1 slots on the shield, aligning them properly with the potentiometers facing outwards, and optionally adding a heatsink to prevent overheating.

Connecting the stepper motor: Connect the stepper motors to the X, Y, Z, and E (extruder) connectors on RAMPS 1.4. Make sure the motor windings are connected properly (usually pairs A and B).

Install Stepper Drivers: Attach stepper drivers to RAMPS 1.4. Make sure the jumpers under the drivers are properly installed to control microstepping as desired.

Connect Endstops: Connect the endstops to the X, Y, and Z min/max ports on RAMPS 1.4. Endstops are typically connected to the signal (S), ground (G), and vcc (V) pins.

Power connection: Connect the power source to RAMPS 1.4. Make sure that the power supply provides enough power to the motors and other components. Usually a 12V or 24V power source will be needed.

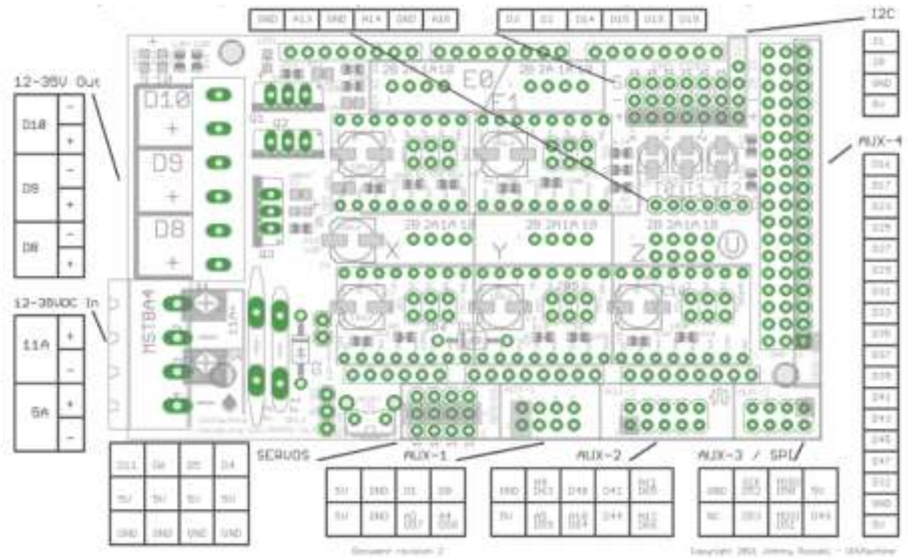

Figure 3–4 RAMPS 1.4 [15]

Specific description: D8, D9, D10 pins (12-35V Out): These are controllable outputs, used to power devices such as fans, heaters or hotbeds on 3D printers. These pins can be powered from a 12-35V power source.

D11, D6, D5 (SERVO) pins: These pins can be used to control servo motors. These outputs provide 5V power, suitable for servos with corresponding voltage levels.

AUX-1, AUX-2, AUX-3, AUX-4 connections:

These AUX pins are often used to connect expansion modules, such as sensors or control screens. These pins are designed to be used for different customization purposes depending on the system.

I2C pins: This is the I2C interface, a popular communication standard used to connect sensors, memories or other peripherals.

Motor Drivers: The areas with pins like 2B, 2A, 1A, 1B (pairs like E0, E1, Z, X, Y) are usually where stepper drivers are mounted. These drivers control the stepper motors for the system axes like X, Y, Z and the Extruder axis on a 3D printer or CNC machine.

12-35VDC In (11A, 5A): This is the power input to the board. The input voltage ranges from 12V to 35V, with two different current levels: 11A and 5A, usually powering different components like motors and heating devices.

Fuses: The board is protected by a fuse to protect the devices in case of voltage or current problems.

Motor Control Signal Pins (X, Y, Z, E0, E1): These are the pins that connect to the motor control drivers for the X, Y, Z, and E (Extruder) axes. These ports are responsible for controlling the movement of the axes in the system.

This board is commonly used in DIY projects for 3D printers, CNCs, and other multi-axis control systems, with corresponding outputs and inputs for devices such as stepper motors, sensors, and power supplies.

3.3 Esp32

The SP32-WROOM-32 is a powerful, intelligent, multi-purpose Wi-Fi + Bluetooth + Bluetooth LE MCU module that guides a wide range of applications. The module delivers superior power, supports frequencies up to 240 MHz, enables smooth multitasking, and meets complex computing requirements. The ESP32-WROOM-32 integrates up to 16 MB of flash memory (depending on configuration) and 520 KB of SRAM, along with advanced wireless communication protocols such as Wi-Fi and Bluetooth, making it an ideal solution component for IoT applications ranging from smart device control to real-time agricultural monitoring. In addition, the module still supports a variety of interfaces such as I2C, SPI, UART, I2S, and ADC, allowing easy integration with sensors, displays, and other peripherals. With low power consumption in deep sleep (deep sleep) mode and power saving mode, ESP32-WROOM-32 is suitable for projects that require long-term battery operation, such as wearables or remote monitoring systems. This flexibility, combined with a large development community and rich supporting documentation, makes the ESP32-WROOM-32 a comprehensive solution for developers and engineers in a wide range of fields, from robotics, home automation to complex industrial systems.

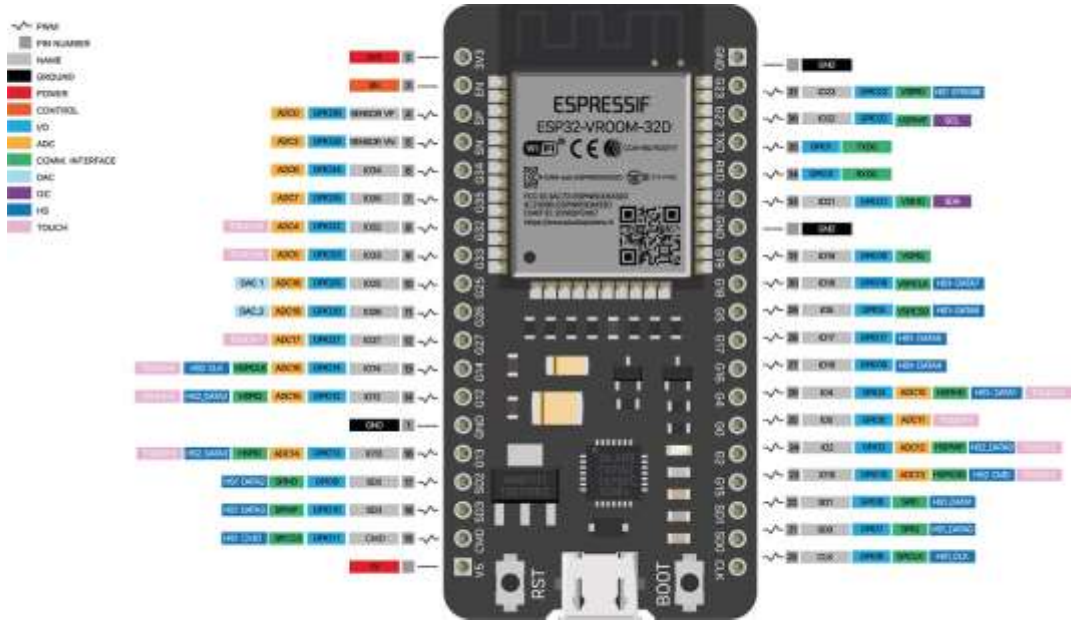

Figure 3–5 Esp32 [16]

- Wi-Fi module operates in the range of 2.4 GHz - 2.5 GHz.
- Three operating modes: 1. Access point. 2. Client. 3. Access point + station.
- 32-bit dual-core processor.
- Operating voltage is 3.3 V.
- Clock frequency from 80 MHz and up to 240 MHz.
- SRAM memory is 512 KB.
- ROM memory is 448 KB.
- External flash memory is supported and up to 32 Mb i.e. 4MB.
- Maximum current in each pin is 12 mA but 6 mA is recommended.
- It has 36 general purpose input/output pins.
- General purpose input/output pins come with PWM/I2C and SPI functionality..
- Operating voltage 2 to 3.6 V.

ESP32 is equipped with about 34 multi-function I/O (Input/Output) pins, allowing flexible programming to communicate with a variety of peripheral devices. These pins can be configured as inputs or outputs, and support many protocols such as digital, analog (ADC/DAC), PWM, SPI, I2C, UART, ... to control sensors, motors, LEDs, or communicate with other microcontrollers. Some pins have special functions such as touch sensors, interrupt functions, or support high-speed data transmission functions.

Name	No.	Type	Function
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS

Figure 3–6 IO expansion pins [16]

3.4 Camera

Resolution: 2K (2560x1440 pixels), for sharp images, suitable for video chatting or high-quality recording.

Microphone: Built-in mic, supports clear voice recording during video calls.

Compatibility: For PC and laptop, can be connected via USB port.

Application: Suitable for video chatting, online meetings, or livestreaming with good image and sound quality.

Figure 3–7 Camera [17]

3.5 Step 42

Stepper motor 42, also known as NEMA 17 stepper motor, is one of the popular types of stepper motors and is widely used in industrial and DIY applications. Known for their high accuracy and fine position control, the 42 stepper motor is the ideal choice for applications requiring precise control of position and speed. The NEMA 17 stepper motor comes in a variety of configurations, with holding torques ranging from 0.3 Nm to 0.5 Nm, depending on the model and current rating, making it suitable for tasks that require moderate torque without sacrificing accuracy. It works well with driver boards such as the A4988 or DRV8825, often used in conjunction with microcontrollers such as the Arduino or ESP32, allowing for seamless integration into custom projects.

Figure 3–8 Step42 [10]

- Motor type: Bipolar or unipolar stepper motor.
- Frame size: NEMA 17 (42mm x 42mm).
- Step angle: 1.8° per step (equivalent to 200 steps per revolution).
- Voltage: Normally from 12V to 24V (depending on engine type).
- Current: About 1A to 2A per coil.
- Torque: Usually ranges from 30Ncm to 60Ncm (equivalent to 300gf.cm to 600gf.cm).
- Number of wires: 4 wires (bipolar) or 6-8 wires (unipolar).

3.6 Driver A4988

The A4988 is a powerful and versatile bipolar stepper motor driver, suitable for a wide range of stepper motor control applications. With current regulation capabilities, support for multiple microstepping modes, and built-in protection functions, the A4988 brings convenience and safety to engineering and DIY projects. Widely used in 3D printers, CNC machines, and automation systems.

Figure 3–9 A4988 [19]

Modes.

Table 3–1 Step modes

MS1	MS2	MS3	Microstep
LOW	LOW	LOW	1:1
HIGH	LOW	LOW	1:2
LOW	HIGH	LOW	1:4
HIGH	HIGH	LOW	1:8
HIGH	HIGH	HIGH	1:16

Adjust A4988 Driver:

The maximum current of stepper motor :

Protect motors and drivers from overload: Excessive current can cause overload and heat up the motor or driver, leading to damage or reduced device life.

Temperature control: Excessive current increases the temperature of both the motor and driver.

3.7 BTS7960

The BTS7960 43A motor driver is a high-power DC motor driver module, using the BTS7960 integrated circuit. This is a powerful dual H-bridge driver, capable of driving DC motors with a maximum current of up to 43A and an operating voltage between 5.5V and 27V. Thanks to the design using high-performance MOSFETs built into the BTS7960 IC, this module allows precise control of the motor speed and rotation direction via PWM pulses, while protecting the circuit against problems such as overheating, overcurrent, and short circuit. The BTS7960 is well suited for applications that require heavy-duty or high-current motor control, such as industrial robots, homemade electric vehicles, mini cranes, automatic conveyors, or mechatronic systems in production lines. The module has a built-in aluminum heatsink for efficient heat dissipation during continuous operation. In addition, BTS7960 also supports easy control from popular microcontrollers such as Arduino, STM32, ESP32 through IN/EN control pins.

Figure 3–10 BTS7960 [20]

Main specifications:

- Power: 6 ~ 27V.
- Load current: 43A (Resistive load) or 15A (Inductive load).
- Control logic signal: 3.3 ~ 5V.
- Maximum control frequency: 25KHz.
- Automatic shutdown when low voltage: to avoid motor control at low voltage, the device will automatically shutdown. If the voltage < 5.5V, the driver will automatically disconnect the power and will reopen after the voltage > 5.5V.
- Dimensions: 40 x 50 x12mm.
- VCC : Logic level control source (5V - 3V3)
- GND : Ground pin.
- R_EN = 0 Disable right half H-bridge. R_EN = 1 : Enable right half H-bridge.
- L_EN = 0 Disable left half H-bridge. L_EN = 1 : Enable left half H-bridge.
- RPWM and LPWM : motor speed and direction control pins.
- RPWM = 1 and LPWM = 0 : Forward rotation motor.
- RPWM = 0 and LPWM = 1 : Reverse rotation motor.
- RPWM = 1 and LPWM = 1 or RPWM = 0 and LPWM = 0 : Stop.
- R_IS and L_IS : combined with resistors to limit the current through the H-bridge

3.8 Module relay.

Specifications:

Can select high or low trigger level.

Maximum load: AC 250V / 10A, DC 30V / 10A.

Using Opto isolation, for stable performance, limiting interference to MCU microcontroller blocks due to magnetic flux when closing and opening the relay, low trigger current: 5mA. Has High/Low trigger jumper. Power indicator LED (green), output indicator LED (red). 4 fixed bolt holes, 3.1 mm hole.

Figure 3–11 Relay [21]

Connection terminal:

- DC+: Connects the positive pole
- DC -: Connected to the negative pole of the power source.
- IN1: Is the relay 1 control pin, depending on the jumper selecting High/Low,
- S1: Is the pin to select the High or Low logic level for relay 1 - 8.
- NO: Normally open contact of the relay.
- COM: COM contact
- NC: Normally closed contact of the relay.

3.9 Gear motor

The motor uses an operating voltage between 12V and 24V DC, with a rated power of 250W, suitable for applications requiring high torque and stable operation for a long time. The motor's main shaft rotates at a speed of about 3300 rpm, but after being transmitted through an integrated gearbox with a transmission ratio of 10:1, the output speed is reduced to about 330 rpm. The use of a gearbox not only reduces the rotation speed but also increases the output torque, making it easier for the motor to pull heavier loads, especially useful in systems such as conveyors and other automated mechanical devices. Thanks to the combination of moderate power, stable rotation speed and efficient transmission, this type of motor becomes the ideal choice for systems requiring high precision and traction. In addition, the motor is designed with a sturdy metal shell, good heat dissipation, helping to maintain stable performance during continuous operation. At that time, the top of the head is often flat or round, easily connecting to transmission mechanisms such as gears, couplings or belts. With flexible operating applications, the motor can be easily integrated into a control system using 12V or 24V power.

Figure 3–12 DC 12V [22]

Power: The motor power must be greater than or equal to the power required to lift/pull the load.

Torque: The greater the torque, the more powerful the motor, the better its ability to overcome heavy loads.

Speed: The motor's rotation speed will determine the lifting/pulling speed of the winch.

Efficiency: High efficiency helps reduce power consumption and increase motor life.

Gear ratio: The gear ratio of the reducer will affect the speed and output torque of the motor. Speed control method: PWM, voltage change,...

3.10 Limit switch

Introduction to limit switches: A limit switch is an electromechanical device used to detect the presence, position, or movement limit of an object in automated systems such as CNC machines, robots, or production lines.When a machine part touches the switch at a limit position the switch is activated and sends a signal to the controller to stop or change the operation, ensuring safety and accuracy.

Figure 3–13 Limit switch [23]

Operating principle of limit switches: Limit switches work by converting mechanical action into electrical signals to control the system. When a moving part (such as a CNC machine shaft or robot arm) touches the mechanical part of the switch (lever, roller), the internal mechanism is activated, closing or opening the electrical contacts, creating an ON/OFF signal. This signal is sent to the controller, which then executes commands such as stopping the motor, reversing motion or reporting an error, to prevent the device from exceeding its travel limit, avoiding collision or damage.

3.11 DHT 11

The DHT11 temperature and humidity sensor has a built-in 5.1k resistor, making it easier for users to connect and use than the DHT11 sensor without a pin. The module gets data via a 1-wire interface. The signal preprocessor built into the sensor helps you get accurate data without having to go through any calculations. The module is designed to operate at a voltage of 5VDC.

Technical specifications:

- Operating voltage: 5VDC
- Communication standard: TTL, 1 wire.
- Humidity measurement range: 20%-80%RH error ± 5%RH
- Temperature measurement range: 0-50°C error ± 2°C
- Maximum sampling frequency 1Hz (1 second / time)
- Dimensions: 28mm x 12mm x10m

Figure 3–14 DHT11 [24]

3.12 Solar battery

Figure 3–15 Solar [25]

Solar cell technology (photovoltaic cells):

Operating principle:

Photovoltaic effect: When sunlight (photons) shines on a semiconductor material (usually silicon), electrons in the material are excited, creating an electric current.

Solar cells consist of many photovoltaic cells (solar cells) that are connected together. Each cell generates a small amount of electricity, which when combined will provide enough electricity.

Solar cell structure:

Photovoltaic cells: Main components, usually made of crystalline silicon (monocrystalline or polycrystalline) or thin film materials (such as CdTe, CIGS).

Anti-reflective layer: Reduces light loss, helping to absorb more photons.

Protective glass: Protects cells from environmental impacts (rain, dust, impact).

Background plate and frame: Provide a sturdy and insulating structure.

Electrical connector: Transmits current from the battery to the system for use or storage.

Benefits of using solar panels:

Create green energy: Sunlight is considered a natural and quite familiar energy source. Through the solar panel, this natural energy source is converted into electricity to serve human life. Reduces the burden on the national power grid which is always overloaded, helping to overcome the shortage of electricity for production and living needs, without interruption during use.Savings: The price of solar panels is quite cheap, along with the cost of installing systems for households, companies, factories... is moderate. With low investment capital but the efficiency of use is very large, helping to reduce costs from the installation system, no maintenance, durability up to more than 25-30 years, helping to save a lot of monthly electricity bills compared to when using the grid power system.

Environmentally friendly: A prominent advantage of solar panels is that they generate clean, environmentally friendly energy, contributing greatly to environmental protection.

3.13 Solar charge controller 12V

Figure 3–16: Solar charger controller [27]

Specifications:

- Support voltage: 12V or 24V (automatic recognition).
- Maximum charging current: 30A.
- Technology: PWM (Pulse Width Modulation).
- USB port: Support device charging (up to 5V, usually 1-2A, depending on design).
- Compatible battery types: Lead-acid, Gel, Lithium (LiFePO4, 14.4V).
- Display: LCD screen for voltage, current, battery status.

- Size and weight: Compact, not specifically shown in the picture but suitable for fixed installation.
- Operating temperature: Usually from 0°C to 50°C (according to conventional PWM standards).

PWM-based chargers work by using variable-width pulses to regulate the charging current into the battery, optimizing performance and protecting the battery. As the battery nears full, the controller gradually reduces the pulse width, reducing the charging current to a safe level, avoiding overcharging or damage.

3.14 Battery

Figure 3–17 *Battery [28]*

Yamato 6-DZF-30 (6-DZM-30) dry battery is a sealed, maintenance-free battery and is one of the best batteries on the market today with a voltage of 12V, 30Ah capacity, suitable for applications requiring stable power supply.

The structure of the battery includes the following main components:

Anode (positive plate): made of lead oxide (PbO2), flatly coated on a lead-calcium alloy frame to increase hardness. The positive plates in a cell are welded together, forming a positive plate cluster.

Anode (negative plate): made of pure lead, coated on a lead alloy frame. The negative plates in a cell are also welded together, forming a negative plate cluster.

Electrolyte solution: The positive and negative plates are connected in parallel, separated by an insulating layer and immersed in an electrolyte solution, usually dilute sulfuric acid (H2SO4), which can be in liquid or gel form.

3.15 DC-DC 5A step down module XL4015

Figure 3–18 XL4015 [29]

The XL4015 5A DC-DC Step-Down Module is an efficient DC-DC voltage converter circuit, using the XL4015 IC, which is specialized for step-down with a maximum output current of 5A (4.5A is recommended for stability). The module has an input voltage of 8V to 36V and an adjustable output of 1.25V to 32V, with a conversion efficiency of up to 96%, low ripple (about 50mV), and an operating frequency of 180KHz, which reduces noise and ensures stable output. The module is compact in size (about 54 x 24 x 15 mm), with a built-in potentiometer for voltage and current adjustment, along with a status indicator and a built-in heat sink.

3.16 LTC3780 voltage regulator circuit

LTC3780 voltage regulator circuit is a high power DC-DC Buck-Boost (step-up and step-down) voltage converter module, designed based on Linear Technology's LTC3780 IC.

Figure 3–19 LTC3780 [29]

Specifications:

- Input voltage: 5 - 32VDC (can fluctuate continuously).
- Output voltage: 1 - 30VDC (Always fixed at 1 level).
- Max output current: 10A.
- For long-term use, use at 6 -> 8A, above this level, upgrade the heatsink).
- Output power: 80W, max 130W.
- Size: 77.6 x 46.5 x 15mm.

3.17 High power laser

The fiber laser module is made of a sturdy aluminum or copper metal shell to protect the internal components and help dissipate heat effectively. Inside, the main component is a phased optical fiber that amplifies the laser light. High-power pump laser diodes emit light at a wavelength of 915 nm or 976 nm, which is fed into the optical fiber through a fiber coupler. The module also integrates an optical filter (Optical Isolator) to prevent back-reflected light from damaging the system, along with an electronic circuit that controls the pump current, monitors temperature, and protects the device.

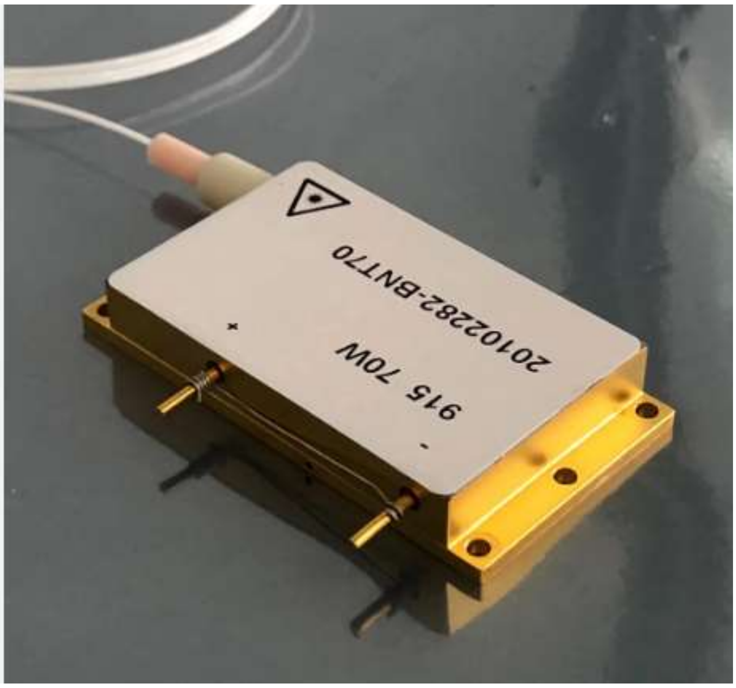

Figure 3–20 Laser 70W [30]

Specifications:

- Standard Specifications: 70W Fiber Laser:
- Output power: 70W (continuous stable)
- Center wavelength: 915nm

- Operating wavelength range: 900 ~ 1080 nm
- Optical fiber type Core size 6 µm ~ 10 µm (single-mode) or 20~25 µm (multi-mode).
- Supply voltage 24VDC or 48VDC
- Consumption current About 8A ~ 15V.

The working principle of a fiber laser begins with a pump source, usually a laser diode, that emits light (usually at a wavelength of 980 nm or 1480 nm) to excite ions in an optical fiber doped with rare earth elements such as Erbium or Ytterbium. The pump light causes these ions to move to a higher energy state, which then emits photons through stimulated emission, producing laser light of the same wavelength, phase, and direction. In the laser module, the light is amplified by continuous reflection inside the fiber core.

CHAPTER 4 SYSTEM DESIGN

4.1 Mechanical system design

4.1.1 Calculation parameter

4.1.1.1 Calculation to select motor

Necessary parameters of the robot.

Table 4–1 Operating parameters

No	Characteristic	Parameter
1	Mass	30-50 kg
2	Wheel diameter	20 cm
3	Speed	0.25 m/s
4	Gear ratio	2
5	Rolling friction	0.3

The coefficient of friction is selected based on reference information from "Machinery's Handbook Engineering Toolbox".[4][34]

Wheel Material	Surface Material	Rolling Friction Coefficient (μ_r)
Steel	Steel	0.001 - 0.002
Steel	Wood	0.002 - 0.005
Hard Rubber	Dry Concrete	0.015 - 0.035
Hard Rubber	Dry Soil	0.04 - 0.08
Soft Rubber	Dry Concrete	0.02 - 0.04
Soft Rubber	Soft Soil (Agricultural)	0.05 - 0.15
Wood	Wood	0.01 - 0.02
Hard Plastic (PU)	Concrete	0.01 - 0.03
Steel (Ball Bearing)	Steel	0.0005 - 0.0015
Soft Rubber	Wet Soil	0.1 - 0.3

Figure 4–1 The coefficient of friction

From the parameters on the table, we can calculate as follows[2]:

Rotation speed of the wheel:

$$\omega = \frac{v}{r} = \frac{0.25}{0.1} = 2.5\left(\frac{rad}{s}\right) (4.1)$$

Revolutions of the wheel:

$$n = \omega.\frac{f}{2\pi} = \frac{2{,}5 \times 60}{2\pi} = 23.9\ (rpm)\ (4.2)$$

With a gear ratio of 3, the motor's rotation speed is 3 times that of the wheel:

$$\omega_{DC} = 2{,}5 \times 2 = 5\left(\frac{rad}{s}\right) (4.3)$$

Revolutions of the motor:

$$n_{DC} = \omega.\frac{f}{2\pi} = \frac{5.60}{2\pi} = 47.7\ (rpm)\ (4.4)$$

Motor rotates at speed 48 (rpm)

Force of the vehicle:

$$F = m.g.\mu = 50 \times 9.8 \times 0.3 = 147\ (N)\ (4.5)$$

Torque at each wheel:

$$M = \frac{F.r}{2} = \frac{147 \times 0.1}{2} = 7.35\ (N.m)\ (4.6)$$

Torque at each wheel:

$$M_{DC} = \frac{7.35}{2} = 3.675\ (N.m)$$

To have a safety margin (the ground may be rough or the vehicle needs to accelerate), choose a motor with a minimum torque of about 5-7 Nm.

Input power of the motor:

$$P = \frac{M.\omega}{n_1.n_2.n_3} = \frac{3.675 \times 5}{0.8 \times 0.9 \times 0.99} \approx 25.77\ (W)\ (4.7)$$

To be safe (farmland conditions can vary), choose a motor with a power of 50-60 W.

Choose the right motor:

Rotation speed: 50-70 RPM.

Torque: ≥ 5 - 7 Nm.

Power: ≥ 50-60 W

Voltage: 12V or 24V (common for robots).

4.1.1.2 Calculation for transmission

Determine the parameters of the chain and chain drive:

Select the number of sprocket teeth:

The gear ratio u = 2 is defined by:

$$u = \frac{N_{large}}{N_{small}} = 2 \ (4.8)$$

where:

N_{large}: Number of teeth of the large sprocket (mounted to the wheel axle).

N_{small}: Number of teeth of the small sprocket (mounted to the engine).

To keep the sprocket running smoothly:

Small sprocket: Usually choose 13 - 15 teeth when the vehicle is operating at low speed (to reduce chain tension and increase life).

Large sprocket: N_{large}=2. N_{small}

Determine the chain pitch P

Pitch is the distance between the links, usually chosen based on load and power:

Transmission power: ~50-60 per motor (total 120W for 2 wheels).

Motor torque: ~5-7 Nm.

Speed: 50 RPM.

With such a small load, you can use a standard chain such as:

Chain No. 35 (35H): Chain pitch 9.525 mm (3/8 inch), Maximum Working Load (~1500-2000 N) (depending on manufacturer, e.g., Tsubaki, Renold).

Chain No. 40: Chain pitch 9.525 mm (3/8 inch), stronger, suitable if you want to overload.

Choose chain No. 40 because of its low power and torque, saving cost and weight.

Sprocket diameter:

Sprocket outer diameter (pitch diameter) is calculated:

$$D = \frac{P \times N}{\pi} \ (4.9)$$

where:

P: Sprocket pitch (mm).

N: Number of teeth.

Small sprocket (12 teeth, P = 12.7 mm):

$$D_{small} = \frac{12.7 \times 12}{\pi} = 48.5\ (mm)$$

Large sprocket (28 teeth, P = 12.7 mm):

$$D_{large} = \frac{12.7 \times 28}{\pi} = 113.2\ (mm)$$

Axle spacing and number of chain link.

The distance between the motor shaft and the wheel shaft (center distance, C) needs to be large enough so that the chain does not sag and is easy to install:

Typically, C=30−50 chain pitches

With a 40 chain (P = 12.7 mm): C≈381−635 mm

Choose C=40 cm (400 mm) for compactness and fit on small vehicles.

Chain length (number of links, L):

Approximate formula:

$$L = \frac{2C}{P} + \frac{N_{large} + N_{small}}{2} + \frac{(N_{large} - N_{small})^2}{4\pi^2 \times \frac{C}{P}}\ (4.10)$$

$$=> L = \frac{2 \times 400}{6.35} + \frac{28 + 12}{2} + \frac{(28 - 12)^2}{4\pi^2 \times \frac{250}{6.35}} = 83.206\ (Chain)$$

Select L = 84 chain link (Chain length = 84 x 12.7 = 1066.8 (mm)).

Check the load through the chain:

Force transmitted through the chain:

$$F = \frac{T_{DC}}{r_{small}} = \frac{5}{(48.5/2) \times 10^{-3}} = 206.2(N)\ (4.11)$$

Force: 206.2 N is far below the #40 chain's maximum working load (2500-3000 N).

Power: Motor power (50-60 W) is well below the chain's capacity (~1800-2200 W at 100 RPM).

Conclusion: ANSI #40 chain is robust, easily handling the load and power requirements.

We have the chain transmission parameters:

Small chain: 12 teeth, diameter ~48.5 mm.

Large chain: 28 teeth, diameter ~113.2 mm.

Chain pitch: 12.7 mm (chain number 40).

Axle distance: ~400 mm (40 cm).

Chain length: ~84 links (~1067 mm).

4.1.1.3 Preliminary calculation of X,Y axis structure.

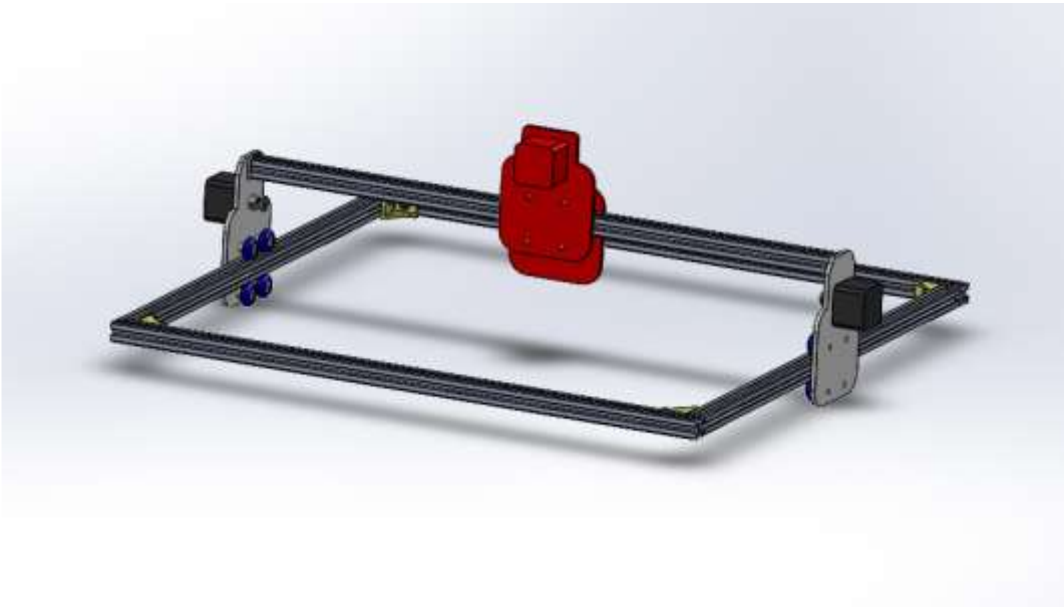

Figure 4–2 X,Y axis

Calculate the approximate force to move the Y cluster:

Here the group will choose a performance: **0.9**

The reasons are as follows: - In a belt drive, the friction that occurs between the belt and the pulley is the main factor causing energy loss. This is necessary to transmit torque from the main pulley motor to the pulley device.

The belt does not always grip the pulley completely. When the load increases or the yellow belt, slippage occurs, resulting in the fact that some of the motion cannot be transmitted from one pulley to the other.

When the belt rotates at high speed, the air resistance and inertia of the belt itself also consume some energy, although this speed is usually smaller than other factors.

Ignore coefficient of friction

Load mass: $m = mX + mt \approx 0.5 + 0.5 = 1$ (kg) (4.12)

In which: m ≈ 1 (kg): mass of Y-axis mechanism assembly

mX ≈ 0.5 (kg): mass of Y assembly located above.

mt ≈ 0.5 (kg: Mass of equipment to load.)

Table 4–2 Y cluster parameters

No	Characteristic	Parameter
1	Mass(kg)	m = 1.0 (kg)
2	Speed	v = 0.5 (m/s)
3	Acceleration of gravity	g = 10 m/s^2

Calculate the approximate force to move the Y cluster:[1]

$F = m \times a = 10 \times 1 = 10N$ (4.13)

Calculating the torque required for maneuvering.

Assume the pulley diameter (D): Temporarily select a pulley diameter of 20 mm (0.02 m) Pulley radius $(r) = \frac{D}{2} = 0.01$ m (4.14)

Identify the formula in the document [1]

Torque (T) : $T = F \times r = 10 \times 0.01 = 0.1$ Nm (4.15)

Adjust according to efficiency (η)

Actual torque required = T / η

T_actual = 0.1 / 0.9 = 0.1(1) Nm (4.16)

Motor selection: The motor is usually selected as a stepper motor. We need to choose a motor with a torque larger than the calculated value, usually 1.5 to 2 times larger.

=> $T_{actual} = 0.1(1) \times 2 = 0.2(2)$Nm (4.17)

Calculate the approximate force to move the X cluster:

Load mass: $m = mt \approx 0.5$ (kg)

In which: mt ≈ 0.5 (kg): Mass of equipment to load.

Table 4–3 X cluster parameters

No	Characteristic	Parameter
1	Mass(kg)	m = 0.5(kg)
2	Speed	v = 0.5 (m/s)
3	Acceleration of gravity	g = 10 m/s^2

Calculate the approximate force to move the X cluste:

$$F = \text{m} * \text{a} = 0.5 * 10 = 5 \text{ (N)} \quad (4.13)$$

Calculating the torque required for maneuvering:

Assume the pulley diameter (D): Temporarily select a pulley diameter of 20 mm (0.02 m)

Pulley radius $(\text{r}) = \frac{\text{D}}{2} = 0.01 \text{ m} \ (4.14)$

Torque (T): $(\text{r}) = \text{D}/2 = 0.01 \text{ m}$

Adjust according to efficiency (η).

Actual torque required = T / η

$\text{T\_actual} = 0.05 / 0.9 = 0.05(5) \text{ Nm} \ (4.15)$

Motor selection

The motor is usually selected as a stepper motor. We need to choose a motor with a torque larger than the calculated value, usually 1.5 to 2 times larger.

$\text{T}_{\text{actual}} = 0.5(5) \times 2 = 0.1(1)\text{Nm} \ (4.16)$

A stepper motor is a special type of electric motor that operates on the principle of dividing rotational motion into discrete steps, allowing precise control of position, speed and direction of rotation without the use of a feedback sensor (encoder). Each control signal sent to the motor will cause the shaft to rotate a certain angle (called a step angle), . There are many different types of stepper motors such as 2-phase, 4-phase, 5-phase, with control types such as full-step, half-step or microstepping, which helps increase the smoothness of the movement. The big advantages of stepper motors are ease of control, the ability to hold position when not rotating and can operate stably at low speeds.

Table 4–4 Engine specifications

Parameter	NEMA 17	NEMA 23
Faceplate Size	42 x 42 mm	57 x 57 mm
Common Step Angle	1.8° (200 steps/rev)	1.8° (200 steps/rev)
Torque (Nm)	0.3 - 0.5 Nm	1.0 - 3.0 Nm
Max Speed (RPM)	1000 - 2000 RPM (no load)	800 - 1500 RPM (practical)
Current(A)	1.0 - 2.0 A	2.0 - 4.0 A
Voltage (V)	12-24 V	24 - 48V

➢ **Motor selection:** With a torque of about 0.2(2) Nm and a speed of a NEMA 17 stepper motor (usually with a torque of about 0.3 - 0.5 Nm at low speed) is suitable.

4.1.1.4 Belt length selection calculation

Features of GT2 belt loop.

High quality materials: The belt loop is made of high quality materials such as rubber and reinforced thin PU fibers to ensure precision and durability in use.

The serrated belt surface is precisely machined to increase grip and reduce slippage.

High durability: The belt loop can operate continuously for a long time without frequent maintenance.

High speed transmission capability: GT2 belt loop has high speed transmission capability and the ability to reduce vibration when the machine is operating.

Easy installation: GT2 jute belt loop has a simple design and is easy to install on various machines and equipment.

Using rounded profiles ensures that the teeth fit tightly and accurately, helping 3d printers, cnc, lasers operate stably.

GT2 belt loop is designed for linear circular motion.

The group calculates the operating range of the base configuration as: 60*60(cm)

Estimated total length: 60 + 60 + 60 =180 cm.

4.1.2 Mechanical 3D design

4.1.2.1 Software for 3D design

SOLIDWORKS is a 3D design software running on Windows operating system of Dassault Systèmes of France. Known for its ease of use and intuitiveness, SOLIDWORKS is a powerful 3D design software and provides users with the best features to design 3D solid details, assemble details to form parts of production machinery.

Figure 4–3 Solid Works [33]

SolidWorks Main Functions:

3D Modeling: SolidWorks allows you to create 3D model details with features such as extrusion, rotation, lofting, and sweeping. Users can design parts, assemblies, and technical drawings visually. The software uses parametric modeling, which automatically updates the entire model when a part changes.

2D Design and Technical Drawings: SolidWorks supports the rapid creation of 2D drawings from 3D models, including dimensions, tolerances, and technical notes. Users can easily export drawings for use in products.

Assembly and Assembly Management: The software allows you to assemble multiple parts into a complete set, simulating how the parts interact with each other. Features like “mate” help to precisely position parts, check and touch and motion.

Simulation and analysis (Simulation: SolidWorks integrates simulation tools to analyze the application, motion, temperature and performance of the design under real-world conditions. This helps to detect errors and optimize designs before manufacturing.

4.1.2.2 3D Model of robot

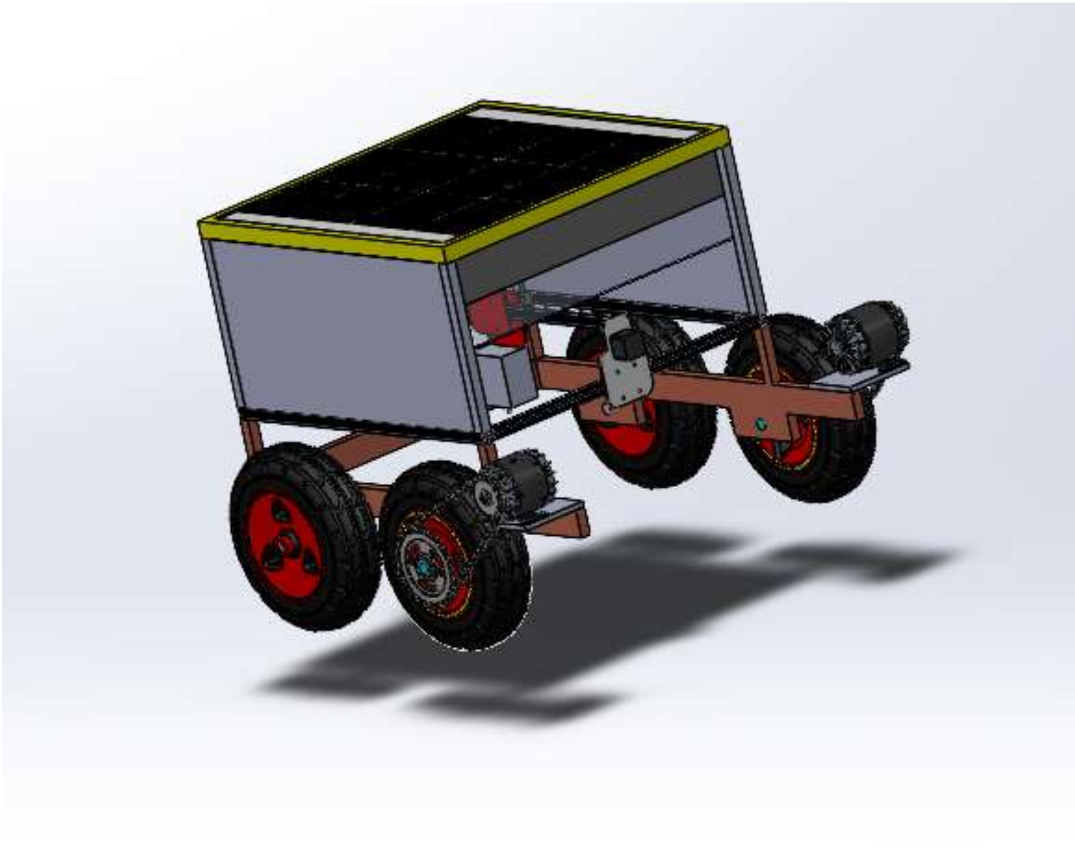

Figure 4–4 3D robot

The team used SolidWorks, a widely used 3D computer-aided design (CAD) software, to create detailed engineering models of the robot pictured here – a mobile device with a chassis, wheels, motors and solar panels. They then used assembly tools to create the model shown above.

The image depicts a robot with a boxy design with four wheels, suitable for ground-moving applications. The upper part of the robot is equipped with a black solar panel, which serves as the main power source. The robot body is made of gray metal or plastic, with a sturdy structure to accommodate the internal components. The four wheels, each of which is attached to its own motor. The transmission system consists of a belt or chain connecting the motor and the wheels, ensuring effective traction. The lower part has a metal frame to support it, while technical details such as wires and electronic components are arranged inside or near the motor. This design suggests that the robot can be used for agricultural purposes of autonomous transportation.

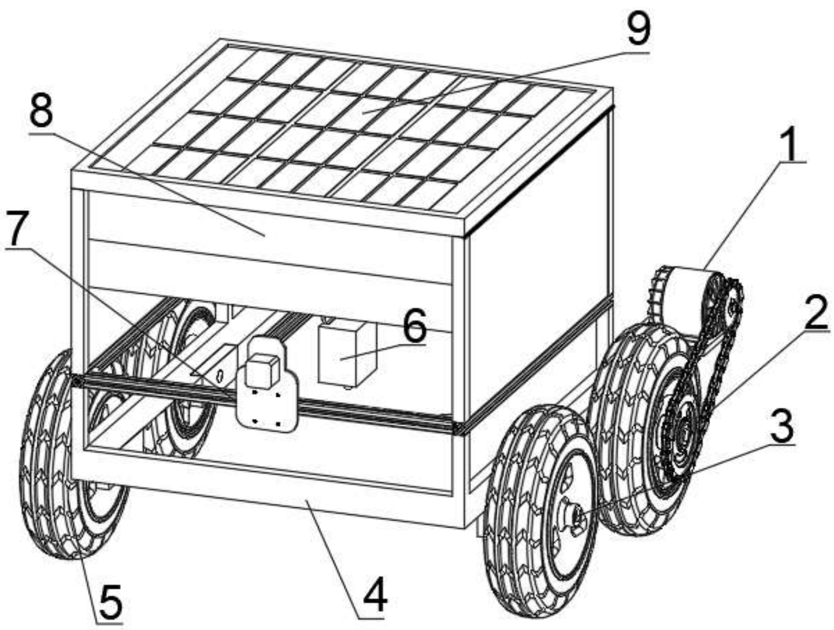

Figure 4–5: Components of a robot

The robot has the following components:

(1) The DC motor provides rotational force to drive the wheel, helping the robot move back and forth, controlling position.

(2) Chain drive: Connects the engine to the wheel, transmitting force from the engine to the wheel to create movement, increasing torque.

(3) Axis: Connect, transmit power, and bear the vehicle's load.

(4) Frame: Helps fix the body of the car and attach accessories to the car.

(5) Tires: Create rotational motion, helping the vehicle move forward.

(6) Laser: Uses electricity to convert into laser beam to burn grass.

(7) Executive structure: 2-axis X,Y cnc actuator to bring to the exact position.

(8) Compartment: Contains circuits, batteries, necessary components.

(9) Solar Panel: Provides energy for Robot operation.

4.2 Algorithm flow chart

4.2.1 Main program

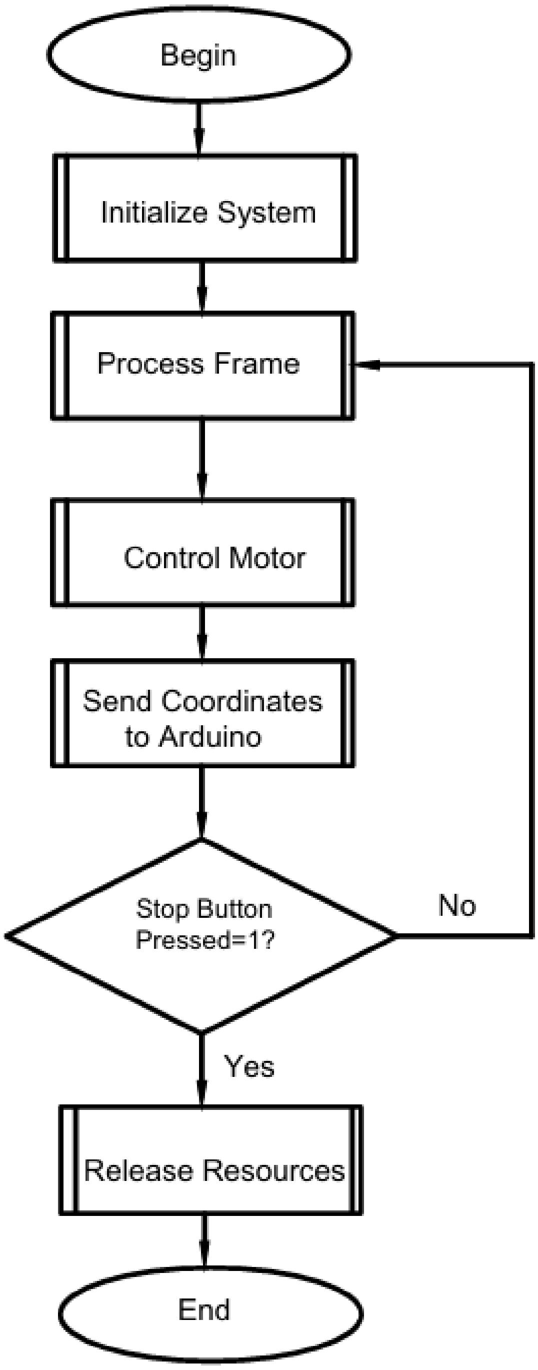

Figure 4–6 Main program

The main program consists of five subprograms that execute sequentially, the system will operate in a loop after finishing the fourth subprogram and return to the second subprogram. If the stop button is pressed, the system will move to the last subprogram and end the program. The execution of the subprograms will be explained specifically in the section below.

4.2.2 System initialization

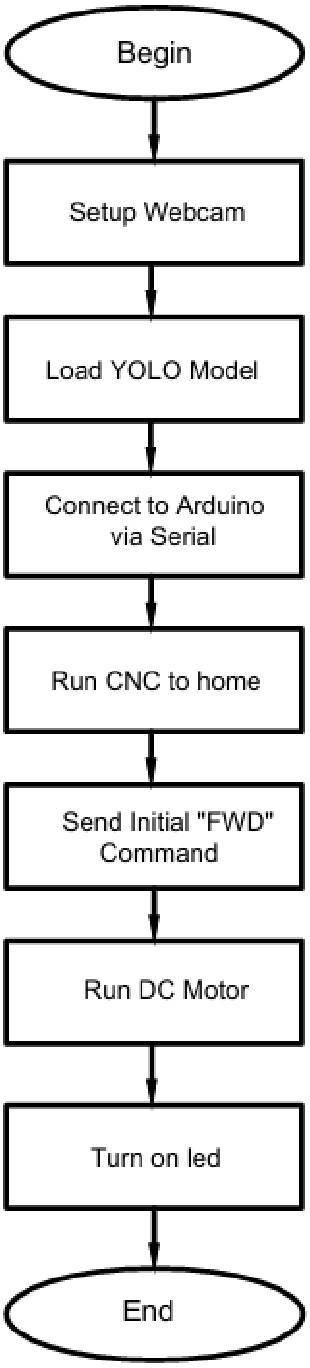

Figure 4–7 Intializiton the system

When starting the system, the first subroutine will be executed, here will set up the initial systems of the robot such as starting the webcam, loading the yolo model to recognize and predict objects via webcam, connecting the com port then the program sends the command to control the CNC axis to move to the Home coordinate, then sends the 'FWD' signal from python to arduino to control the movement of the vehicle, the LED light function is also turned on to ensure light for the AI model to easily recognize and detect crops.

4.2.3 Process frame

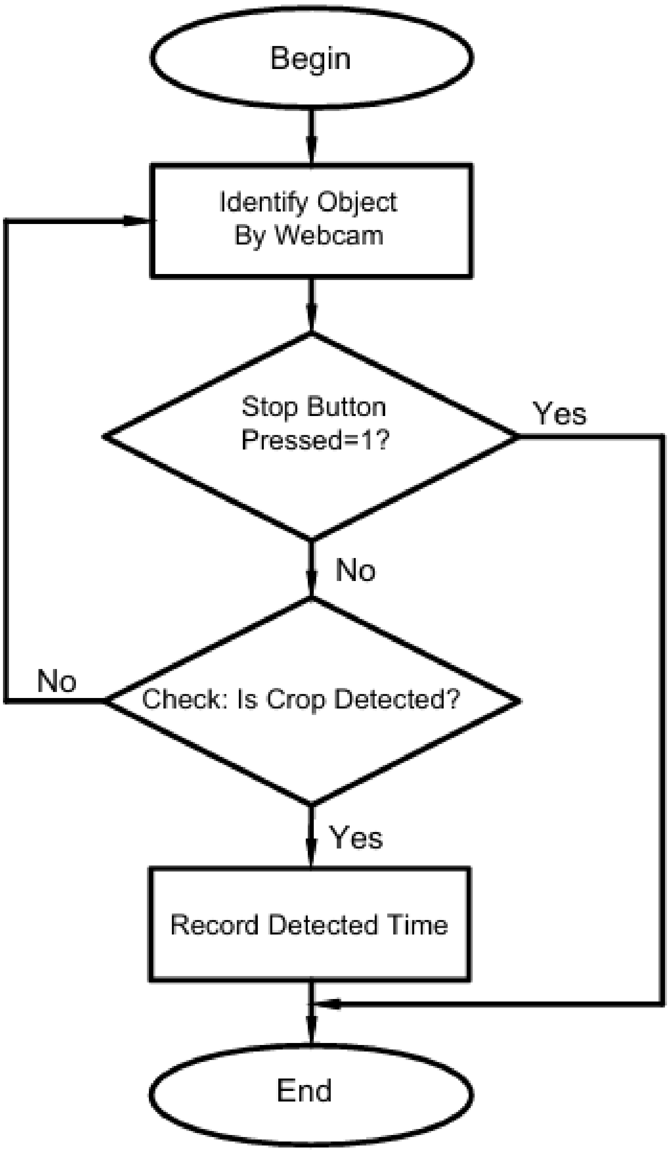

Figure 4–8 Process frame

The second subroutine handles the logic that if the AI model detects a plant then the counter variables are set, preparing the logic for the third subroutine.

4.2.4 Control motor

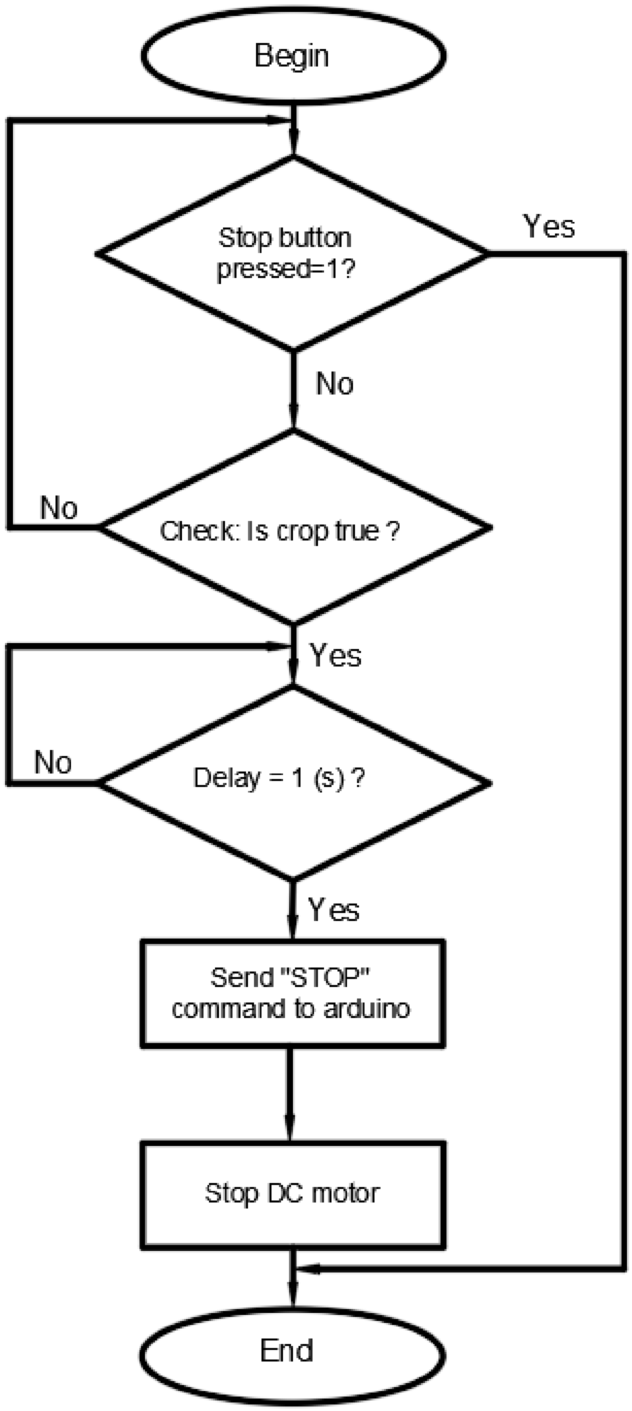

Figure 4–9 Control motor

The second subroutine handles the logic after the AI model detects a plant and the counter set in the previous program reaches the required level then sends a 'Stop' signal to the Arduino to stop the Robot.

4.2.5 Arduino coordinate transmission

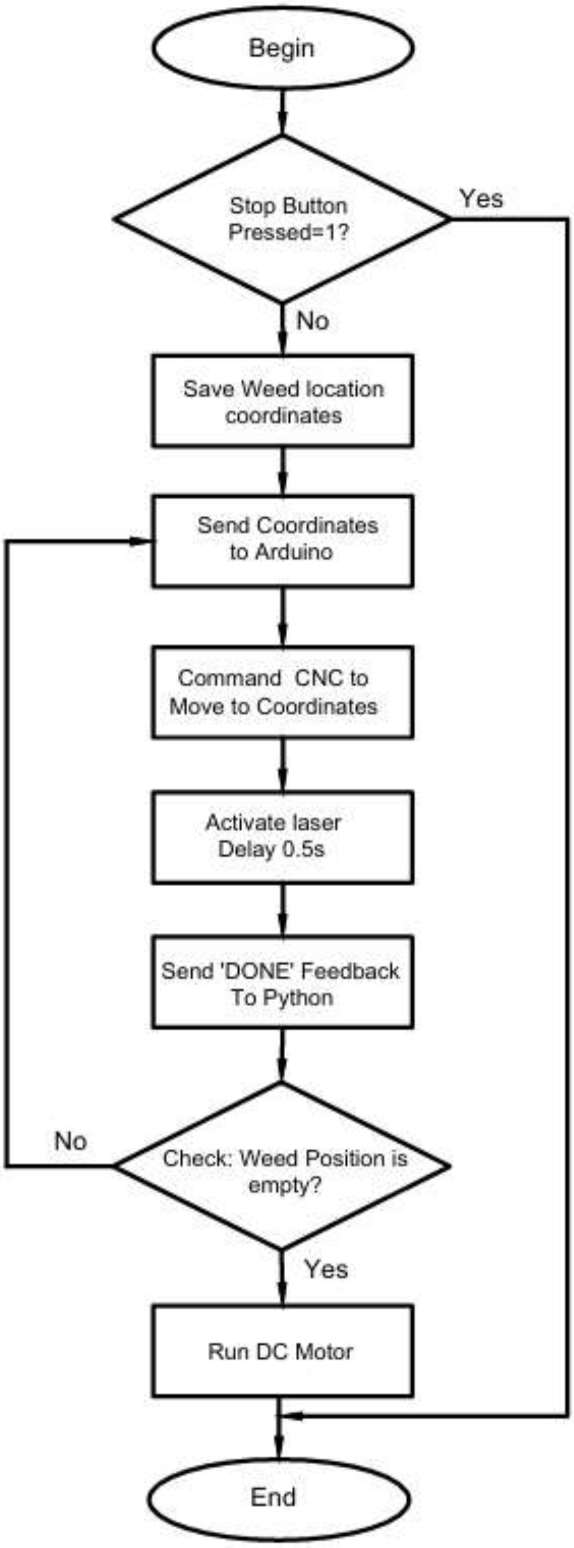

Figure 4–10 Send coordinates

The fourth subroutine takes on the role of processing and sending the coordinates of each grass position to the arduino in turn. After the arduino receives the grass position coordinates, it will control the CNC axis to bring the laser beam to the correct position to destroy the grass. After processing, the arduino will send a "Done" signal back to python to let it know to send the grass positions that need to be processed next. If all the grass positions have been processed, Python will send a signal to the arduino to control the robot to continue moving and here it will return to the loop in the second subroutine.

4.2.6 Release resources

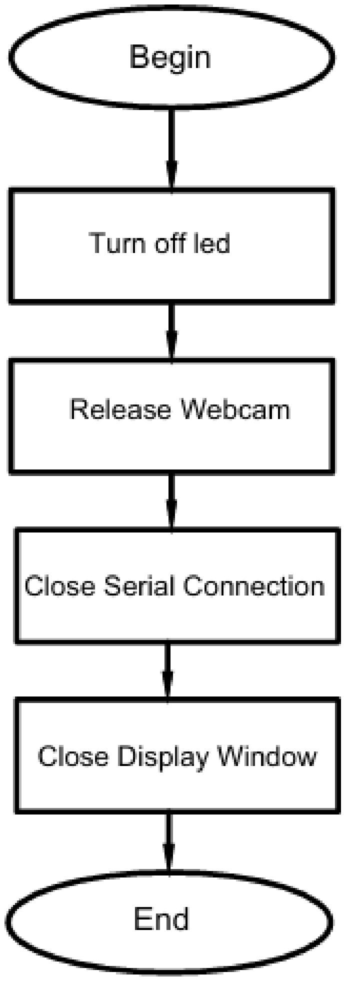

Figure 4–11 Release resources

All running subroutines will be transferred to the fifth subroutine to release resources, close the webcam, disconnect, end the program and stop the Robots.

4.3 Wiring

4.3.1 Sensor block

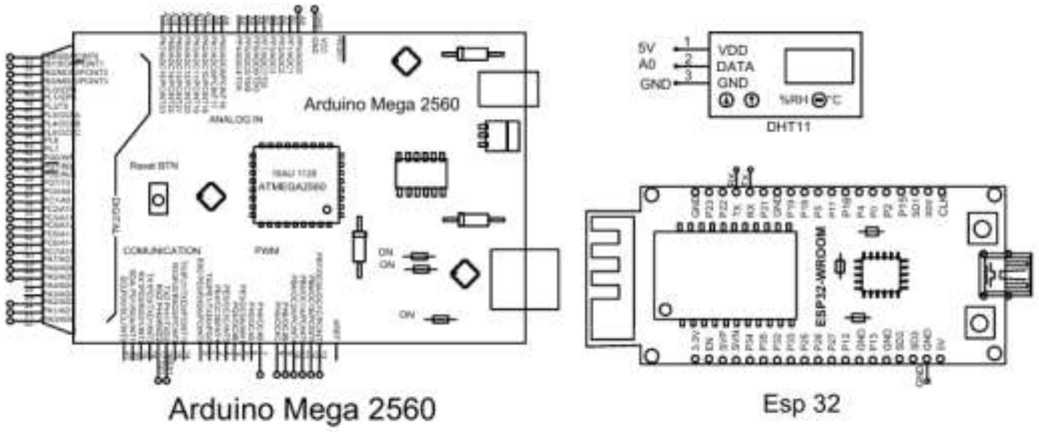

Figure 4–12: Connection the sensor

Here use A0 to connect to the signal pin of DHT11 sensor. Use UART communication standard to transfer data from Arduino to Esp32:

UART (Universal Asynchronous Receiver-Transmitter) is an asynchronous serial data transmitter and receiver, this is one of the most widely used device-to-device communication protocols. Data is transmitted and received over these lines in the form of data frames with a standard structure, with a start bit, a number of data bits, a parity bit, and one or more stop bits.

Advantages:

- Only 2 data transmission lines are needed
- No clock signal is needed
- There are 2 parity bits so errors can be checked easily
- The data packet structure can be changed as long as both sides are set up to communicate with each other
- The Uart communication method has many instructions and is also a widely used data transmission method today

Disadvantages:

- The maximum data frame size is limited to 9 bits, which is quite small compared to the needs of use
- Not supported by many master and slave systems.

4.3.2 Dc control block

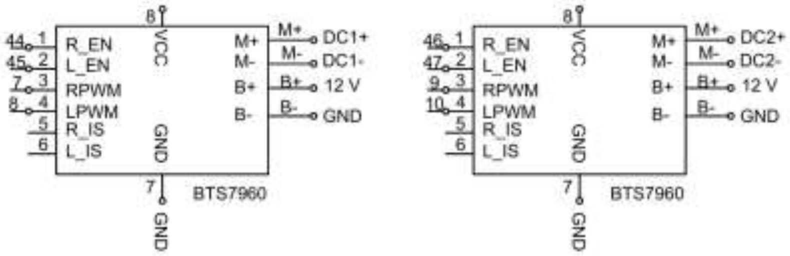

Figure 4–13 DC control

For BTS7960 drive, use 2 Mega 2560 pulse pins to supply RPWM and LPWM, supply 5V high level to 2 Enable pins. On the M+, M- pin side, connect to DC motor, B+, B- supply power at a certain level.

4.3.3 Step control block

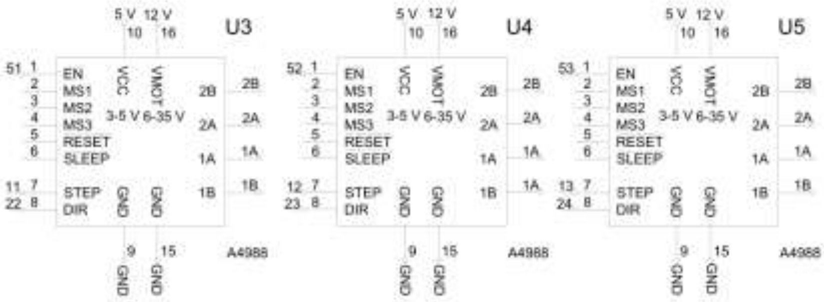

Figure 4–14 Step motor

Control Connections

STEP pin: Connect to Arduino pin of Arduino. Each pulse on this pin makes the motor rotate 1 step.

DIR pin: Connect to Arduino digital pin. Decides the rotation direction.

ENABLE pin: Connect to digital pin or leave blank (default on). LOW to enable the driver, HIGH to disable.

SLEEP and RESET pins: Connect these 2 pins together to control normal operation.

MS1, MS2, MS3 pins: Connect to GND or VCC (5V) to select microstep mode (full step, 1/2, 1/4, 1/8, 1/16). For example:

Full step: MS1, MS2, MS3 = LOW.

Step 1/16: MS1, MS2, MS3 = HIGH.

4.3.4 Wiring connection

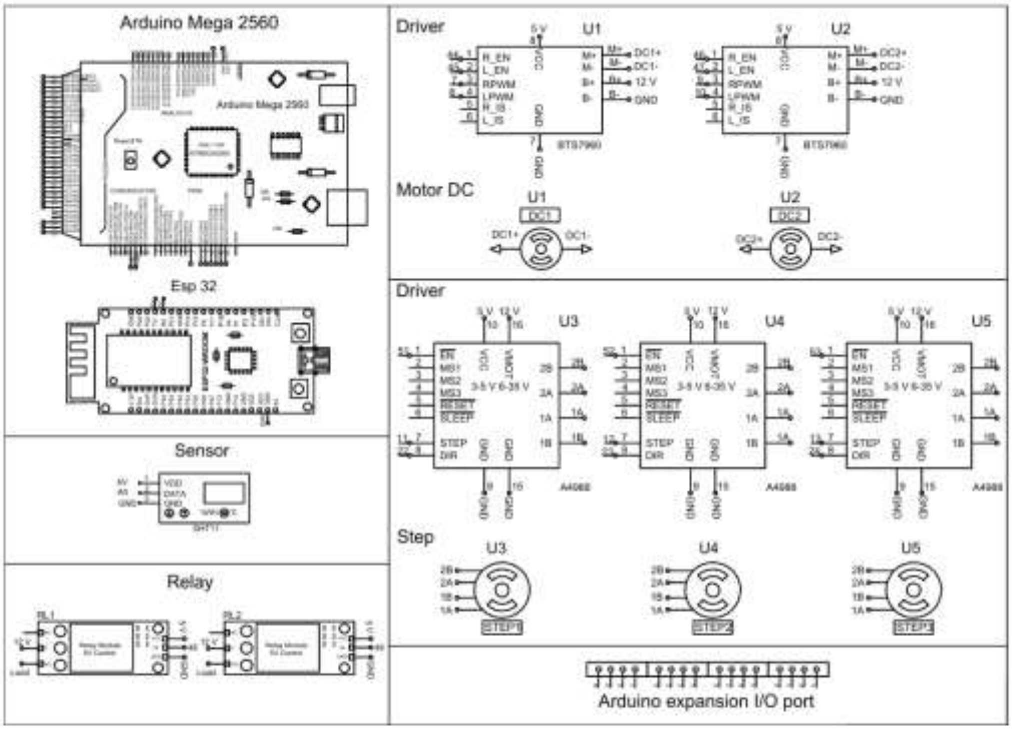

Figure 4–15 Overall

Here is the overall wiring diagram of the project:

Arduino Mega 2560: Is the main control center controller with many expansion pins, many pulse output pins suitable for the project

ESP32: Wi-Fi/Bluetooth module, used for wired connection to the Mega controller to monitor temperature and humidity.

Sensor: Use DHT11 (temperature and humidity sensor), connected via data pin/

DC motor: Connect via BTS7960 driver, with EN, PWM, and DC+/DC- pins to control speed and rotation direction.

Stepper motor: Use A4988 driver, connected to STEP and DIR pins to control step and direction, supports up to 3 stepper motors (STEP1, STEP2, STEP3).

Relay: 2-channel relay module (RL1, RL2) to control electrical loads with 12V power supply, can be used to turn on/off peripheral devices. Driver: Including BTS7960 (for DC motor) and A4988 (for stepper motor), ensuring precise and safe control of the motor.

4.4 Control program

The code is added in the appendix below.

CHAPTER 5 IMAGE PROCESSING

5.1 Introduction to AI image processing technology (YOLO)

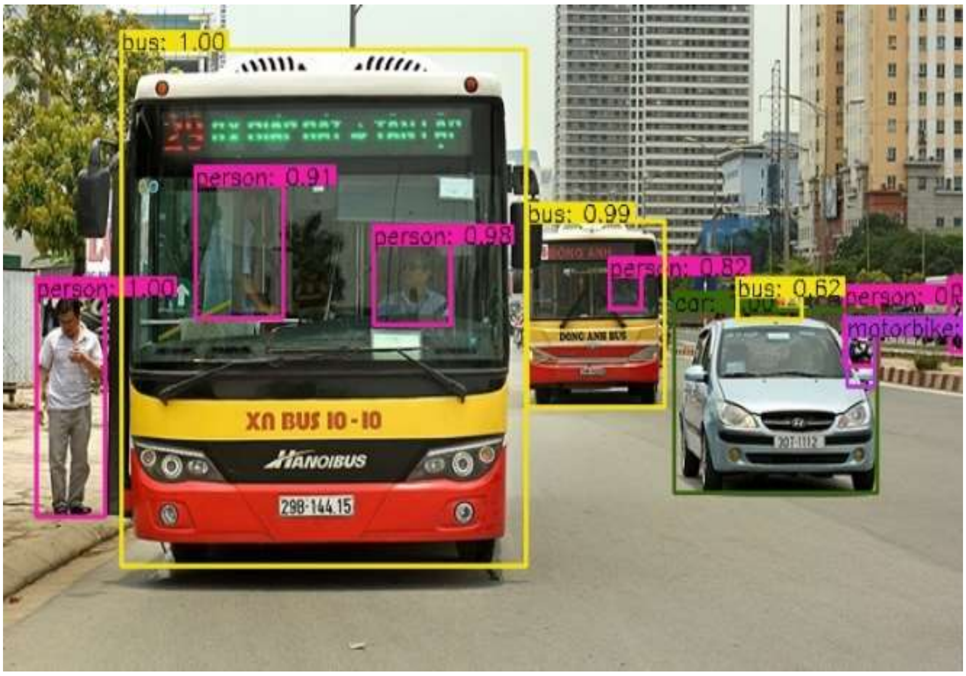

Figure 5–1 Image processing technology YOLO [38]

YOLO (You Only Look Once) means an algorithm that detects all objects in an image/frame in just one search. In terms of accuracy, YOLO may not be the best algorithm but it is the fastest algorithm in the class of object detection models. It can achieve almost real-time speed without losing too much accuracy compared to top models.

YOLO is an object detection algorithm, so the goal of the model is not only to predict labels for objects like classification problems, but it also determines the location of the object. Therefore, YOLO can detect many objects with different labels in an image instead of only classifying one label for an image.

YOLOv8 : The YOLO algorithm divides the starting image into a group of cells and for each cell it predicts the appearance of an object and the bounding box of the object. It can also predict the class of the object.

YOLOv8 is the latest iteration in the YOLO series of real-time object detectors, providing cutting-edge performance in accuracy and speed. Building on the advancements of previous YOLO YOLOv8 version introduces new features and optimizations that make it an ideal choice for various object detection tasks in a variety of applications.

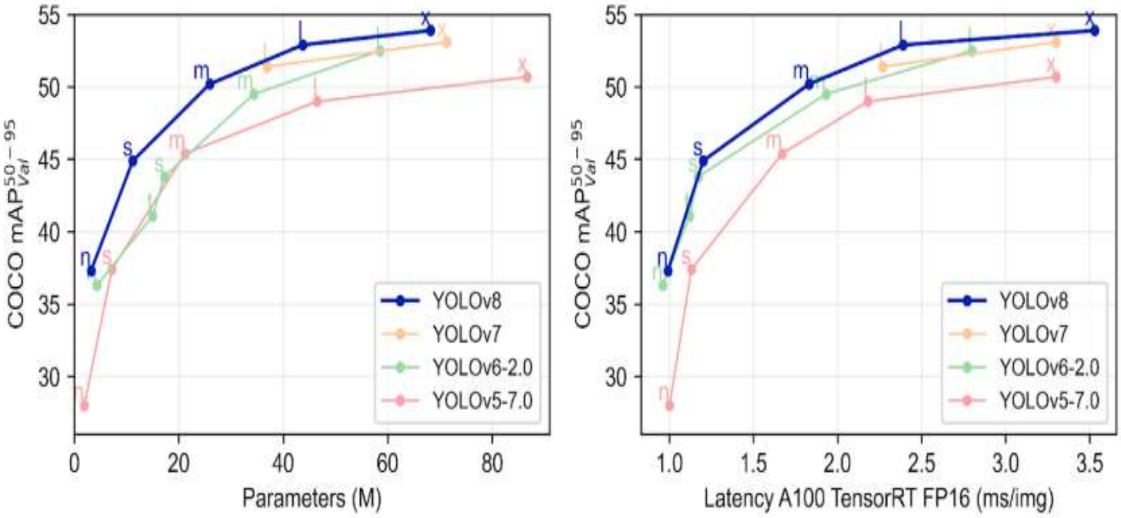

Figure 5–2 Improved performance of YOLOv8

Main features of yolov8:

Advanced Backbone and Neck Architecture: YOLOv8 uses a modern spine and neck architecture, resulting in improved feature extraction and object detection performance.

Head Ultralytics Anchorless Splitting: YOLOv8 applies Head Ultralytics anchorless splitting, which contributes to better accuracy and a more efficient detection process compared to anchor-based approaches.

Optimizing the accuracy-speed trade-off: Focused on maintaining an optimal balance between accuracy and speed, YOLOv8 is suitable for real-time object detection tasks in various fields. diverse application areas.

Multiple Pre-trained models: YOLOv8 Provides a wide range of pre-trained models to cater to different tasks and performance requirements, making it easy to find the right model for your tool use case your body.

5.2 Dataset

The data is divided into two sets: “Weed” product data set and “Crop” product data set.

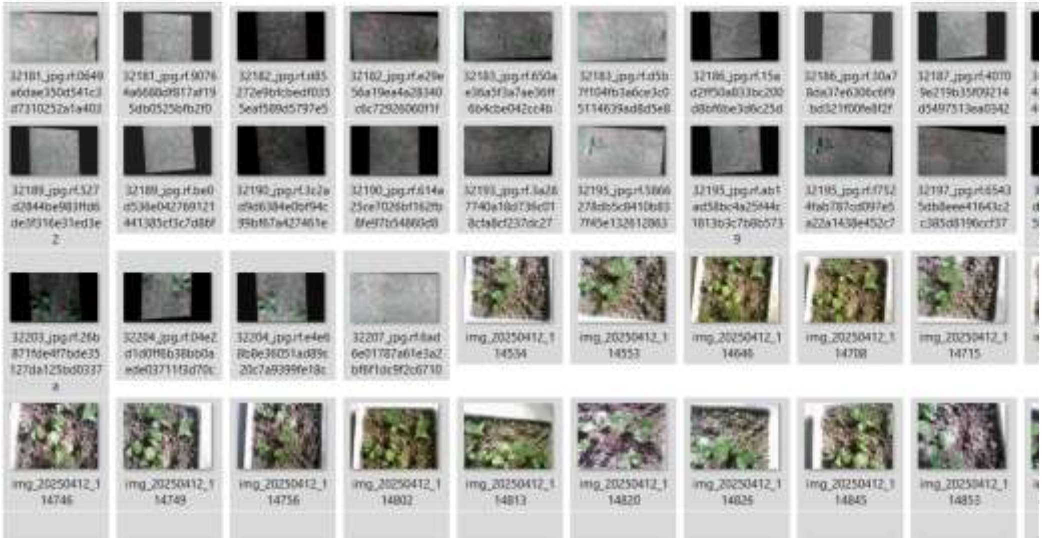

Figure 5–3 Dataset

- Data set Traning:
  - Train: 2520 images
  - Valid: 259 images
  - Size image: 640x640
- Model traning:

```
                   from  n    params  module                                       arguments
  0                  -1  1       928  ultralytics.nn.modules.conv.Conv             [3, 32, 3, 2]
  1                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  2                  -1  1     29056  ultralytics.nn.modules.block.C2f             [64, 64, 1, True]
  3                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  4                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  5                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  6                  -1  2    788480  ultralytics.nn.modules.block.C2f             [256, 256, 2, True]
  7                  -1  1   1180672  ultralytics.nn.modules.conv.Conv             [256, 512, 3, 2]
  8                  -1  1   1838080  ultralytics.nn.modules.block.C2f             [512, 512, 1, True]
  9                  -1  1    656896  ultralytics.nn.modules.block.SPPF            [512, 512, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  1    591360  ultralytics.nn.modules.block.C2f             [768, 256, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  1    148224  ultralytics.nn.modules.block.C2f             [384, 128, 1]
 16                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  1    493056  ultralytics.nn.modules.block.C2f             [384, 256, 1]
 19                  -1  1    590336  ultralytics.nn.modules.conv.Conv             [256, 256, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  1   1969152  ultralytics.nn.modules.block.C2f             [768, 512, 1]
 22        [15, 18, 21]  1   2116822  ultralytics.nn.modules.head.Detect           [2, [128, 256, 512]]
Model summary: 129 layers, 11,136,374 parameters, 11,136,358 gradients, 28.6 GFLOPs
```

Figure 5–4 Model traning

Model overview:

- Number of Classes: 129 classes
- Number of Parameters: 11,136,374 parameters

- Number of Gradients: 11,136,358 gradients
- GFLOPs: 28.6 GFLOPs (Giga Floating Point Operations Per Second)
- Some other parameters:
- lr0 = 0.01, lrf = 0.01: lr0 (initial learning rate): The initial learning rate is set to 0.01, lrf (final learning rate): The final learning rate is also set to 0.01.
- momentum = 0.937, weight_decay = 0.0005: (momentum) This parameter helps speed up the convergence of the optimization process using previous gradients. Momentum is set to 0.937. eight_decay: The weight decay coefficient is 0.0005. This parameter helps reduce overfitting by penalizing large weights during training.
- hsv_h=0.015, hsv_s=0.7, hsv_v= 0.4: These parameters adjust the enhancement of color data in the HSV color space. (hsv_h): Change the (hsv_s): Change saturation, (hsv_v): Change the value.
- Flipud=0.0, fliplr = 0.5: Probability of image flipping during data augmentation.
- mosaic=1.0: The probability of using the mosaic enhancement technique is 1.0, meaning this technique will always be applied. Mosaic is a data augmentation technique by combining multiple images to create a new image.
- iou=0.7: The Intersection over Union (IoU) threshold for non-max suppression is 0.7. IoU is a measure to determine the degree of overlap between two frames. This threshold determines whether two frames are considered co-detected or not.
- Max_det=300: The maximum number of detections per image is set to 300. This limits the number of objects the model can detect in an image.

5.3 Training results and conclusions

Model's prediction rate: (50 epoch):

```
ralytics 8.3.103 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
el summary (fused): 72 layers, 11,126,358 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all        235       1605      0.708      0.746      0.782      0.388
                  crop         17         47       0.66       0.83       0.82      0.417
                  weed        228       1558      0.755      0.663      0.744      0.359
ed: 0.3ms preprocess, 3.3ms inference, 0.0ms loss, 2.3ms postprocess per image
```

Figure 5–5 Model traning result

Model summary (fused):

168 layers: The model has a total of 72 layers.

3006038 parameters: The model has 11,126,358 parameters.

0 gradients: No gradients were calculated. This may be because the model is in inference mode, not training.

28.4 GFLOPs: The model requires 8.1 billion floating point operations per second (Giga Floating Point Operations Per Second) to perform predictions.

Precision (Box(P)): Overall (all): 0.708 → 70.8% of the model's detections are correct, Crop: 0.66 → relatively low; the model misclassifies crops more often. Weed: 0.755 → higher precision; the model detects weeds more accurately.

The model is better at detecting weeds than crops. Overall precision is acceptable, but there's still room for improvement if high precision is required (>90%).

Recall (R): Overall (all): 0.746 → the model detects 74.6% of all real objects. Crop: 0.83 → very good recall; the model misses fewer crops. Weed: 0.663 → lower recall; it misses more weeds.

Crop detection recall is strong (model rarely misses crops).

Weed detection recall is weaker (model misses more weeds).

mAP50 (mean Average Precision at IoU=0.5): Overall: 0.782 → good performance (~78% accurate localization). Crop: 0.82 → very good. Weed: 0.744 → slightly lower than crops.

When allowing a bit of looseness (IoU ≥ 0.5), the model performs quite well.

mAP50-95 (mean AP across IoU 0.5 to 0.95): Overall: 0.388 → relatively low. Crop: 0.417 → higher than weed. Weed: 0.359 → lower.

The model struggles with tight and precise bounding boxes.

This is expected because weeds often have irregular shapes, making exact detection harder.

Conclude:

Strengths:

- Good balance between precision and recall.
- Crop detection recall is strong.
- mAP50 is relatively high, indicating good object localization.

Weaknesses:

- Lower precision for crops → model tends to misclassify crops.
- Lower recall for weeds → model misses more weed objects.
- mAP50-95 is low, meaning bounding box accuracy still needs improvement.

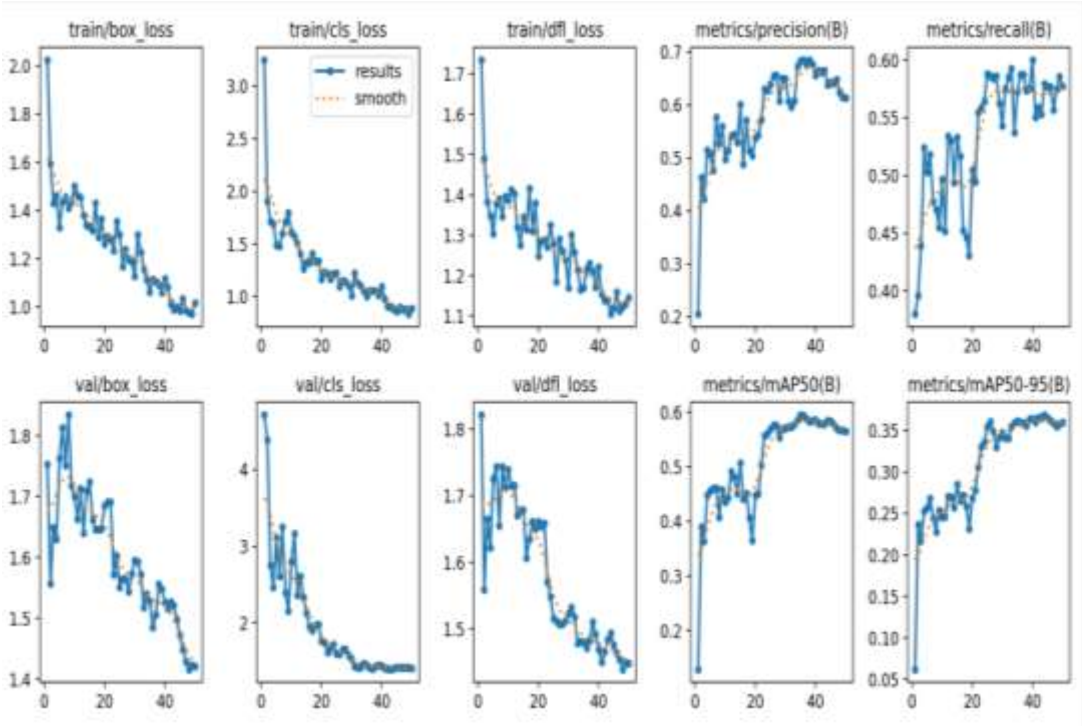

Figure 5–6 Training and validation loss and metrics curves

Train/Box_loss: train/box_loss is a metric that measures the model's ability to predict the correct bounding box during the training phase. This loss gradually decreases, indicating that the model learns better at detecting objects and correctly locating them. - Val/box_loss measures the difference between the predicted bounding box and the actual bounding box on the test data. It is an important metric to evaluate the model's ability to generalize to new data. val/box_loss can fluctuate more, because the test data can be complex and different from the training data.

Train/cls_loss and val/cls_loss similar to Train/Box_loss, Val/box_loss but they are used to evaluate the level of error in object classification during training.

Conclude:

- Train/box_loss, train/cls_loss, train/dfl_los: The loss gradually decreases over epochs, showing that the model is learning to predict bounding boxes more accurately. Improved classification capabilities.

- Metrics/precision(B): Precision increases gradually over epochs, reaching higher values, showing that the model is less confused in its predictions.

- Metrics/recall(B): Recall increases, reaching a high value, showing that the model detects most objects.

Results after the tests:

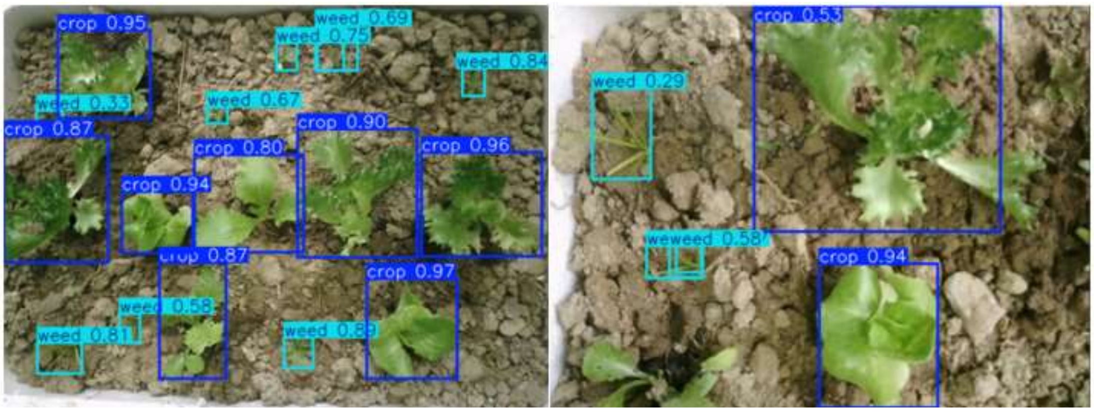

Figure 5–7 Model prediction result

Conclude: Looking at the model's predictions, we see that the model makes predictions with high accuracy even though the products are similar, quite similar, but the model still makes correct predictions. This shows that the model is quite reliable to use.

The image shows an image recognition system used in agriculture to classify crops and weeds, possibly based on artificial intelligence technology. The plants are outlined in blue, labeled "crop" or "weed" and have a confidence score ranging from 0 to 1, reflecting the accuracy of the distinction. Many plants are recognized with high probabilities (such as "crop 0.95" or "weed 0.81"), indicating that the system works well, with crops appearing to dominate both sides of the image. However, some labels have low probabilities (e.g. "weed 0.29", "crop 0.53"), indicating that improvements are needed to increase confidence in difficult-to-distinguish cases.

CHAPTER 6 INTERNET OF THINGS

6.1 Define of IOT (Internet of Things)

The term IoT or Internet of Things refers to a network that consists of smart devices and technologies facilitating convenient communication between devices and the cloud, as well as among devices themselves. Thanks to the emergence of inexpensive computer chips and high-bandwidth telecommunications technology, today we have billions of devices connected to the internet.

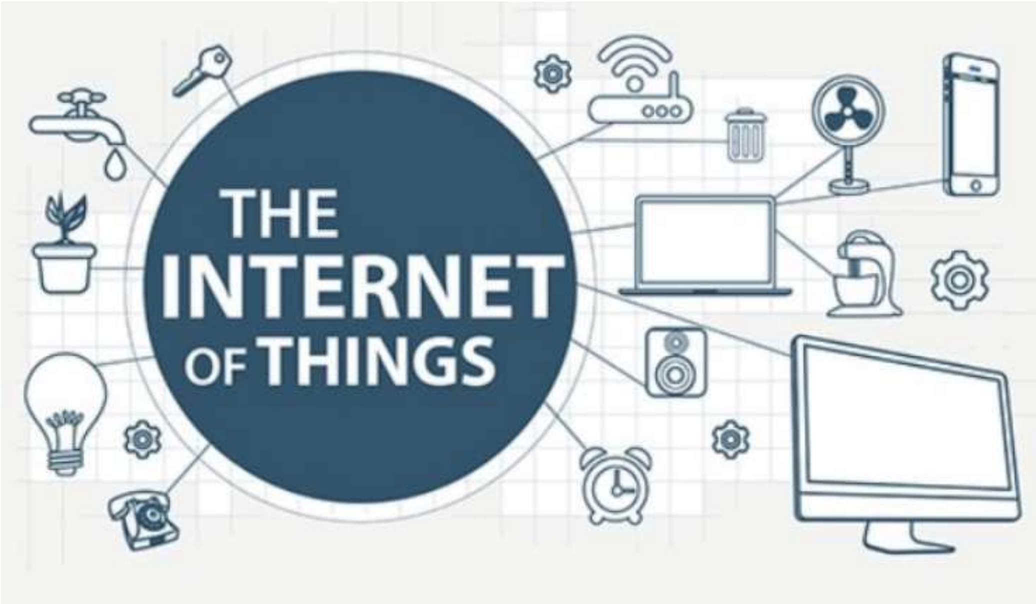

Figure 6–1 Internet of Things [39]

This means that everyday devices such as toothbrushes, vacuum cleaners, cars, and machinery can utilize sensors to gather data and intelligently respond to users.

Figure 6–2 Smart home [39]

IoT applications bring users a more convenient and easier life. Some popular IoT applications in smart homes include:

Lighting system: Smart lights allow remote control anytime, anywhere, saving time and energy. With IoT, you can set up schedules to turn on/off lights for each area in the house without having to go there directly to operate.

Home appliance system: Devices such as air conditioners, water heaters or plant support systems usually need to be started. Thanks to IoT, you can activate them remotely before coming home, ensuring everything is ready as soon as you enter.

Security system: Smart cameras and motion sensors that merge IoT allow you to monitor your home remotely via your phone. You can receive alerts if unusual activity is detected and control smart door locks for added security.

Automatic system: IoT allows remote door adjustment or dynamic opening/closing schedule setting according to time or ambient light, helping to save energy and create a comfortable living space.

IoT Applications In Agriculture:

Continuous weather monitoring: Weather monitoring stations using IoT are a popular solution for monitoring climate conditions. Placed in farming areas, these stations collect data, figures and information to create meteorological maps. Thanks to that, farmers can choose suitable crop and livestock varieties, and at the same time provide optimal care methods, ensuring the quality and productivity of agricultural products.

Figure 6–3 Smart Agriculture [39]

Greenhouse automation: IoT effectively supports greenhouse farming by collecting information on light, temperature, humidity and soil conditions. This data helps to adjust the greenhouse environment to the most ideal, creating favorable conditions for crop growth.

Livestock management and tracking: IoT technology allows monitoring the location of livestock when they are released on pasture. IoT devices attached to livestock also provide information on health and biological indicators, helping to detect diseases early and provide timely treatment measures.

Farm management system: IoT system integrates multiple sensors and monitoring tools across the farm, combined with powerful analytics software. The system collects, aggregates data and generates detailed reports based on agricultural and accounting standards, helping to manage the farm effectively from anywhere and at any time.

6.2 General introduction to firebase

6.2.1 Overview of firebase

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, ios, web, and unity. It provides cloud.

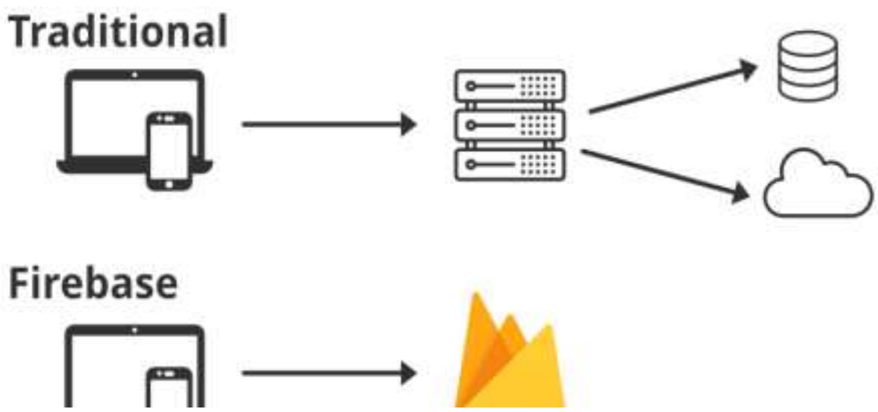

Figure 6–4 Orerview of firebase [40]

Google Firebase includes:

Real-time data - Firebase Realtime Database: just register an account on firebase to create a real-time database to be able to access and adjust data quickly and accurately.

User authentication: Firebase has built functionality for user authentication with Email, Facebook, Twitter, GitHub, Google.

Firebase Hosting: Firebase provides hosting and is delivered via SSL from CDN which will help you save a lot of time in building applications.

6.2.2 Firebase realtime database

By continuously reading data from the Web and an ESP32, Firebase can facilitate easy communication between a website and a microcontroller by reading data from Firebase.

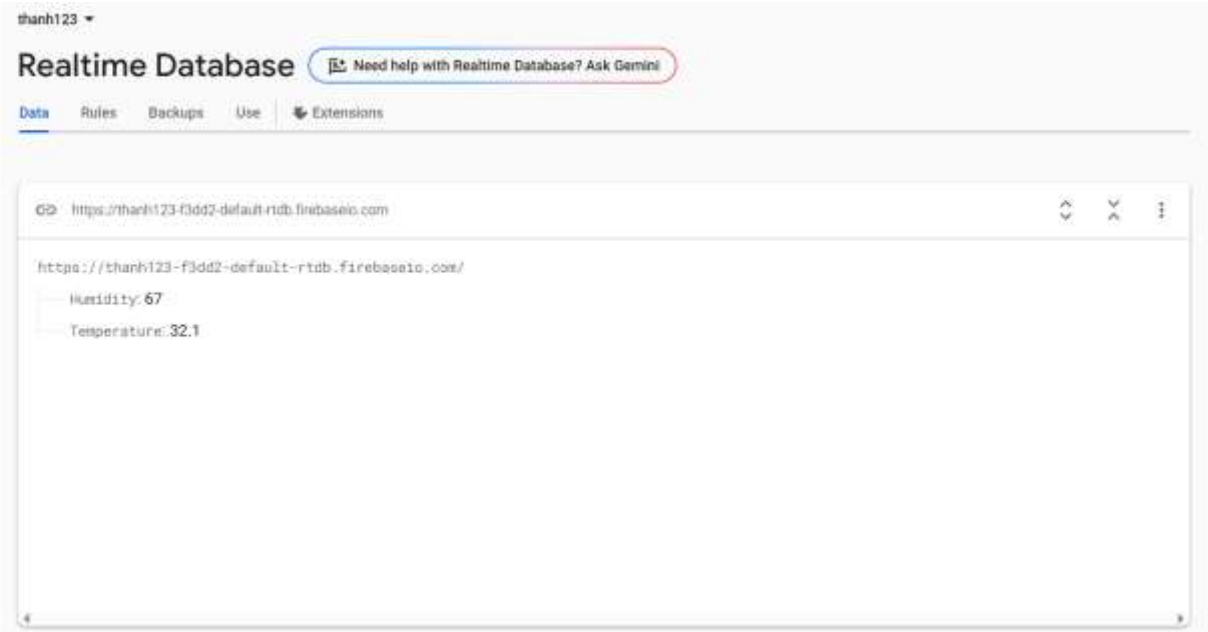

Figure 6–5 Realtime database

To transmit data from the microcontroller to the Web, follow these steps: first, read the data from the sensor modules, then send the data to Firebase. The data is sent continuously, and after Firebase reads the data from the sensors, it forwards the data to the website.

6.3 Interface

Here the group used Web programming languages such as Html, Css, Java to design a simple monitoring Web. By using some support tools, the group knew how to build the Web into main areas:

Title area: The top of the page introduces the project name "Design and Manufacturing Agricultural Robots" along with information about the students.

Data environment area (DataBoard): The middle part displays real-time parameters of air temperature and humidity, displayed in two boxes.

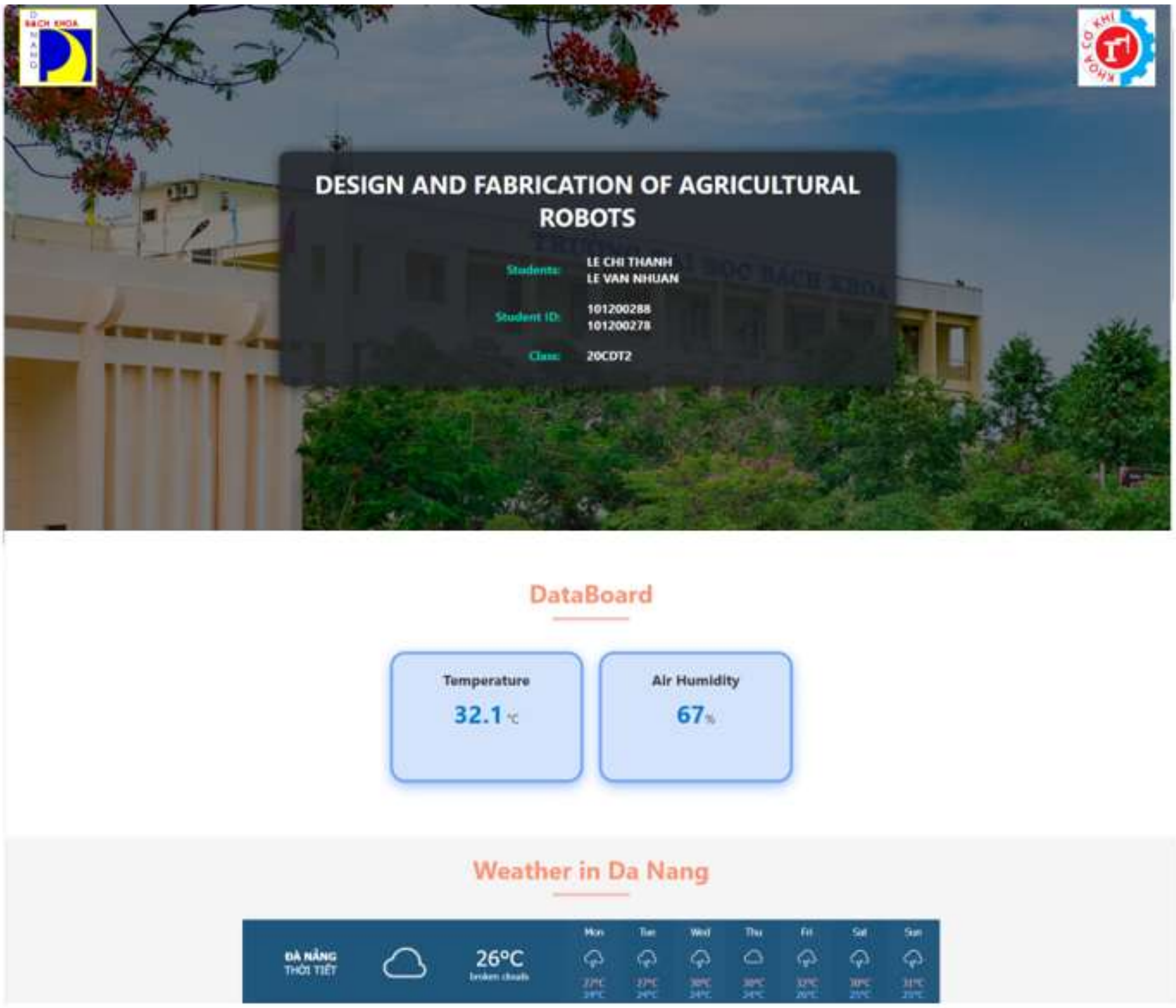

Figure 6–6 Interfeace

This is a simple website that the team has made. It can be developed later through versions such as monitoring, controlling, storing data, evaluating, giving ideas and being able to display some information about the Robot.

CHAPTER 7 CONCLUSION AND DEVELOPMENT DIRECTION

7.1 Realistic review of robot operation

This is an advanced laser weed killer robot that the team has created in agriculture, using laser technology and artificial intelligence (AI) to identify and destroy weeds without chemicals. It helps save time and effort for farmers, while protecting the environment. They work by scanning the field, identifying weeds with high accuracy, then using lasers to destroy them without harming crops. This is a sustainable solution, especially suitable for the trend of non-traditional farming. However, due to limited costs and short time, the team still has many things to try to adjust.

Figure 7–1 Realistic model

The team is looking to the future. The robot operates autonomously, moving through fields using advanced sensors and mapping technology to efficiently cover large areas. Its AI system is trained on extensive datasets of plant species, allowing it to recognize different types of weeds in different environmental conditions, whether it be changing light, soil type or weather patterns.

Below is the data table obtained when testing the Robot.

Table 7–1 Robot operation survey

No	Number of weed detectionlocations	Total exact location	Total Time
1	20	17	29.63 s
2	10	8	14.33 s
3	15	12	20.77 s
4	15	13	22.12 s
5	15	13	22.83 s
6	22	20	32.37 s
7	18	15	28.18 s
8	19	16	30.77 s
9	10	9	15.24 s
10	13	10	19.64 s

Data from above, the drawing group can represent the accuracy of the Robot.

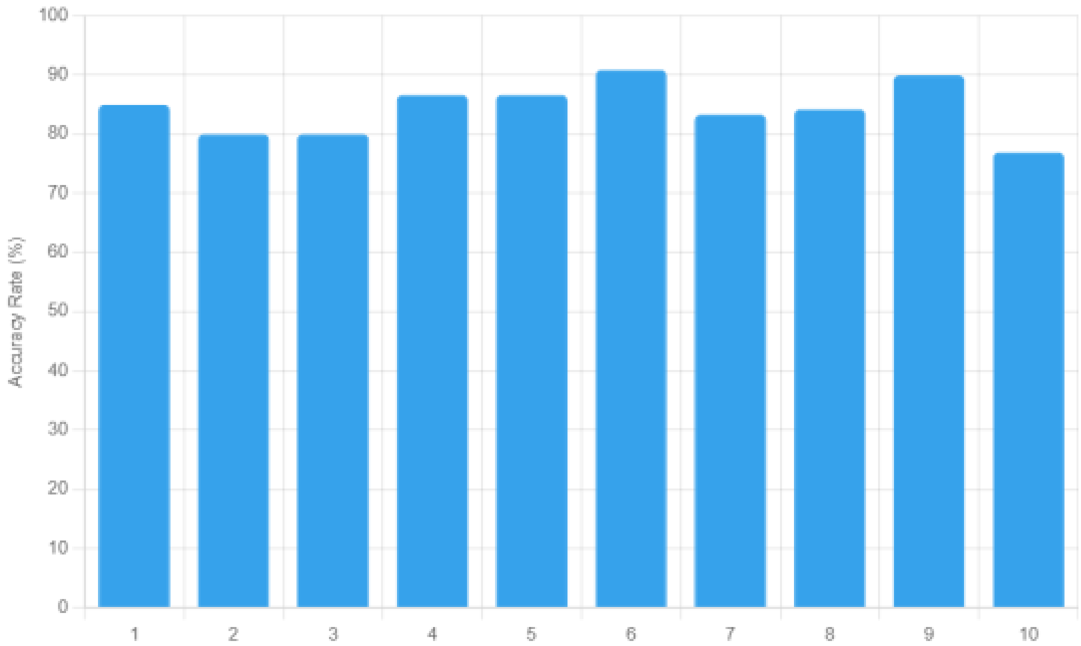

Figure 7–2 Chart for accuracy

Data from above, Draw the average time it takes for the Robot to remove 1 grass plant.

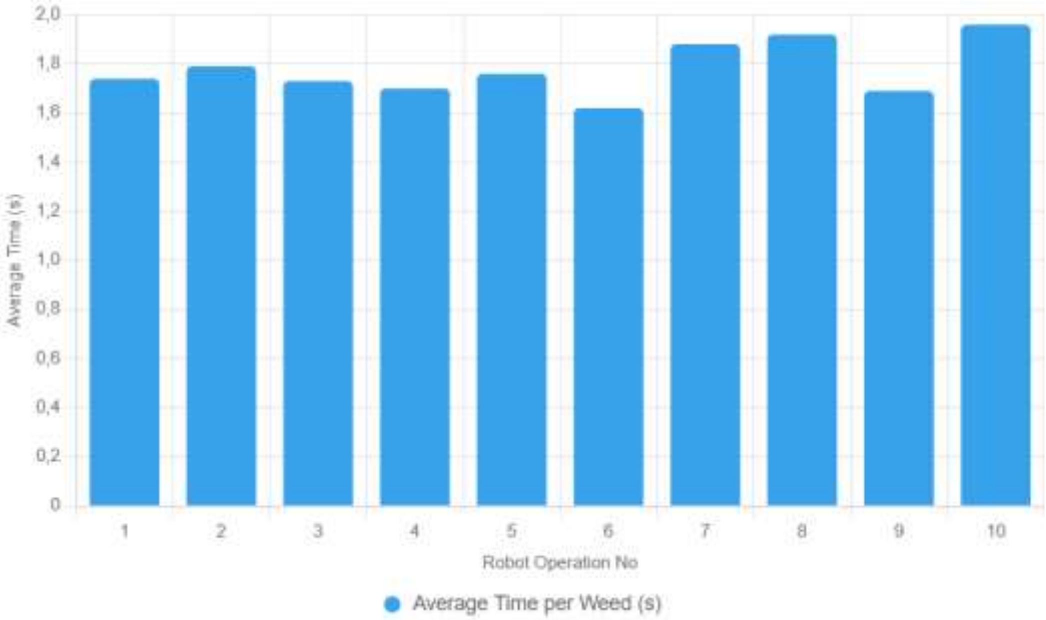

Figure 7–3 Time chart

Comments and reviews: From the trials and visual charts, we observe the following:

- Accuracy: The robot achieved an average accuracy of 83%, exceeding the target of 80%. This shows that the team performed image processing, weed detection and removal well, and was able to accurately locate weeds quite effectively.
- Stability: Accuracy ranged from 77% to 90%, with most values above 80%, demonstrating that the system had relatively high stability.
- Removal speed: The average speed was only 34 plants per minute, significantly lower than the target of 50 plants per minute. The robot took too long to process each type of weed, resulting in unsatisfactory performance.

The team discovered several causes:

In terms of accuracy, the robot faces challenges such as bright lighting conditions that hinder weed detection. Additionally, the weeds are too small in size, combined with the resolution of the webcam, preventing accurate recognition. The CNC mechanism also contributes to inaccuracies in its periodic error locations. Furthermore, the data used to train the AI is still limited, limited.

Regarding speed:

- The data processing and transmission time is still slow.
- The CNC mechanism can only carry one laser head, so it can only handle one weed at a time, resulting in time spent moving between positions.

- The laser beam is not hot enough to burn the weed quickly.

7.2 Improvement and future development direction

- Research and integrate multiple laser beams to eliminate several weeds simultaneously.
- Optimize the AI model, improve environmental conditions, add a review and verification mechanism.
- Upgrade hardware components such as the CNC system, webcam, and wheel drive mechanism.
- Additionally, incorporate new technologies like map scanning to enable autonomous navigation for the robot.
- Monitor and evaluate crop condition.

7.3 Conclusion

Through this project, the group has consolidated a lot of knowledge learned from the past school years, tried their best and succeeded in some problems as follows: Knowing how to process images to distinguish successfully, control multiple motors and devices at the same time, transmit and receive data between the computer and the microcontroller, between 2 microcontrollers. In addition, knowing how to apply Iot, using clean energy to move towards a better future

REFERENCES

[1].Nguyễn Trọng Hiệp. Nguyễn Văn Lẫm. Thiết kế chi tiết máy. Nhà xuất bản Giáo Dục, Hà Nội, 1999.

[2].Trịnh Chất. Lê Văn Uyển. Tính toán thiết kế hệ dẫn động cơ khí tập 1. Nhà xuất bản giáo dục (Tái bản lần thứ 6).

[3]."Shigley's Mechanical Engineering Design" (Richard G. Budynas & J. Keith Nisbett), McGraw-Hill Education, 10th Edition, 2015.

[4]."Machinery's Handbook" (Erik Oberg et al.), Industrial Press, 30th Edition, 2016.

[5]."Introduction to Robotics: Mechanics and Control" (John J. Craig), Pearson Education, 3rd Edition, 2005.

[6]."Friction and Lubrication in Mechanical Design" (A. Seireg), CRC Press, 1998.

[7].https://www.elcom.com.vn/iot-khien-nganh-nong-nghiep-tro-nen-thong-minh-nhu-the-nao-1699440994,19/3/2025.

[8].https://daithanhtech.com/tu-dong-hoa-trong-nong-nghiep/,10/03/2025.

[9].https://haophuong.com/news/robot-nong-nghiep/, 10/03/2025.

[10].https://tudonghoaatvn.com/dong-co-step/, 12/03/2025.

[11].https://quattranmy.com/dong-co-dc-la-gi/, 12/03/2025.

[12].https://congnghiepviet.com.vn/truyen-dong-dai-la-gi-cac-loai-va-uu-nhuoc-diem-cua-truyen-dong-dai.htm, 12/03/2025.

[13].https://truyendongcokhi.vn/bo-truyen-xich-trong-thiet-ke-may-nhong-sen-dia, 19/03/2025.

[14].https://www.mouser.vn/ProductDetail/Arduino, 19/03/2025.

[15].https://nshopvn.com/product/mach-dieu-khien-may-in-3d-ramps-1-4/, 19/03/2025.

[16].http://arduino.vn/tutorial/1570-gioi-thieu-module-esp32-va-huong-dan-cai-trinh-bien-dich-tren-arduino-ide, 20/03/2025.

[17].https://ictsaigon.com.vn/hikvision-ds-, 20/03/2025.

[18].https://www.iotzone.vn/esp32/esp32-co-ban/gioi-thieu-esp32-la-gi/, 20/03/2025.

[19].https://nshopvn.com/product/mach-dieu-khien-dong-co-buoc-a498E, 20/03/2025.

[20].https://dientuspider.com/shop/product/mach-dieu-khien-dong-co-bts796, 20/03/2025.

[21].https://linhkienvietnam.vn/bo-mach-module-relay-2-kenh-24v-10a. 22/03/2025

[22].https://linhkiendientuthuduc.com/dong-co-giam-toc12v-600w,25/03/2025

[23].https://vuletech.com/cong-tac-hanh-trinh-la-gi-vuletech/, 25/03/2025

[24].https://nshopvn.com/product/module-cam-bien-do-am-nhiet-do-dht11/, 30/3/2025.

[25].https://suntek.vn/tam-pin-nang-luong-mat-troi-1-pcc27.html, 30/03/2025.

[26].https://www.dhcsolar.com/doc-hieu-thong-so-tam-pin-nang-luong-mat-troi/, 30/3/2025.

[27].https://givasolar.com/tim-hieu-ve-dieu-khien-sac-nang-luong-mat-troi1n, 01/04/2025.

[28].https://shopee.vn/productE.01/04/2025.

[29] https://dientunguyenhien.vn/show/70, 04/04/2025

[30].https://shopee.vn/Module-laser-fibe5, 04/04/2025.

[31].https://aws.amazon.com/vi/what-is/iot/, 06/04/2025.

[32].https://www.google.com/search, 12/04/2025..

[33].https://vihoth.com/tin-tuc/tong-quan-phan-mem-solidworks-solidworks-la-gi, 20/4/2025.

[34].Engineering Toolbox (engineeringtoolbox.com), 20/3/2025.

[35].Tsubaki Chain Catalog (tsubakimoto.com), 20/3/2025.

[36].Renold Chain Catalog (renold.com), 21/3/2025.

[37].https://www.kaggle.com/datasets?search=Weed, 10/4/2025.

[38].https://mctt.vn/yolo-la-gi, 19/3/2025.

[39].https://aws.amazon.com/vi/what-is/iot/, 19/3/2025.

[40].https://fptshop.com.vn/tin-tuc/thu-thuat/firebase-la-gi-159218, 25/04/2025.

APPENDIX

Arduino	Python

#include <Arduino.h>
#include <AccelStepper.h>
// KHAI BÁO ĐỘNG CƠ
// Trục X: 2 động cơ chạy ngược chiều
int step_X1 = 60; int Dir_X1 = 61; int ena_X1 = 56; // Động cơ 1
int step_X2 = 46; int Dir_X2 = 48; int ena_X2 = 62; // Động cơ 2
// Trục Y
int step_Y = 54; int Dir_Y = 55; int ena_Y = 38;
AccelStepper Step_X1(1, step_X1, Dir_X1);
AccelStepper Step_X2(1, step_X2, Dir_X2);
AccelStepper Step_Y(1, step_Y, Dir_Y);
// KHAI BÁO CÔNG TẮC HÀNH TRÌNH
#define end_X1 14 // Công tắc hành trình cho động cơ X1
#define end_X2 18 // Công tắc hành trình cho động cơ X2
#define end_Y   3   // Công tắc hành trình cho trục Y
	import cv2
	import serial
	import time
	from ultralytics import YOLO
	import numpy as np
	import math
	import torch
	import threading
	import queue
	# Tải mô hình YOLOv8n đã huấn luyện
	print("Đang tải mô hình YOLO...")
	model = YOLO("e:/DATN/model/model50chuptay/best50.pt").to(device)
	print("Mô hình YOLO đã được tải thành công.")
	# Khởi động webcam
	cap = cv2.VideoCapture(0)
	cap.set(3, 640)
	cap.set(4, 480)
	cap.set(cv2.CAP_PROP_FPS, 30)
	if not cap.isOpened():
	print("Không thể mở webcam")
	exit()
	# Thông số hiệu chỉnh
	mm_per_pixel_x = 0.961

// KHAI BÁO PIN ĐIỀU KHIỂN LASER
 	
const int laserPin = 45;
 	
const int ledPin = 30;
 	
// KHAI BÁO PIN ĐIỀU KHIỂN ĐỘNG CƠ DC (BTS7960)
 	
// Động cơ 1
 	
const int RPWM1 = 4;
 	
const int LPWM1 = 5;
 	
const int REN1 = 43;
 	
const int LEN1 = 42;
 	
// Động cơ 2
 	
const int RPWM2 = 6;
 	
const int LPWM2 = 11;
 	
const int REN2 = 41;
 	
const int LEN2 = 40;
 	
// KHAI BÁO PIN NÚT NHẤN
 	
const int forwardButtonPin = 39; // Chân cho nút Tiến
 	
const int backwardButtonPin = 37; // Chân cho nút Lùi
 	
const int stopButtonPin = 36;   // Chân cho nút Stop
 	
//const int resetButtonPin = 5;   // Chân cho nút Reset
 	
int speedMotor = 30; // Tốc độ PWM (0-255) cho cả 2 động cơ
 	
// KHAI BÁO BIẾN LƯU TỌA ĐỘ
	mm_per_pixel_y = 1.058
	
	x_offset = 64.93
	
	y_offset = 36.32
	
	frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
	
	frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
	
	print(f"Độ phân giải: {frame_width}x{frame_height}")
	
	# Hàm pixel_to_mm
	
	def pixel_to_mm(x_pixel, y_pixel):
	
	y_pixel_adjusted = frame_height - y_pixel
	
	x_mm_base = (x_pixel - x_offset) * mm_per_pixel_x
	
	y_mm_base = (y_pixel_adjusted - y_offset) * mm_per_pixel_y
	
	distance = math.sqrt((x_pixel - x_offset)**2 + (y_pixel_adjusted - y_offset)**2)
	
	x_mm = x_mm_base * (1 + scale_x * distance)-2
	
	y_mm = y_mm_base * (1 + scale_y * distance)
	
	return round(x_mm, 1), round(y_mm, 1), DK
	
	# Hiển thị khung hình webcam với dự đoán YOLO, không đóng cửa sổ
	
	print("Hiển thị webcam với dự đoán YOLO trong 5 giây trước khi gửi lệnh HOME...")
	
	start_time = time.time()
	
	prev_time = time.time()
	
	home_sent = False

float x_mm = 0;
float y_mm = 0;
bool newPositionReceived = false;
// KHAI BÁO BIẾN TRẠNG THÁI
bool motorRunning = false; // Trạng thái động cơ (chạy hay dừng)
bool waitingForDirection = false; // Đánh dấu trạng thái chờ nhấn nút Tiến hoặc Lùi
int lastDirection = 0; // 0: chưa chọn, 1: tiến, 2: lùi
// KHAI BÁO HẰNG SỐ CHUYỂN ĐỔI
const float STEPS_PER_MM = 10; // 10 bước/mm
const int DIRECTION_FORWARD = 1;
const int DIRECTION_BACKWARD = 2;
// KHAI BÁO HÀM
void moveForward();
void moveBackward();
void stopMotors();
void home_X();
void home_Y();
void home();
	arduino_connected = False
	while not home_sent or not arduino_connected:
	ret, frame = cap.read()
	if not ret:
	print("Không thể đọc khung hình từ webcam")
	cap.release()
	cv2.destroyAllWindows()
	exit()
	# Tính FPS
	current_time = time.time()
	fps = 1 / (current_time - prev_time) if current_time != prev_time else 0
	prev_time = current_time
	cv2.putText(frame, f"FPS: {fps:.2f}", (10, 30),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	# Chạy mô hình YOLO để dự đoán
	results = list(model.track(frame, max_det=30, stream=True, device=device, persist=True, verbose=False))
	# Hiển thị thông báo trạng thái
	if not home_sent and time.time() - start_time < 3:
	cv2.putText(frame, "Tự động gửi lệnh HOME sau 5 giây", (10, 60),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	elif not home_sent:

void setup();
void loop();
// TRIỂN KHAI HÀM ĐIỀU KHIỂN ĐỘNG CƠ DC
void moveForward()
Serial.println("Moving forward...");
digitalWrite(LEN1, HIGH);
digitalWrite(REN1, HIGH);
digitalWrite(LEN2, HIGH);
digitalWrite(REN2, HIGH);
analogWrite(LPWM1, speedMotor); // Động cơ 1 tiến
analogWrite(RPWM1, 0);
analogWrite(LPWM2, 0); // Động cơ 2 tiến
analogWrite(RPWM2, speedMotor);
motorRunning = true;
lastDirection = DIRECTION_FORWARD; // Cập nhật hướng cuối cùng
digitalWrite(ledPin, HIGH); // Bật LED khi xe chạy
}
void moveBackward() {
Serial.println("Moving backward...");
digitalWrite(LEN1, HIGH);
digitalWrite(REN1, HIGH);
	cv2.putText(frame, "Đang gửi lệnh HOME...", (10, 60),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	elif not arduino_connected:
	cv2.putText(frame, "Đang chờ phản hồi từ Arduino...", (10, 60),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	# Sử dụng cùng tên cửa sổ với vòng lặp chính
	cv2.imshow("weed Detection", frame)
	# Gửi lệnh HOME sau 5 giây
	if not home_sent and time.time() - start_time >= 3: try:
	ser = serial.Serial('COM5', 9600, timeout=1)
	time.sleep(2)
	ser.write("HOME\n".encode())
	print("Sending command to move CNC to home position")
	time.sleep(3)
	ser.write("FWD\n".encode())
	print("Sending initial motor command: FWD")
	home_sent = True
	except serial.SerialException as e:
	print(f"Error connecting to Arduino: {e}")
	cap.release()
	cv2.destroyAllWindows()
	exit()
	# Chờ phản hồi từ Arduino

digitalWrite(LEN2, HIGH);
digitalWrite(REN2, HIGH);
analogWrite(LPWM1, 0); // Động cơ 1 lùi
analogWrite(RPWM1, speedMotor);
analogWrite(LPWM2, speedMotor); // Động cơ 2 lùi
analogWrite(RPWM2, 0);
motorRunning = true;
lastDirection = DIRECTION_BACKWARD; // Cập nhật hướng cuối cùng
digitalWrite(ledPin, HIGH); // Bật LED khi xe chạy
}
void stopMotors() {
Serial.println("Stopping motors...");
digitalWrite(LEN1, LOW);
digitalWrite(REN1, LOW);
digitalWrite(LEN2, LOW);
digitalWrite(REN2, LOW);
analogWrite(LPWM1, 0);
analogWrite(RPWM1, 0);
analogWrite(LPWM2, 0);
analogWrite(RPWM2, 0);
motorRunning = false;
	if home_sent and not arduino_connected:
	try:
	start_time_arduino = time.time()
	while time.time() - start_time_arduino < 10:
	if ser.in_waiting > 0:
	response = ser.readline().decode().strip
	print(f"Debug: Received from Arduino: {response}")
	if response == "FORWARD_PRESSED":
	print("Received FORWARD_PRESSED, vehicle starting forward...")
	arduino_connected = True
	break
	elif response == "BACKWARD_PRESSED":
	print("Received BACKWARD_PRESSED, vehicle starting backward...")
	arduino_connected = True
	break
	# Tiếp tục hiển thị webcam trong khi chờ Arduino
	ret, frame = cap.read()
	if not ret:
	print("Không thể đọc khung hình từ webcam")
	cap.release()
	cv2.destroyAllWindows()
	exit()
	# Tính FPS
	current_time = time.time()

waitingForDirection = true; // Reset trạng thái chờ hướng
//digitalWrite(ledPin, HIGH); // Tắt LED khi xe dừng
}
// TRIỂN KHAI HÀM HOMING
void home_X() {
Serial.println("Homing X...");
int homeX1 = 0;
int homeX2 = 0;
Step_X1.setMaxSpeed(3500);
Step_X1.setAcceleration(3500);
Step_X2.setMaxSpeed(3500);
Step_X2.setAcceleration(3500);
Step_X1.enableOutputs();
Step_X2.enableOutputs();
while (digitalRead(end_X1) == 0 && digitalRead(end_X2) == 0) {
Step_X1.moveTo(homeX1++);
Step_X2.moveTo(homeX2--);
Step_X1.run();
Step_X2.run();
}
while (digitalRead(end_X1) == 0) {
Step_X1.moveTo(homeX1++);
Step_X1.run();
	fps = 1 / (current_time - prev_time) if current_time != prev_time else 0
	prev_time = current_time
	cv2.putText(frame, f"FPS: {fps:.2f}", (10, 30),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	cv2.putText(frame, "Đang chờ phản hồi từ Arduino...", (10, 60),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	cv2.imshow("Weed Detection", frame)
	if cv2.waitKey(1) & 0xFF == ord('q'):
	print("Nhấn 'q'. Thoát chương trình.")
	cap.release()
	cv2.destroyAllWindows()
	ser.close()
	exit()
	time.sleep(0.1)
	else:
	print("Timeout: No button press detected from Arduino. Check connection or buttons.")
	cap.release()
	cv2.destroyAllWindows()
	ser.close()
	exit()
	except serial.SerialException as e:
	print(f"Error communicating with Arduino: {e}")
	cap.release()

}
Step_X1.setCurrentPosition(0);
while (digitalRead(end_X2) == 0) {
Step_X2.moveTo(homeX2--);
Step_X2.run();
}
Step_X2.setCurrentPosition(0);
}
void home_Y() {
Serial.println("Homing Y...");
int homeY = 0;
Step_Y.setMaxSpeed(3500);
Step_Y.setAcceleration(3500);
Step_Y.enableOutputs();
 	
while (digitalRead(end_Y) == 0) {
Step_Y.moveTo(homeY++);
Step_Y.run();
}
Step_Y.setCurrentPosition(0);
}
void home() {
home_X();
home_Y();
Serial.println("HOME_DONE"); // Gửi phản hồi xác nhận
	cv2.destroyAllWindows()
	ser.close()
	exit
	if cv2.waitKey(1) & 0xFF == ord('q'):
	print("Nhấn 'q'. Thoát chương trình.")
	cap.release()
	cv2.destroyAllWindows()
	if 'ser' in locals():
	ser.close()
	exit()
	if torch.cuda.is_available():
	torch.cuda.empty_cache()
	# Danh sách lưu tọa độ Weed
	weed_positions = queue.Queue()
	processed_positions = set()
	processing = False
	robot_stopped = False
	frame_count = 0
	pending_weeds = False
	waiting_for_stop = False
	pause_detection_until = 0
	# **Thêm biến để theo dõi thời gian sau CONTINUE**
	just_continued_until = 0
	def is_position_processed(x_mm, y_mm, tolerance=2):
	for processed_x, processed_y in processed_positions:

}
// TRIỂN KHAI HÀM SETUP
void setup() {
Serial.begin(9600);
// CẤU HÌNH ENDSTOP
pinMode(end_X1, INPUT_PULLUP);
pinMode(end_X2, INPUT_PULLUP);
pinMode(end_Y, INPUT_PULLUP);
// CẤU HÌNH NÚT NHẤN
pinMode(forwardButtonPin, INPUT_PULLUP); // Nút Tiến
pinMode(backwardButtonPin, INPUT_PULLUP); // Nút Lùi
pinMode(stopButtonPin, INPUT_PULLUP); // Nút Stop
// CẤU HÌNH LASER
pinMode(laserPin, OUTPUT);
digitalWrite(laserPin, LOW);
// CẤU HÌNH ĐÈN LED
pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, LOW); // Tắt đèn LED khi khởi động
// CẤU HÌNH ĐỘNG CƠ DC
pinMode(LEN1, OUTPUT);
pinMode(REN1, OUTPUT);
pinMode(LPWM1, OUTPUT);
	distance = math.sqrt((x_mm - processed_x)**2 + (y_mm - processed_y)**2)
	if distance < tolerance:
	return True
	return False
	def send_to_arduino():
	global processing, robot_stopped, pending_weeds, pause_detection_until, just_continued_until
	while True:
	if not weed_positions.empty() and not processing:
	processing = True
	position = weed_positions.get()
	x_mm, y_mm, x_center, y_center = position
	command = f"MOVE {x_mm},{y_mm}\n"
	print(f"Sending command: Position=({x_mm:.1f}, {y_mm:.1f})")
	ser.write(command.encode())
	start_time = time.time()
	while time.time() - start_time < 2:
	if ser.in_waiting > 0:
	response = ser.readline().decode().strip()
	print(f"Received response: {response}")
	if response == "DONE":
	print(f"Processed weed: Position=({x_mm:.1f}, {y_mm:.1f})")
	processed_positions.add((round(x_mm, 1), round(y_mm, 1)))

pinMode(RPWM1, OUTPUT);
pinMode(LEN2, OUTPUT);
pinMode(REN2, OUTPUT);
pinMode(LPWM2, OUTPUT);
pinMode(RPWM2, OUTPUT);
stopMotors();
// CU HÌNH ĐỘNG CƠ BƯỚC
Step_X1.setEnablePin(ena_X1);
Step_X1.setPinsInverted(false, false, true);
Step_X2.setEnablePin(ena_X2);
Step_X2.setPinsInverted(false, false, true);
Step_Y.setEnablePin(ena_Y);
Step_Y.setPinsInverted(false, false, true);
Step_X1.setMaxSpeed(3500);
Step_X1.setAcceleration(3500);
Step_X2.setMaxSpeed(3500);
Step_X2.setAcceleration(3500);
Step_Y.setMaxSpeed(3500);
Step_Y.setAcceleration(3500);
waitingForDirection = true; // Sẵn sàng chờ hướng sau khởi động
// CHẠY HOMING
//home();
// Đợi lệnh FWD từ Python
}
	processing = False
	break
	else:
	print("Timeout: Bỏ qua phản hồi từ Arduino")
	processing = False
	elif weed_positions.empty() and robot_stopped and not processing and not pending_weeds:
	try:
	ser.write("CONTINUE\n".encode())
	print("Sent CONTINUE command to Arduino")
	robot_stopped = False
	pause_detection_until = time.time() + 5
	# **Đặt thời gian để dừng ngay sau CONTINUE**
	just_continued_until = time.time() + 4 # 2s pause + 2s sau đó
	print("Pausing YOLO detection for 2 seconds after CONTINUE")
	except serial.SerialException as e:
	print(f"Error writing to Serial: {e}")
	time.sleep(0.1)
	
	arduino_thread = threading.Thread(target=send_to_arduino, daemon=True)
	stop_thread = threading.Thread(target=handle_stop_after_delay, daemon=True)
	arduino_thread.start()

// TRIỂN KHAI HÀM LOOP
void loop() {
// KIỂM TRA NÚT NHẤN
// Nút Tiến
if (digitalRead(forwardButtonPin) == LOW) { // Nút được nhấn (LOW do pull-up)
delay(50); // Debounce
if (digitalRead(forwardButtonPin) == LOW) {
if (waitingForDirection && !motorRunning) {
Serial.println("FORWARD_PRESSED");
moveForward();
waitingForDirection = false;
}
while (digitalRead(forwardButtonPin) == LOW); // Chờ nút được nhả
}
}
// Nút Lùi
if (digitalRead(backwardButtonPin) == LOW) { // Nút được nhấn (LOW do pull-up)
delay(50); // Debounce
	stop_thread.start()
	
	prev_time = time.time()
	weeds_detected = False
	last_weed_frame = 0
	# **Đổi tên biến weed_detected_time thành crop_detected_time**
	crop_detected_time = None
	counting_frames = False
	
	while True:
	ret, frame = cap.read()
	if not ret:
	print("Không thể đọc khung hình")
	break
	if counting_frames:
	frame_count += 1
	current_time = time.time()
	fps = 1 / (current_time - prev_time) if current_time != prev_time else 0
	prev_time = current_time
	
	cv2.putText(frame, f"FPS: {fps:.2f}", (10, 30),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
	cv2.putText(frame, f"Frame: {frame_count}", (10, 50),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)

if
(digitalRead(backwardButtonPin
) == LOW) {
if (waitingForDirection &&
!motorRunning) {
Serial.println("BACKWARD_P
RESSED");
moveBackward();
waitingForDirection = false;
}
while
(digitalRead(backwardButtonPin
) == LOW); // Chờ nút được nhả
}
}
 	
//Nút Stop
if (digitalRead(stopButtonPin)
== LOW) { // Nút được nhấn
(LOW do pull-up)
delay(50); // Debounce
if (digitalRead(stopButtonPin)
== LOW) {
if (motorRunning) {
Serial.println("STOP_PRESSED
");
stopMotors();
newPositionReceived = false; //
Reset vị trí khi dừng
}
	# Kiểm tra nếu đang tạm ngừng nhận diện
	YOLO
	if time.time() < pause_detection_until:
	cv2.putText(frame, "Tạm ngừng nhận diện
	YOLO...", (10, 80),
	cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255,
	255), 2)
	cv2.imshow("Weed Detection", frame)
	if cv2.waitKey(1) & 0xFF == ord('q'):
	break
	if torch.cuda.is_available():
	torch.cuda.empty_cache()
	continue
	
	results = list(model.track(frame, max_det=20,
	stream=True, device=device, persist=True,
	verbose=False))
	
	for r in results:
	boxes = r.boxes
	track_ids = r.boxes.id
	for box, track_id in zip(boxes,
	track_ids.int().tolist() if track_ids is not None
	else [None] * len(boxes)):
	cls = int(box.cls[0])
	conf = box.conf[0]
	label = model.names[cls]

while (digitalRead(stopButtonPin) == LOW); // Chờ nút được nhả
}
}
 	
// NHẬN LỆNH TỪ SERIAL
if (Serial.available() > 0) {
String command = Serial.readStringUntil('\n');
command.trim();
Serial.print("Received command: ");
Serial.println(command);
 	
if (command == "STOP") {
stopMotors();
newPositionReceived = false; // Reset vị trí khi dừng
}
else if (command == "CONTINUE" || command == "FWD") {
if (command == "FWD") {
waitingForDirection = true; // Chờ nhấn nút Tiến hoặc Lùi
Serial.println("Waiting for direction (Forward or Backward)...");
}
	if label == "crop" and conf < 0.3:
	continue
	if label == "weed" and conf < 0.2:
	continue
	x1, y1, x2, y2 = box.xyxy[0]
	x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
	x_center = (x1 + x2) // 2
	y_center = (y1 + y2) // 2
	x_mm, y_mm, DK = pixel_to_mm(x_center, y_center)
	cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0) if label == "crop" else (0, 0, 255), 1)
	label_text = f"{label} {conf:.2f}"
	if track_id is not None:
	label_text += f" ID:{track_id}"
	cv2.putText(frame, label_text, (x1, y1 - 10),
	cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0, 255, 0) if label == "crop" else (0, 0, 255), 1)
	if label == "weed":
	cv2.putText(frame, f"({x_mm:.1f}, {y_mm:.1f}) mm", (x1, y1 + 20),
	cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0, 0, 255), 1)
	
	# **Sửa logic kiểm tra Crop để dừng ngay sau CONTINUE**
	has_crop = any(
	model.names[int(box.cls[0])] == "crop" and box.conf[0] >= 0.3

else if (command == "CONTINUE" && !motorRunning) { // Sau khi xử lý cỏ
if (lastDirection == DIRECTION_FORWARD) {
home();
moveForward(); // Tiếp tục chạy tiến
} else if (lastDirection == DIRECTION_BACKWARD) {
home();
moveBackward(); // Tiếp tục chạy lùi
} else {
home();
moveForward(); // Mặc định chạy tiến nếu chưa có hướng
}
}
}
else if (command == "HOME") {
home();
}
else if (command.startsWith("MOVE")) {
// Xử lý lệnh "MOVE x_mm,y_mm"
	for r in results
	for box in r.boxes
	)
	if not robot_stopped and has_crop and not waiting_for_stop:
	if time.time() < just_continued_until:
	try:
	ser.write("STOP\n".encode())
	print("Sent STOP command to Arduino (new crops detected immediately after CONTINUE)")
	robot_stopped = True
	frame_count = 0
	counting_frames = True
	weeds_detected = False
	last_weed_frame = 0
	crop_detected_time = None
	pending_weeds = True
	waiting_for_stop = False
	except serial.SerialException as e:
	print(f"Error writing to Serial: {e}")
	else:
	if crop_detected_time is None:
	crop_detected_time = time.time()
	print("Detected Crop, waiting 2 seconds before stopping...")
	waiting_for_stop = True
	
	# Xử lý khi xe dừng: Lưu tọa độ Weed

int spaceIndex = command.indexOf(' ');
if (spaceIndex != -1) {
String coords = command.substring(spaceIndex + 1);
int commaIndex = coords.indexOf(',');
if (commaIndex != -1) {
x_mm = coords.substring(0, commaIndex).toFloat();
y_mm = coords.substring(commaIndex + 1).toFloat();
newPositionReceived = true;
Serial.print("Parsed: x_mm = ");
Serial.print(x_mm);
Serial.print(", y_mm = ");
Serial.println(y_mm);
}
}
}
}
// DI CHUYỂN ĐẾN VỊ TRÍ CỎ VÀ KÍCH HOẠT LASER
if (newPositionReceived) {
long x_steps = x_mm * STEPS_PER_MM;
long y_steps = -y_mm * STEPS_PER_MM;
	current_positions = []
	if not weeds_detected and robot_stopped:
	for r in results:
	boxes = r.boxes
	track_ids = r.boxes.id
	for box, track_id in zip(boxes, track_ids.int().tolist() if track_ids is not None else [None] * len(boxes)):
	cls = int(box.cls[0])
	conf = box.conf[0]
	label = model.names[cls]
	if label == "weed" and conf < 0.2:
	continue
	x1, y1, x2, y2 = box.xyxy[0]
	x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
	x_center = (x1 + x2) // 2
	y_center = (y1 + y2) //
	x_mm, y_mm, DK = pixel_to_mm(x_center, y_center)
	if label == "weed" and 10 <= x_mm <= 400 and 10 <= y_mm <= 400:
	if not is_position_processed(x_mm, y_mm, tolerance=5):
	if frame_count >= 25:
	print(f"Frame {frame_count}: Adding weed - Position=({x_mm:.1f}, {y_mm:.1f}), ID={track_id}")
	current_positions.append((x_mm, y_mm, x_center, y_center))

// Kiểm tra giới hạn tọa độ (tùy chỉnh theo hệ thống)
if (x_mm < 0 || x_mm > 400 || y_mm < 0 || y_mm > 400) {
Serial.println("ERROR: Coordinates out of range");
newPositionReceived = false;
return;
}
Step_X1.moveTo(-x_steps);
Step_X2.moveTo(x_steps);
Step_Y.moveTo(y_steps);
 	
Serial.println("Moving to position...");
while (Step_X1.distanceToGo() != 0 || Step_X2.distanceToGo() != 0 || Step_Y.distanceToGo() != 0) {
Step_X1.run();
Step_X2.run();
Step_Y.run();
}
digitalWrite(laserPin, HIGH); // Kích hoạt laser tự động
delay(500); // Giữ laser trong 500ms
digitalWrite(laserPin, LOW); // Tắt laser
	if frame_count >= 20 and current_positions and not weeds_detected:
	current_positions.sort(key=lambda pos: math.sqrt(pos[0]**2 + pos[1]**2))
	for pos in current_positions:
	weed_positions.put(pos)
	print(f"Frame {frame_count}: Detected {len(current_positions)} weeds. Sorted positions: {current_positions}")
	weeds_detected = True
	last_weed_frame = frame_count
	
	pending_weeds = False
	cv2.imshow("Weed Detection", frame)
	if cv2.waitKey(1) & 0xFF == ord('q'):
	break
	if torch.cuda.is_available():
	torch.cuda.empty_cache()
	cap.release()
	cv2.destroyAllWindows()
	if 'ser' in locals():
	ser.close()

Serial.println("DONE"); // Gửi phản hồi
newPositionReceived = false;
}
Step_X1.run();
Step_X2.run();
Step_Y.run();
}
// ĐIỂM VÀO CHƯƠNG TRÌNH
int main() {
init(); // Khởi tạo phần cứng Arduino
setup(); // Gọi hàm setup
while (1) { // Vòng lặp vô hạn
loop(); // Gọi hàm loop
}
return 0;	

```
Code Esp32
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <ArduinoJson.h>
// Wi-Fi Configuration
#define WIFI_SSID "iPhone"        // Replace with your Wi-Fi SSID
#define WIFI_PASSWORD "11111121" // Replace with your Wi-Fi password
// Firebase Configuration
```

```
#define FIREBASE_HOST "thanh123-f3dd2-default-rtdb.firebaseio.com" // Your
Firebase Database URL
#define FIREBASE_AUTH
"RPhfYj5lh01PQNNO5ASZKoCmw4GoartYL1jaZE2P"                // Replace with
your Firebase Secret Key
// Serial2 Configuration for ESP32
#define RXD2 16 // GPIO16 as RX
#define TXD2 17 // GPIO17 as TX (not used in this case)
// Define Firebase objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
void setup() {
// Start Serial Monitor for debugging
Serial.begin(115200);
// Start Serial2 to communicate with Arduino
Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
// Connect to Wi-Fi
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED) {
Serial.print(".");
delay(500);
}
Serial.println();
Serial.print("Connected, IP address: ");
Serial.println(WiFi.localIP());
// Configure Firebase
```

```
config.host = FIREBASE_HOST;
config.signer.tokens.legacy_token = FIREBASE_AUTH;
// Initialize Firebase
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true); // Automatically reconnect to Wi-Fi if disconnected
        }
void loop() {
if (Serial2.available()) {
String data = Serial2.readStringUntil('\n'); // Read string until newline
if (data == "ERROR") {
Serial.println("Error: Failed to receive data from DHT11!");
} else {
// Parse temperature and humidity data
int commaIndex = data.indexOf(',');
float temperature = data.substring(0, commaIndex).toFloat();
float humidity = data.substring(commaIndex + 1).toFloat();
// Display data on Serial Monitor
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.print(" *C, Humidity: ");
Serial.print(humidity);
Serial.println(" %");
// Send data to Firebase
if (Firebase.setFloat(fbdo, "/Temperature", temperature)) {
Serial.println("Temperature sent successfully");
} else {
Serial.print("Failed to send Temperature: ");
```

```
Serial.println(fbdo.errorReason());
}
if (Firebase.setFloat(fbdo, "/Humidity", humidity)) {
Serial.println("Humidity sent successfully");
} else {
Serial.print("Failed to send Humidity: ");
Serial.println(fbdo.errorReason());
}
}
}
delay(100); // Avoid reading too fast
}
```

Code Interface

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Web IoT - Agricultural Robot</title>
    <style>
      /* Giữ nguyên CSS của bạn */
      * {
        padding: 0;
```

```
          ),
          url("Resources/img/bkdn2-16900982870901717218480_6629d570a957e.jpg");
        background-size: cover;
        background-position: center;
        background-attachment: fixed;
        height: 100vh;
        color: white;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
```

margin: 0;
box-sizing: border-box;
}
html {
font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
font-size: 16px;
background-color: #f5f5f5;
color: #333;
}
 	
body {
margin: 0;
}
header {
background-image: linear-gradient(
rgba(0, 0, 0, 0.5),
rgba(0, 0, 0, 0.5)
color: #ffffff;
}
td {
padding: 8px 16px;
}
.label {
text-align: right;
color: #00d1b2;
}
	text-align: center;
	}
	
	.header-content {
	max-width: 800px;
	padding: 20px;
	background-color: rgba(33, 37, 41, 0.85);
	border-radius: 12px;
	box-shadow: 0 0 15px rgba(0, 0, 0, 0.6);
	}
	h1 {
	font-size: 2rem;
	margin-bottom: 1rem;
	}
	table {
	margin: auto;
	font-size: 1rem;
	font-weight: bold;
	}
	h2 {
	font-size: 2rem;
	color: #fa9576;
	text-align: center;
	margin-bottom: 30px;
	position: relative;
	}

```
.value {
  text-align: left;
}

.logo {
  position: absolute;
  top: 20px;
  width: 100px;
}
.logo.left {
  left: 20px;
}

.logo.right {
  right: 20px;
}
.DataBoard {
  padding: 60px 20px;
  background-color: #fff;
  margin: 10px;
  text-align: center;
  padding: 20px;
}
.analys-box h3 {
  margin-bottom: 10px;
}
.sensor-value {
```

```
h2::after {
  content: "";
  display: block;
  width: 100px;
  height: 4px;
  background-color: rgba(250,
149, 118, 0.6);
  margin: 10px auto 0;
}
.row {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
}
.analys-box {
  height: 170px;
  width: 250px;
  background-color: #d3e3fc;
  border: 4px solid #77a6f7;
  border-radius: 20px;
  box-shadow: 0px 5px 15px
rgba(119, 166, 247, 0.7);
    </tr>
    <tr>
      <td
class="label">Class:</td>
      <td
class="value">20CDT2</td>
```

```
        font-size: 2rem;
        color: #007acc;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <header>
      <img class="logo left"
src="./Resources/img/logoBk.jpg"
alt="logo" />
      <div class="header-content">
        <h1>DESIGN AND
FABRICATION OF AGRICULTURAL
ROBOTS</h1>
        <table>
          <tr>
            <td
class="label">Students:</td>
            <td class="value">LE
CHI THANH<br />LE VAN
NHUAN</td>
          </tr>
          <tr>
            <td class="label">Student
ID:</td>
            <td
class="value">101200288<br
/>101200278</td>
            <div class="header">
          </tr>
        </table>
      </div>
      <img class="logo right"
src="./Resources/img/logoCokhi.png"
alt="logo" />
    </header>
    <section id="DataBoard"
class="DataBoard">
      <h2>DataBoard</h2>
      <div class="row">
        <div class="analys-box">
          <h3>Temperature</h3>
          <p><span class="sensor-
value" id="temperature"></span>
°C</p>
        </div>
        <div class="analys-box">
          <h3>Air Humidity</h3>
          <p><span class="sensor-
value" id="Humi"></span>%</p>
        </div>
      </div>
    </section>

    <section>
      <!-- WEATHER WIDGET --
>
```

```
                <h2>Weather in Da
Nang</h2>
              </div>
              <div class="body"
style="text-align: center; width: 900px">
                <a
                  class="weatherwidget-
io"

href="https://forecast7.com/en/16d05108
d20/da-nang/"
                  data-label_1="ĐÀ
NẴNG"
                  data-label_2="THỜI
TIẾT"
                  data-theme="original"
                >
                  Weather in Da Nang
                </a>
              </div>
            </div>
          </div>
        </div>
      </section>
      <script type="module"
src="/js.js"></script>
      <!-- Script để load widget thời
tiết -->
      <script>
        !(function (d, s, id) {
```

```
          <div class="row clearfix"
style="margin-top: 20px">
            <div class="col-xs-12 col-
sm-12 col-md-4 col-lg-12">
              <div class="card">

fjs.parentNode.insertBefore(js, fjs);
          }
        })(document, "script",
"weatherwidget-io-js");
      </script>
    </body>
  </html>
  code js.
  import { initializeApp } from
"https://www.gstatic.com/firebasejs/11.6.
1/firebase-app.js";
  import {
    getDatabase,
    ref,
    onValue,
  } from
"https://www.gstatic.com/firebasejs/11.6.
1/firebase-database.js";

  const firebaseConfig = {
    apiKey:
"AIzaSyA8AvcbPHetrcQ6jZmNysU2m
XJIw1RyLLs",
```

```
            var js,
             fjs =
d.getElementsByTagName(s)[0];
            if (!d.getElementById(id)) {
             js = d.createElement(s);
             js.id = id;
             js.src =
"https://weatherwidget.io/js/widget.min.j
s";
        const app =
initializeApp(firebaseConfig);
        // Initialize Realtime Database
        const database =
getDatabase(app);
        // Auto load Temperature
        const tempRef = ref(database,
"/Temperature");
        onValue(
          tempRef,
          (snapshot) => {
            const nd = snapshot.val();
            if (nd !== null && nd !==
undefined && nd >= -50 && nd <= 100)
{

document.getElementById("temperature
").innerHTML = nd;
              console.log("Temperature:",
nd);
```

```
        authDomain: "thanh123-
f3dd2.firebaseapp.com",
        databaseURL: "https://thanh123-
f3dd2-default-rtdb.firebaseio.com",
        projectId: "thanh123-f3dd2",
        storageBucket: "thanh123-
f3dd2.firebasestorage.app",
        messagingSenderId:
"415383355790",
        appId:
"1:415383355790:web:65a3689478da34
052c24df",
        measurementId: "G-
YJYWW5YC4F",
      };
      // Initialize Firebase
        }
      );


      // Auto load Air Humidity
      const humiRef = ref(database,
"/Humidity");
      onValue(
        humiRef,
        (snapshot) => {
          const nd = snapshot.val();
          if (nd !== null && nd !==
undefined && nd >= 0 && nd <= 100) {
```

} else {
document.getElementById("temperature").innerHTML = "N/A";
console.log("Temperature: Invalid or no data");
}
},
(error) => {
console.error("Error fetching temperature:", error);
document.getElementById("temperature").innerHTML = "Error";
	document.getElementById("Humi").innerHTML = nd;
	console.log("Humidity:", nd);
	} else {
	document.getElementById("Humi").innerHTML = "N/A";
	console.log("Humidity: Invalid or no data");
	}
	},
	(error) => {
	console.error("Error fetching humidity:", error);
	document.getElementById("Humi").innerHTML = "Error";