Tallinn University of Technology

Rudolf Põldma

212087IVEM

Final Project

ITS8050 – Embedded Software Workshop

Tallinn 2022

CONTENTS

1 INTRODUCTION

1.1 OVERVIEW

As a final exercise for the course, we had to create a project using Node-RED, MQTT and lab hardware. For this I created a DC motor project that can be controlled using Node-RED. This report describes the parts used, the design and logic behind the build.

1.2 GOALS

In this project, following elements were to be implemented:

1) Raspberry Pi
2) MQTT Broker
3) Node-RED

1.3 HARDWARE

The hardware used in this project:

1) Raspberry Pi 2 Model B
2) Arduino Uno
3) Wanptek DC Power Supply WPS3010H
4) 2x JQC-3FF-S-Z Relay
5) 12V DC motor with encoder

1.4 SOFTWARE

The software used in this project:

1) Raspberry Pi OS
2) Node-RED
3) Eclipse Mosquitto broker
4) Arduino IDE

2 DESIGN

2.1 OVERVIEW OF THE DESIGN

On Raspberry Pi Node-RED is ran that controls the Arduino, which controls the DC Motor. Arduino is used on this project because Arduino GPIO pins can output 5 Volts, while Raspberry Pi pins can only output 3.3 Volts. Relays need 5 V as its input to properly function. Two relays are used to drive the DC Motor both ways (Clockwise and counterclockwise).

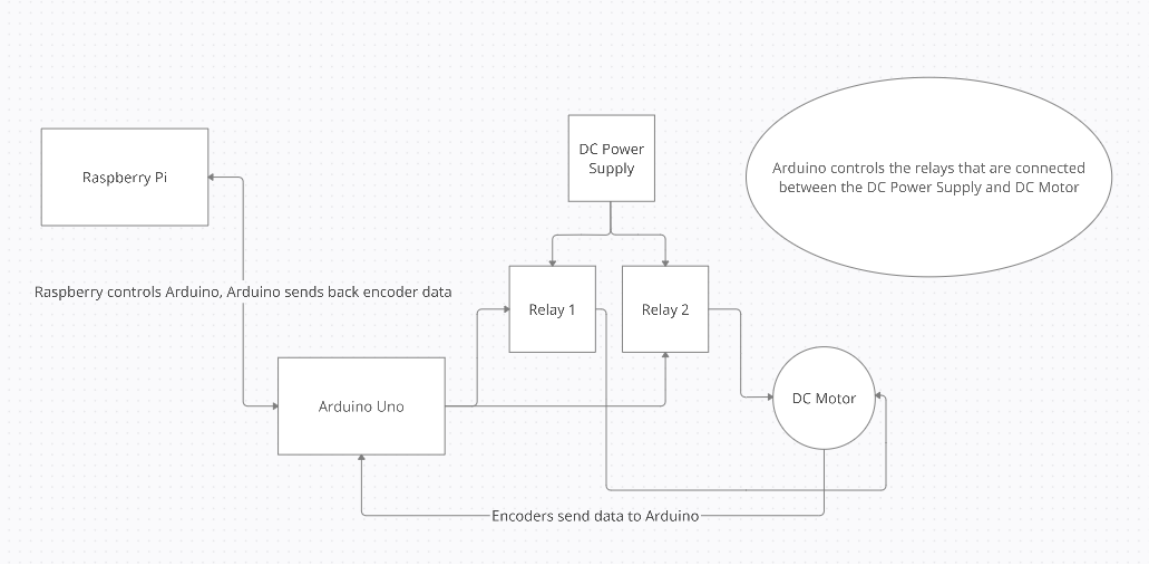

Figure 1. Design overview of the project.

2.2 RASPBERRY PI – ARDUINO LOGIC

Raspberry Pi is connected to Arduino with 2 GPIO pins:

1) Raspberry Pi pin 15 (GPIO22) - connected to Arduino pin 8.
2) Raspberry Pi pin 13 (GPIO27) - connected to Arduino pin 6.

With these pins, the Raspberry Pi tells Arduino to turn the DC motor on and move it either clockwise or counterclockwise.

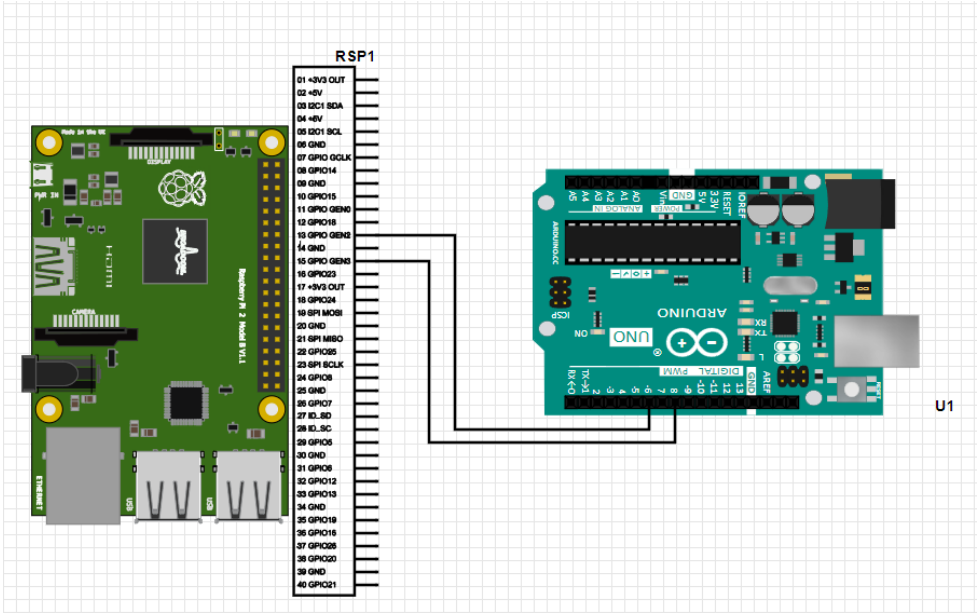

Figure 2. Raspberry Pi – Arduino Uno connection.

2.3 Arduino – relay – DC Motor logic

Arduino controls 2 relays, that decide which way the motor turns. Arduino pin 10 is connected to Relay 1 and pin 7 is connected to Relay 2. In Figure 3, DC Power Supply is missing, but in reality, it is connected to the relays.

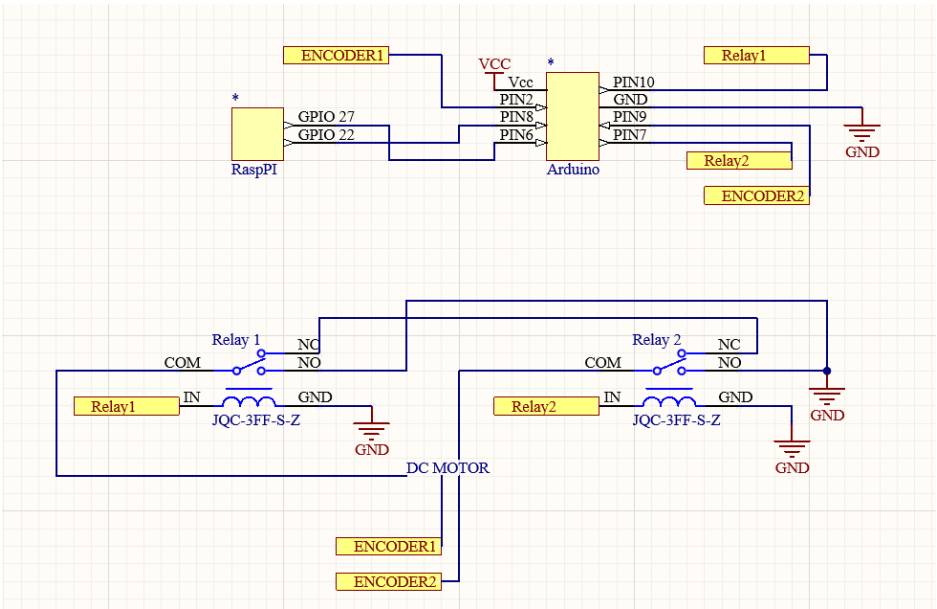

Figure 3. Simplified schematic of the logic.

2.4 ARDUINO CODE

To control the motor the following code was uploaded to Arduino (Figure 4). Testing the code, the attachInterrupt does not work, so it was commented out. So, the encoder values are incremented by the direction of the motor but not by its encoders.

```
#define encoder1 2
#define encoder2 9
#define relay1 10
#define relay2 7
#define raspi15 6
#define raspi22 8

int pos = 0;
int currentState = 9;
int lastState = 9;
int slowOutput = 1000;

void setup() {
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(encoder2, INPUT);
  pinMode(raspi22, INPUT);
  pinMode(raspi15, INPUT);
  pinMode(encoder1, INPUT);
  Serial.begin(9600);
  //attachInterrupt(digitalPinToInterrupt(encoder1), encoder, RISING);
}

// the loop function runs over and over again forever
void loop() {
  if(digitalRead(raspi22) == LOW && digitalRead(raspi15) == HIGH ){
    digitalWrite(relay1, HIGH);
    digitalWrite(relay2, LOW);
  if(slowOutput >= 1000){
    pos--;
    }
  }
  if(digitalRead(raspi22) == HIGH && digitalRead(raspi15) == LOW ){
    digitalWrite(relay1, LOW);
    digitalWrite(relay2, HIGH);
    if(slowOutput >= 1000){
    pos++;
    }
  }
  if(digitalRead(raspi22) == LOW && digitalRead(raspi15) == LOW){
    //
    digitalWrite(relay1, HIGH);
    digitalWrite(relay2, HIGH);
  }
  if(slowOutput >= 1000){
    Serial.println(pos);
    slowOutput = 0;
  }
  slowOutput ++;
}
void encoder(){
  currentState = digitalRead(encoder2);
  if (currentState == lastState){
    pos--;
  }
  else{
    pos++;
  }
  lastState = currentState;
}
```

Figure 4. Code uploaded to Arduino.

2.5 NODE-RED LOGIC

In Node-RED we can command the Raspberry Pi to send instructions to Arduino to turn the DC motor ON or OFF. The commands are sent through the MQTT broker. On the dashboard there are 3 buttons (OFF, Turn CW, Turn CCW). Data is sent back from Arduino to the RasPi to display the encoder values.

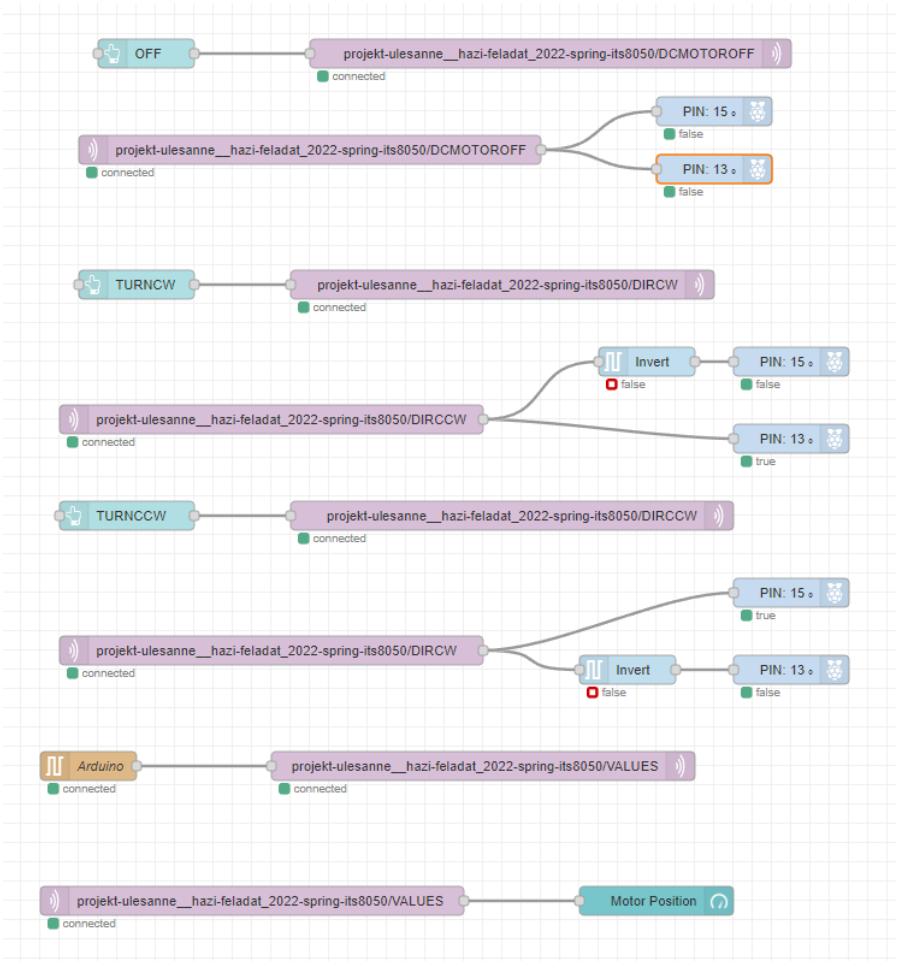

Figure 5. Node-RED flow.

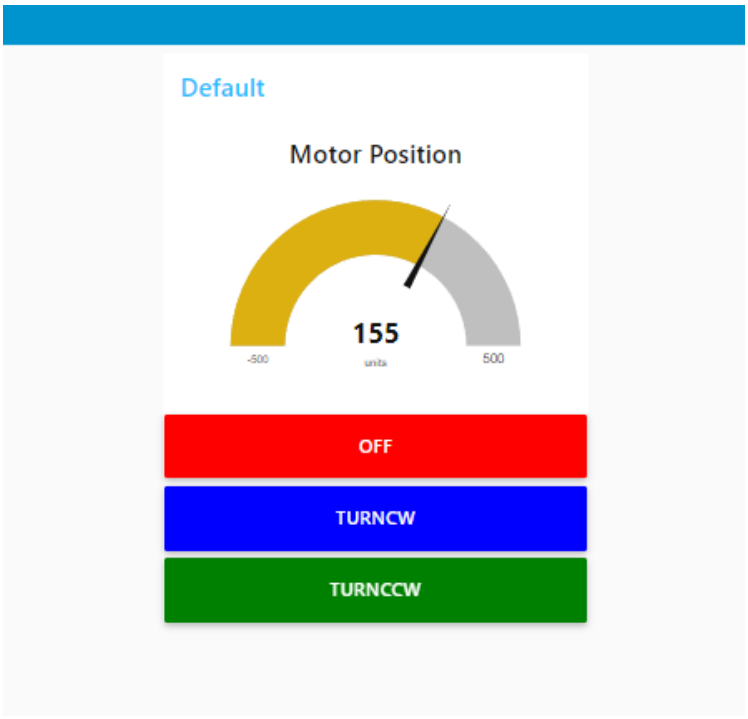

Figure 6. Node-RED dashboard.

2.6 PROJECT FILES

The project code, Node-RED flow and example video is uploaded to a public GitHub repository:

https://github.com/ruts9/ITS8050Project

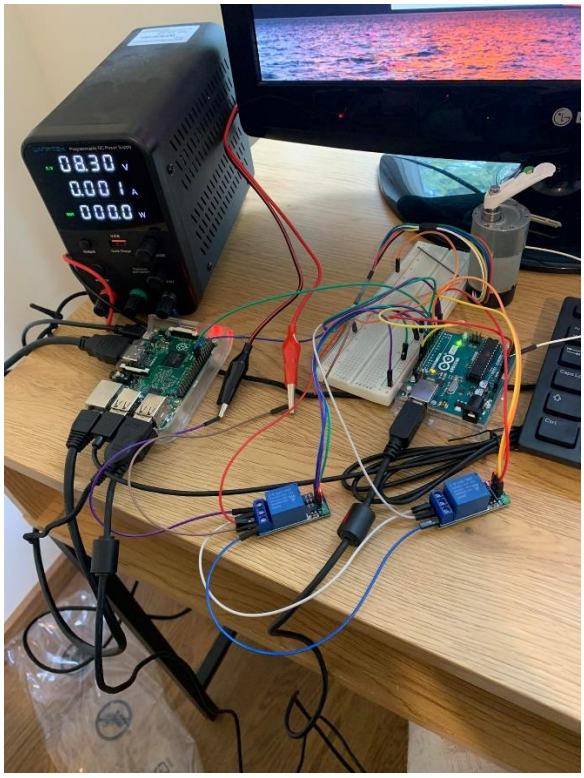

Figure 7. Picture of the project.

3 CONCLUSION

The project was completed successfully, with minor difficulties. The main problem was with the Arduino, where the interrupt worked incorrectly. Other than that, the DC motor could turn ON and OFF using Node-RED and the encoder data was sent back to be displayed on the dashboard.