Applying Proteus Software to Design and Manufacture Electronic Circuits of Rice-Picking Robots to Meet the Requirements of the Robocon 2024 Innovation Contest in Vietnam

Do Huy Tung, Duong Thi Hoa

***Abstract*— In the context of the rapid development of automation and robotics technology, the 2024 Asia-Pacific Robot Creativity Competition, held in Vietnam, has become an exciting platform for students and young engineers to showcase their creativity and technical skills. One of the biggest challenges in the competition is the design and construction of a robot capable of performing a specific task, which is rice harvesting. This task requires not only high precision but also the ability to operate effectively in a diverse and complex competition environment. This paper presents the results of applying Proteus software in designing and constructing control circuits for the actuators of the rice-harvesting robot. The results show that the control system is precise, and the robot's ability to function in a complex and flexible spatial environment meets the competition's requirements, opening up opportunities for the development of autonomous robots that can improve harvesting efficiency in the agriculture sector.**

***Index Terms*— Simulation, robot, competition, rice harvesting, robocon**

I. INTRODUCTION

The rice-harvesting robot in the 2024 Robocon Creativity Competition in Vietnam will be an important part of the challenge that participating teams must face. Creating a robot that can automatically detect and harvest rice in a complex spatial environment is a significant challenge. This requires teams to develop technologies such as object recognition sensors, precise control systems, and stable operation in real-world conditions. The design and construction ideas for the rice-harvesting robot not only help teams form strategies to win the competition but also represent a major challenge with large-scale implications for robot applications in the agriculture sector. If the robot can meet the tasks set in the competition, it will open up opportunities for the development of autonomous robots that can improve crop harvesting efficiency, reduce human labor, and enhance sustainability in agricultural production. To achieve this, the robot must be capable of recognizing and analyzing environmental data (such as the shape and color of the rice), while also handling potential situations during the harvesting process, such as fallen rice or obstacles. The use of Proteus software for designing and simulating the rice-harvesting robot for the 2024 Robocon Creativity Competition in Vietnam offers many benefits, especially during the design and simulation phase. It helps teams save time, optimize designs, and test circuits and control systems before actual implementation. This not only improves manufacturing efficiency but also ensures that the robot can operate stably in real-world environments, contributing to the advancement of creativity and the application of advanced technology in agriculture

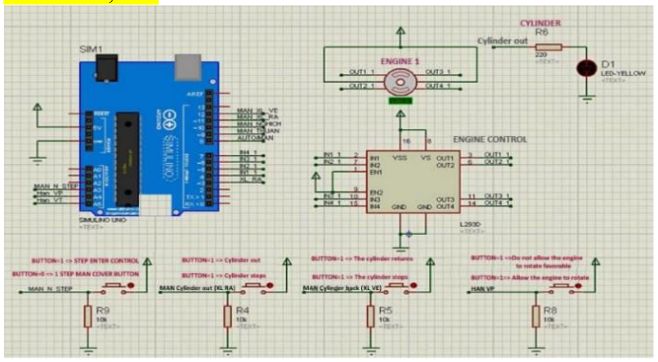
Manuscript received January 6, 2025

Do Huy Tung, Sao Do University - No. 76, Nguyen Thi Due Street, Thai Hoc II Street, Sao Do Ward, Chi Linh City, Hai Duong Province
Duong Thi Hoa, Sao Do University - No. 76, Nguyen Thi Due Street, Thai Hoc II Street, Sao Do Ward, Chi Linh City, Hai Duong Province

II. CONTROL CIRCUIT SIMULATION

A. Simulation process on Proteus

Click on P on the toolbar P L DEVICES - the "Pick Devices" window will appear. In the keywords field, type "Arduino" to select the design tool, then type the keywords of other components. For example: led matrix, button, buzzer, 74HC595, etc.


SIM1
SIMULINO UNO
AREF
MAN N STEP
Man VP
Man VT
AUTOMAN
ENGINE 1
OUT1 1
OUT2 1
OUT3 1
OUT4 1
CYLINDER
R6
Cylinder out
220
D1
LED-YELLOW
ENGINE CONTROL
IN1
IN2
EN1
EN2
IN3
IN4
VSS
VS
GND
OUT1
OUT2
OUT3
OUT4
L293D
BUTTON=1 => STEP ENTER CONTROL
BUTTON=0 => 1 STEP MAN COVER BUTTON
BUTTON=1 => Cylinder out
BUTTON=1 => Cylinder stops
BUTTON=1 => The cylinder returns
BUTTON=1 => The cylinder stops
BUTTON=1 =>Do not allow the engine to rotate favorable
BUTTON=1=> Allow the engine to rotate
MAN N STEP
MAN Cylinder out (XL RA)
MAN Cylinder back (XL VE)
MAN VP
R9
R4
R5
R8
10k
Figure 1: Simulation on Proteus

Extract the hex file from Arduino IDE

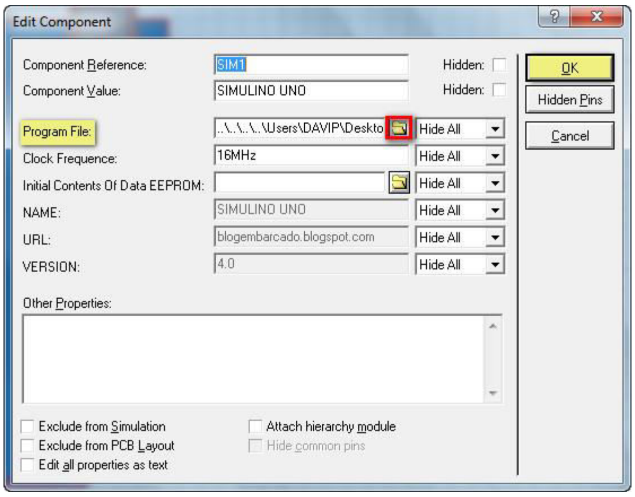
The hex file is short for Hexadecimal, a number system created from the code and the compiler. Once the hex file is extracted from Arduino IDE, it will be loaded into Proteus 8. This process is similar to connecting physical wires to the Arduino and running the code on the computer. After clicking the 'Verify' button, the Arduino IDE will generate the hex file and then upload it to the circuit.

Step 1: Open the file 'preferences.txt' at: C:\ProgramFiles\Arduino\lib\preferences.txt.

Be sure to select the board that matches the one to be simulated.

Step 2: Add a line at the end of the preferences file pointing to the location where the hex file should be saved. You can create a "hex" folder on the Desktop for storage.

Step 3: Exit the Arduino IDE, reopen it, and upload the code. The hex file will appear and be saved in the specified folder.



Edit Component

Component Reference:	SIM1	Hidden:	
Component Value:	SIMULINO UNO	Hidden:	
Program File:	..\..\..\..\Users\DAVIP\Deskto	Hide All	
Clock Frequence:	16MHz	Hide All	
Initial Contents Of Data EEPROM:		Hide All	
NAME:	SIMULINO UNO	Hide All	
URL:	blogembarcado.blogspot.com	Hide All	
VERSION:	4.0	Hide All	

Other Properties:

Figure 2: Uploading the Hex file into Proteus

Open the previously designed circuit file, double-click on the Arduino board. The Edit Component dialog will appear. In the Program File section, locate the folder containing the saved .hex file and click OK.

B. Setup the mode for the PID circuit

Use a USB-to-serial module to connect the computer with the driver, ensuring correct pin connections: 0v -> 0v, 5v -> 5v, Tx -> Rx, Rx -> Tx. After the connection, the LED indicator will show the operational status.

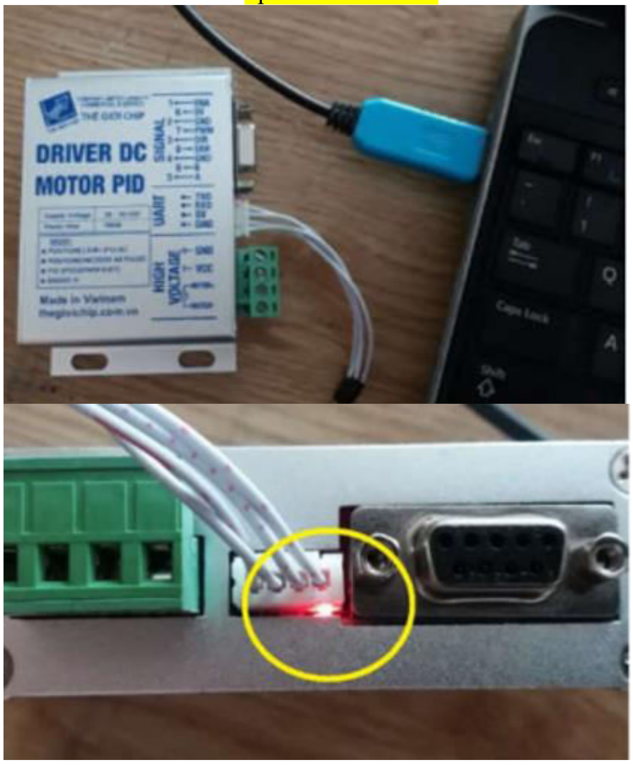


Figure 3: Connecting PID with the software

The baudrate section is where you select the connection baud rate with the driver; the default setting is 19200.

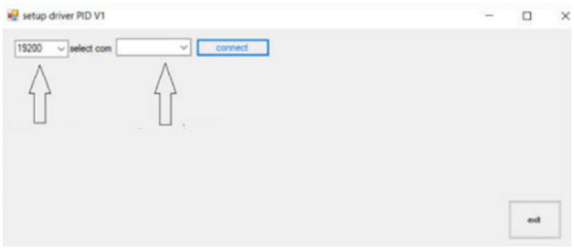
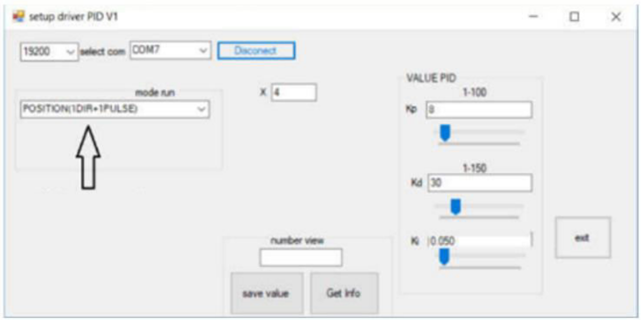


Figure 4: Select the input port for PID

The "select com" section is where you choose the COM port to connect with the driver.

Click "connect" to establish the connection. The interface will display after a successful connection.

Note: When selecting or entering any parameter on the driver, observe the "number view" box, which will show the parameter you just selected. This indicates that the parameter you entered has been sent to the driver.



setup driver PID V1

19200 select com COM7 Disconect

mode run: POSITION(1DIR+1PULSE) X: 4

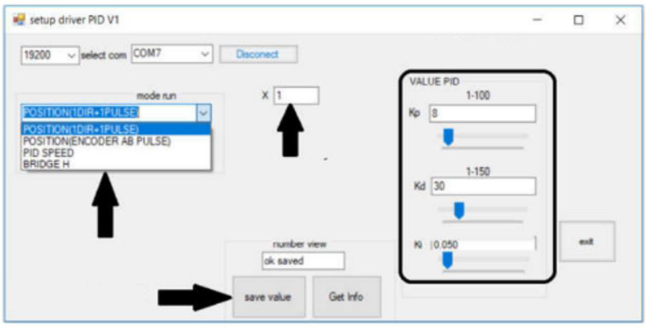
VALUE PID: Kp (1-100): 8; Kd (1-150): 30; Ki: 0.050

Figure 5: Select the mode for PID

C. Position PID mode

Connect the computer to the driver, open the setup software, and select POSITION (1DIR+1PULSE) in the run mode. In the PID VALUE section, enter the values for Kp, Kd, Ki. Click 'Save Value' to save the settings. The 'Number View' will display 'Ok'.

Click 'Exit' to close the program. In this POSITION mode, the driver is controlled through three pins.



setup driver PID V1

19200 select com COM7 Disconect

mode run: POSITION(1DIR+1PULSE) X: 1

POSITION(1DIR+1PULSE)
POSITION(ENCODER AB PULSE)
PID SPEED
BRIDGE H

VALUE PID: Kp (1-100): 8; Kd (1-150): 30; Ki: 0.050

number view: ok saved

Figure 6: Setup PID parameters

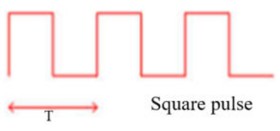
- Pin (1) Enable: When a logic low signal (0) is applied, it allows the driver to operate. When a logic high signal (1) is applied, it disables the driver from operating.
- Pin (3) Direction: Controls the motor to rotate forward when the signal is at level 0, and rotate in reverse when the signal is at level 1.
- Pin (7) Pulse: This pin sends pulses to the motor to operate. When one pulse is sent, the motor will rotate by one encoder pulse. Use the following formula to calculate the number of pulses needed for the motor to complete one full rotation:

Pulse = (4 * encoder count * gearbox ratio) / coefficient

For example, we have a motor with 500 encoder pulses per revolution, without using a gearbox, meaning the gearbox ratio is 1, and the coefficient is 1 as shown in the illustration. Using the above formula, the required pulse count can be calculated as:

Pulse = 4 * 500 * 1 / 1 = 2000 pulses, so the motor will complete 1 full revolution.

For example, if the coefficient is 4 as shown on the side, the number of pulses required for the motor to complete 1 revolution is:



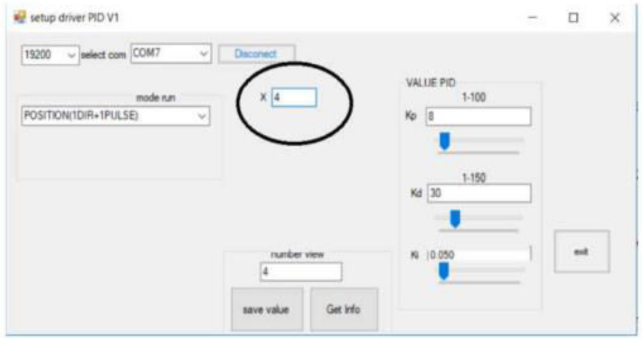

setup driver PID V1
19200 select com COM7 Disconect
mode run POSITION(1DIR+1PULSE)
X 4
VALUE PID
1-100 Kp 8
1-150 Kd 30
Ki 0.050
number view 4
save value Get Info exit

Figure 7: Select PID coefficient equal to 4

In this mode, the speed of the motor's rotation depends on the pulse frequency. Using the above formula, we have:

Pulse = 4 * 500 * 1 / 4 = 500 pulses

The driver supports a pulse frequency of up to 1 MHz.

III. BUILDING PID CIRCUIT CONNECTION FOR RICE HARVESTING ROBOT

A. PID Speed Mode

Open the setup software, in the run mode, select PID SPEED mode, and in the select mode section, choose PWM for controlling the driver via PWM pulses, or UART for controlling the driver through UART communication. In the speed round/minute field, enter the maximum number of revolutions the motor can achieve per minute, and in the encoder/1 round field, enter the number of pulses when the motor completes one full revolution. Then click "save value" to store the configured settings.

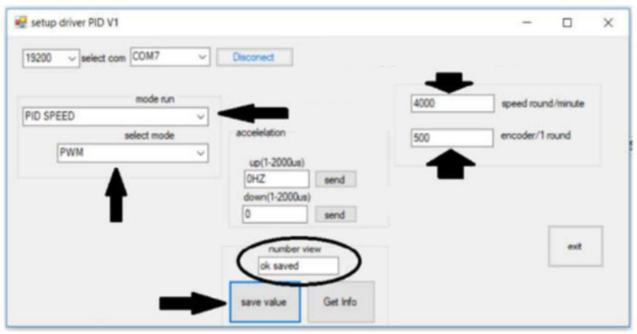

setup driver PID V1
19200 select com COM7 Disconect
mode run PID SPEED
select mode PWM
accelelation
up(1-2000us) 0HZ send
down(1-2000us) 0 send
4000 speed round/minute
500 encoder/1 round
number view ok saved
save value Get Info exit

Figure 8: Setup the PID mode for speed

PID SPEED Mode controlled by PWM

The driver is controlled through 3 pins:

Pin (3) Direction: controls the motor to rotate clockwise when the signal is at level 0, and counterclockwise when at level 1. Or vice versa. (used only in PWM pulse control mode)

- Pin (7) Pulse: this pin provides the pulse signal to operate the motor.
- Pin (1) Enable: when a logic signal of level 0 is applied, the driver operates with a pulse signal according to the RC Servo standard. When a logic signal of level 1 is applied, the driver operates with a PWM pulse signal at 8-bit resolution.

RC Pulse Control:

RC Servo pulse is a pulse with a frequency of 50Hz (period 20 ms), with a pulse width ranging from 1 to 2 ms.

Normally, when the pulse is at the Neutral Pulse state (pulse width 1.5 ms), the motor remains stationary. As the pulse width gradually increases from the Neutral Pulse state to the Maximum Pulse (increasing from 1.5 to 2 ms), the motor will rotate faster in the clockwise direction. When the pulse width decreases from the Neutral Pulse state to the Minimum Pulse (decreasing from 1.5 to 1 ms), the motor will rotate faster in the counterclockwise direction.

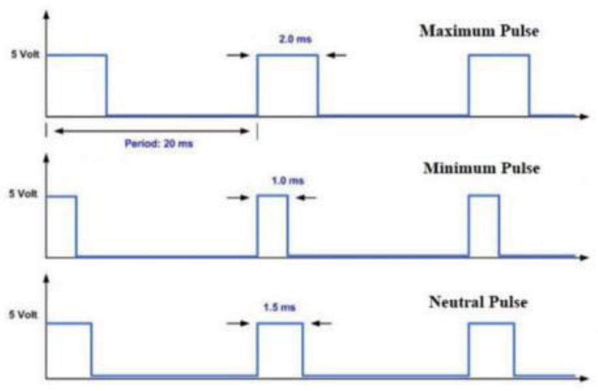


Figure 9: PID control with RC pulse

B. Control via PWM pulses

In this mode, the motor will rotate faster or slower depending on the pulse width of the PWM signal, as described in Figure 10. When the Duty Cycle = 0%, the motor remains stationary. As the Duty Cycle percentage increases, the motor will rotate faster in proportion to the Duty Cycle. The direction of the motor's rotation is controlled by the Dir pin. The driver supports PWM pulse frequencies ranging from 5 kHz to 50 Hz.

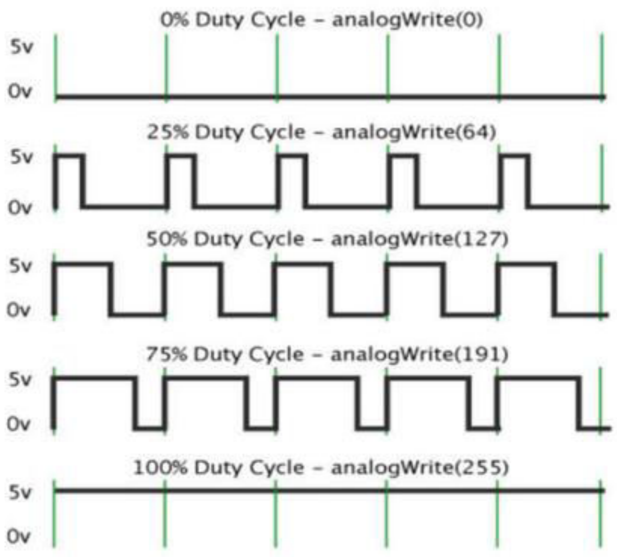

0% Duty Cycle - analogWrite(0)
5v
0v
25% Duty Cycle - analogWrite(64)
5v
0v
50% Duty Cycle - analogWrite(127)
5v
0v
75% Duty Cycle - analogWrite(191)
5v
0v
100% Duty Cycle - analogWrite(255)
5v
0v

Figure 10: PID control with PWM pulse

C. PID SPEED mode controlled by UART

In UART control mode, 3 bytes (in hex code) are continuously sent to the driver. The first byte contains attributes, with bit 7 indicating the rotation direction and the remaining bits being the address. The second byte represents the speed, and the third byte is always 0xFF. If the timeout is exceeded without sending data, the driver will stop.

For example, to set up the driver with address 1 as shown in Figure 5, to rotate forward at a speed of 150 (hex code 0x96), we send 3 bytes as follows: 0x81 0x96 0xFF

For reverse rotation at a speed of 50 (hex code 0x32): 0x01 0x32 0xFF Note: When setting the PID SPEED mode controlled by UART, the driver will not be able to reconnect to the computer. To reconnect, the Dir pin must be connected to GND. 11

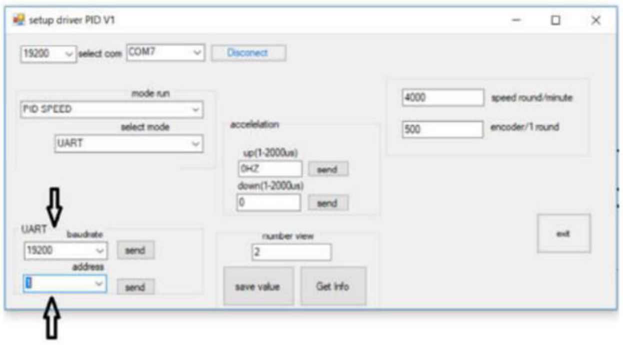


Figure 11: PID control via UART

D. H-Bridge Mode

Connect the computer to the driver, open the setup software, and in the run mode, select BRIDGE H. In the select mode, there are 3 control types similar to PID Speed mode; select PWM for control. In the number view, click "ok saved" to store the value.

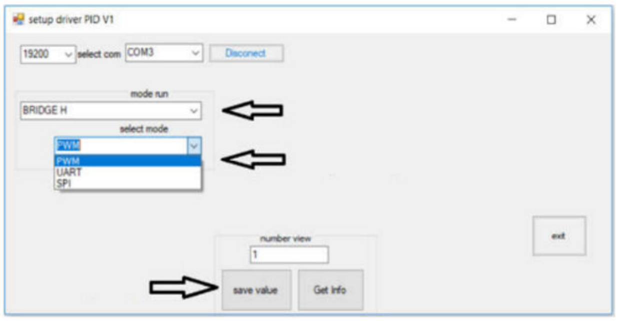


Figure 12: H-Bridge Mode

IV. DESIGN AND CONSTRUCTION OF THE ACTUATOR CONTROL CIRCUIT FOR THE RICE HARVESTING ROBOT

A. POSITION Mode (ENCODER AB PULSE)

Connect the computer to the driver, open the setup software, and in the run mode, select POSITION (ENCODER AB PULSE). In the PID VALUE section, select the parameters Kp, Kd, Ki, and then click "save value" to store the settings.

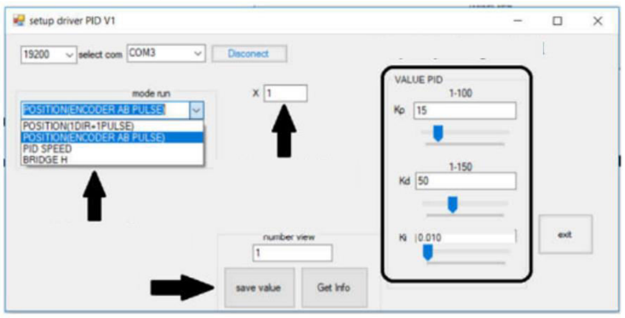


Figure 13: POSITION Mode

In this mode, another encoder is used to control the motor. The two AB channels of the encoder are connected to the pulse pin (pin 7) and the direction pin (pin 3). When the encoder rotates, the motor will rotate in proportion to the set ratio. For example, if the motor has 500 pulses per revolution and the encoder used for control has 200 pulses per revolution, with a coefficient set to 1 as shown in Figure 7, each pulse from the encoder will make the motor rotate 1 pulse of the encoder. This means that when the encoder rotates 2.5 times, the motor will rotate 1 full revolution. If the coefficient is set to 2, each pulse from the encoder will make the motor rotate 2 encoder pulses, so when the encoder rotates 1.25 times, the motor will rotate 1 full revolution.

Note: In this mode, the Enable pin must be connected to GND to allow the driver to operate.

B. Rice Harvesting Mechanism Control

To control the rice picking mechanism in the robot, use the central control circuit, the Arduino Mega 2560, through programming code on the Arduino IDE software, providing signals through the Digital output pins on the Arduino circuit.

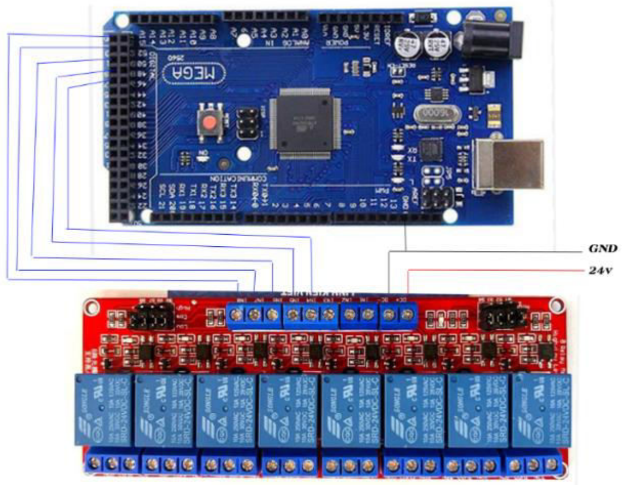


Figure 14: Connecting the 24V 8-Channel Relay Module with Arduino Mega 2560

Since the output signals of the Arduino Mega2560 circuit are 5V, they do not provide enough power to activate the solenoid coils of the pneumatic valves, which require a 24V voltage.

The 8-channel 24V relay circuit serves as an intermediary device that opens and closes contact cables to control the solenoid valves from the Arduino circuit to the output devices.

Connecting the power and control signals:

- **VCC of the relay module**: Connect to the 5V pin on the Arduino.
- **GND of the relay module**: Connect to the GND pin on the Arduino.
- **IN1 - IN8 (control signals) of the relay module**: Connect to the digital pins on the Arduino (for example, pins 22 to 29).

Connecting the devices to control:

- **COM (Common)**: Connect to the 24V power supply or the load to be controlled.
- **NO (Normally Open)**: Connect to the load (device to be controlled).
- **NC (Normally Closed)**: Generally not used unless you want the device to be always on and turned off when the relay is activated.

The rice-harvesting mechanism uses 5 output signal ports:

- IN 4 – Signal pin 22
- IN 5 – Signal pin 23
- IN 6 – Signal pin 24
- IN 7 – Signal pin 25
- IN 8 – Signal pin 26

The gripper mechanism uses a pneumatic cylinder widely used in industrial automation applications. Controlling this system with an Arduino Mega and a 24V 8-channel relay module increases the flexibility and accuracy of control.

C. Principle of Operation of the Gripper System Using a Pneumatic Cylinder

The pneumatic gripper system typically consists of the following main components:

- ***Pneumatic Cylinder:*** Converts pneumatic energy into mechanical motion to grip and pick up objects.
- ***Pneumatic Control Valve:*** Controls the airflow into and out of the cylinder to perform gripping and releasing actions.
- ***Pneumatic Power Supply:*** Provides compressed air to the system.
- ***Control Circuit (Arduino Mega):*** Controls the pneumatic valves via the relay module.

Using the Arduino Mega and the 24V 8-channel relay module to control the pneumatic gripper system offers many advantages such as high flexibility, easy control, and programming. This system can be widely applied in industrial automation applications, helping increase efficiency and precision in production processes.

- ***Pneumatic Cylinder:*** Used to perform the gripping and releasing actions.
- ***Pneumatic Control Valve (Solenoid Valve):*** Controls the airflow into and out of the cylinder.
- ***Pneumatic Power Supply:*** Provides air to the system.
- ***Arduino Mega 2560:*** Controls the pneumatic valve via the relay module.
- ***8-channel 24V relay circuit:*** Acts as an intermediary between the Arduino and the pneumatic valve.

The compressed air supply is connected to the pneumatic control valve (Solenoid Valve).

The pneumatic cylinder is connected to the control valve to perform gripping and releasing actions.

The Arduino Mega 2560 is connected to the 8-channel 24V relay circuit to control the pneumatic valves. The 8-channel 24V relay circuit receives control signals from the Arduino and activates the pneumatic valves.

The rice-harvesting gripper system uses pneumatic cylinders and is controlled by the Arduino Mega with the 8-channel 24V relay circuit, providing an effective solution for agricultural automation applications. With its precise and flexible control capabilities, this system helps increase productivity and work efficiency in the rice harvesting process.

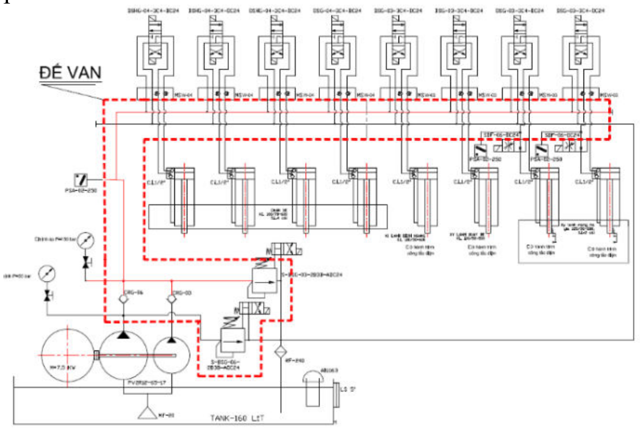


Figure 15: Pneumatic Circuit Diagram of the Rice Harvesting System

V. RESULTS AND DISCUSSIONS

The competitions have shown that the robot operates with high stability. The mechanical structures, combined with the control system, function flexibly and meet the requirements of the Vietnam Robot Innovation Competition 2024.

Figure 16: Pneumatic Circuit Diagram of the Rice Harvesting System

Optimization and Performance Enhancement: Specialized software has been applied to calculate, design, and simulate control circuits to meet the strict requirements of the competition, such as ensuring durability, mobility flexibility, and especially the robot's weight, in order to improve its automation and intelligence capabilities.

Application in Practical Production: Expanding research to apply the rice-harvesting robot to real production processes, thereby increasing productivity and efficiency in agriculture and industry. Developing robotic solutions tailored to the specific needs of different industries and production fields. Exploring and applying new technologies such as artificial intelligence and smart sensors in the design and development of robotic systems.

REFERENCES

[1]Tran The San, Nguyen Ngoc Phuong (2022), Practice of manufacturing remote-controlled robots. Science and Technology Publishing House.

[2]Nguyen Van Hiep (2022), Robot Engineering. Science and Technology Publishing House.

[3]Tran The San, Nguyen Ngoc Phuong (2022), Instructions for designing and assembling robots from common components. Science and Technology Publishing House.

[4]Duong Vinh Trung (2024), Smart robots - Basic applications. Ho Chi Minh City National University

Publishing House.
[5] https://vtv.vn/truyen-hinh/di-tim-chuc-vo-dich-robocon-viet-nam-2024-20240512114618996.htm