SeAMK
SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Kiet Vu, Ha Truong, Chi Nguyen

Advanced Electronics Documentation

H-bridge connection controlling a DC motor

Assignment
Spring 2023
SeAMK Faculty
AE21

TABLE OF CONTENTS

1 THEORY

1.1 Introduction of DC motor control

H-bridge connection is a popular method for controlling the direction of a DC motor's rotation. It consists of four switches that can be turned on and off to change the polarity of the voltage applied to the motor. By controlling the switch configuration, an H-bridge connection can drive a DC motor forward, backward, or stop it. This technique is widely used in robotics, automation, and electric vehicle systems, where precise control of the motor's movement is essential.

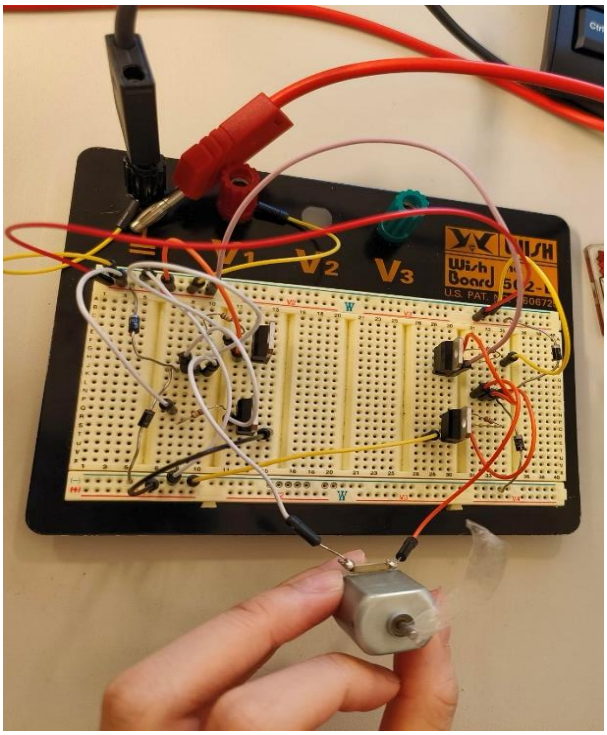
1.2 Components

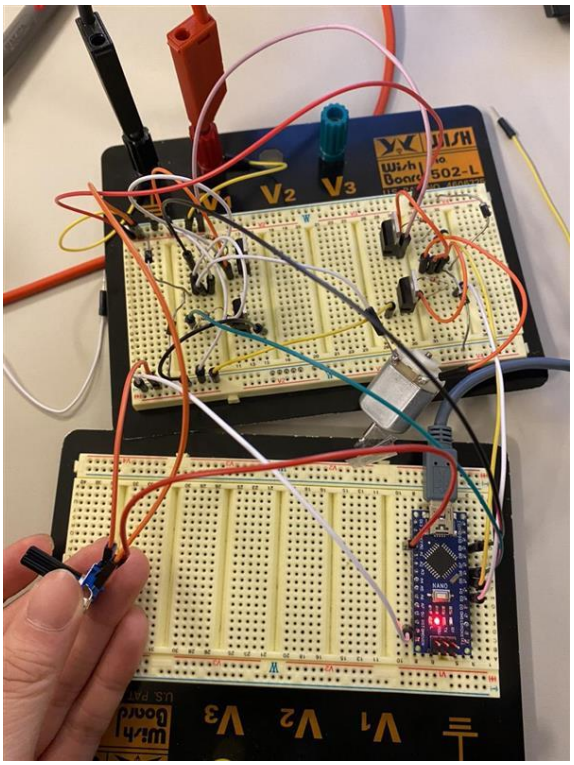
- 1 DC motor
- 4 diodes: Diodes provide a safe path for the motor energy to be dispersed or returned to the battery when the motor is commanded to coast or stop.
- 4 1k resistors: These resistors prevent too much current from passing through the base control pin of the transistor.
- NPN Transistors (TIP 122): They connect the motor to ground
- PNP transistors (TIP 127): They connect the motor to the power supply
- Power supply
- Arduino: control the direction/ rotation of the motor

2 PRACTICAL

2.1 Test circuit

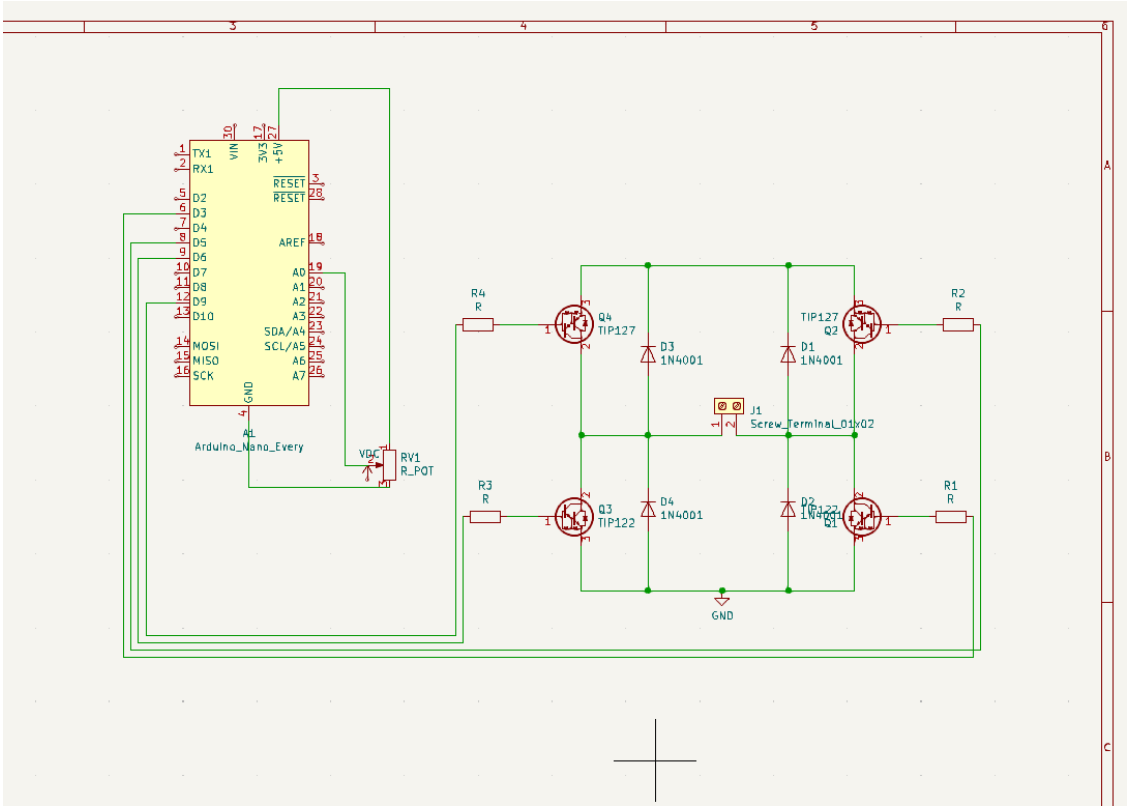
First, we took all the components we need for this assignment and tested them with this board. The first version is an H-bridge design without Arduino, and the second is with Arduino. And all the component works perfectly together.

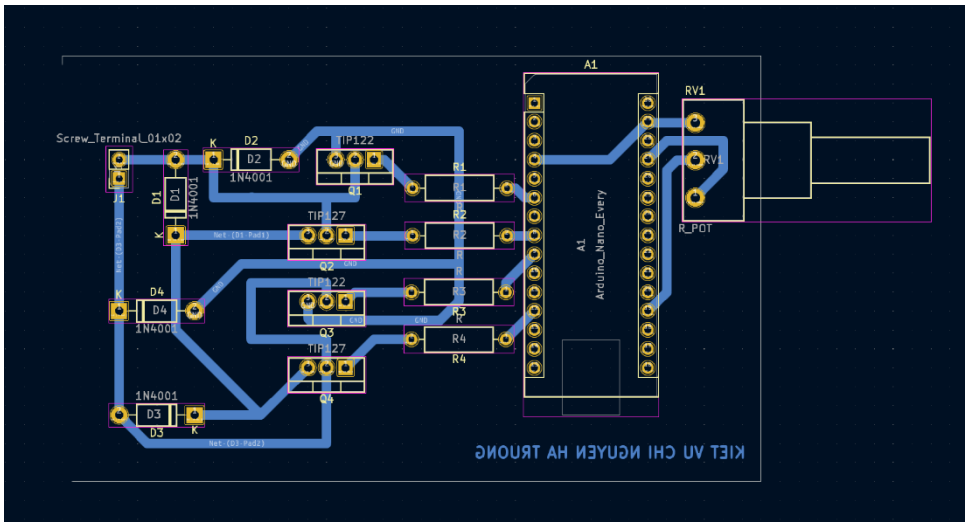




2.2 Circuit design

We utilized KiCAD software to design the circuit and establish the connections for all the components. Subsequently, we revised the design to ensure that it fits correctly onto the PCB board. During this process, we meticulously selected the appropriate footprint for each component and determined the suitable wire size and material. However, upon completion, we might discover that the design is incomplete and missing necessary connections, such as those linking the Arduino and motor. In such cases, we would need to add more tin to replace the missing wire and ensure the circuit functions optimally.





2.3 PCB manufacturing

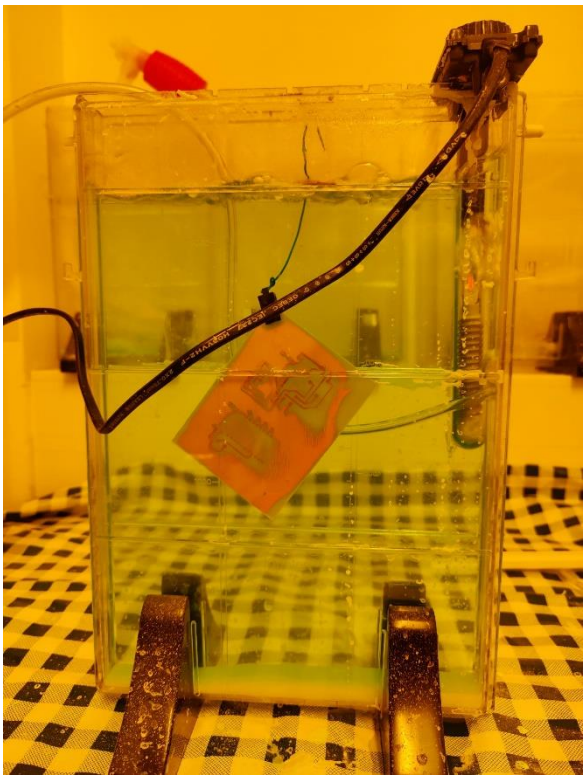
Peel off the tape of the circuit board

Paste the print onto the copper plate

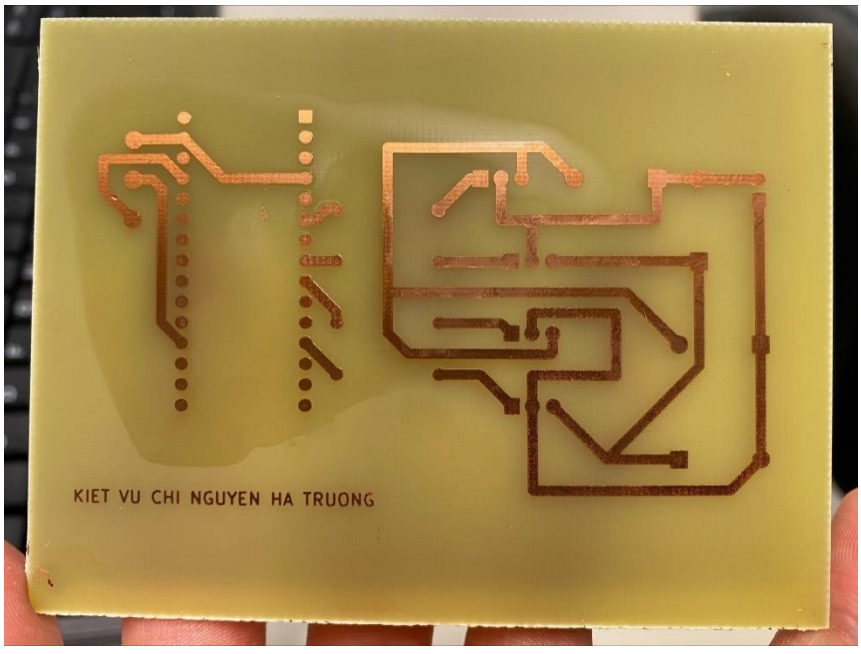
Put in UV machine - print on copper plate (1m30s)

Add the board into NaOH solution to make the image appear (10s) then quickly put the board into clean water

Remove the board from clean water and dip the entire board in Na_2SO4 solution in the PCB washer for about 10 minutes - the outer part of copper is lost, leaving only the desired part of the circuit board



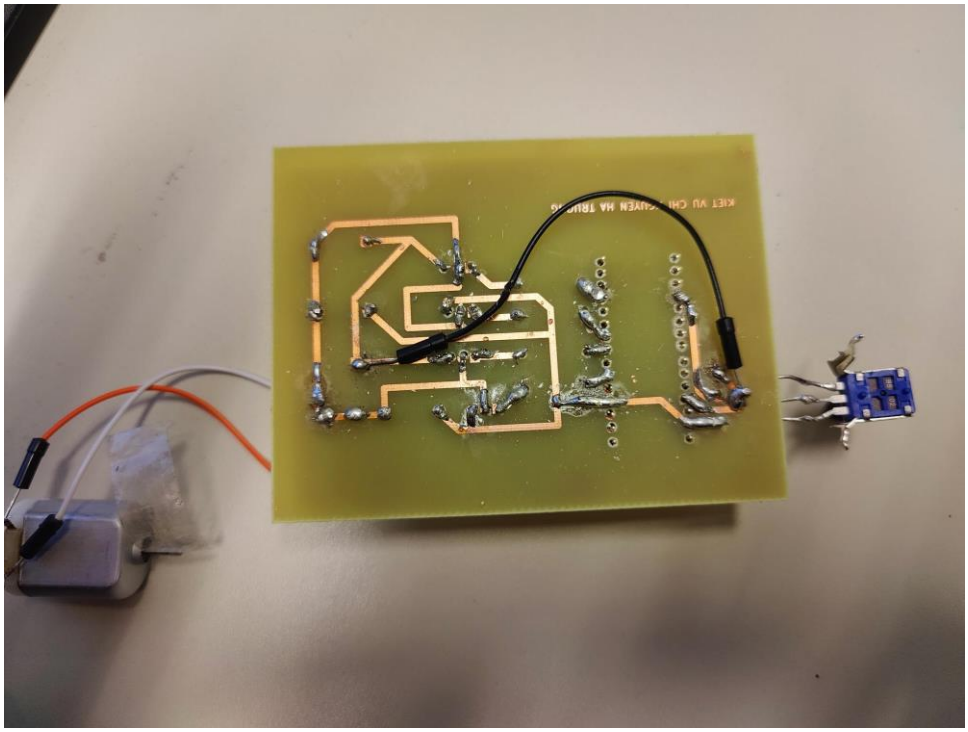
Our result after the process:



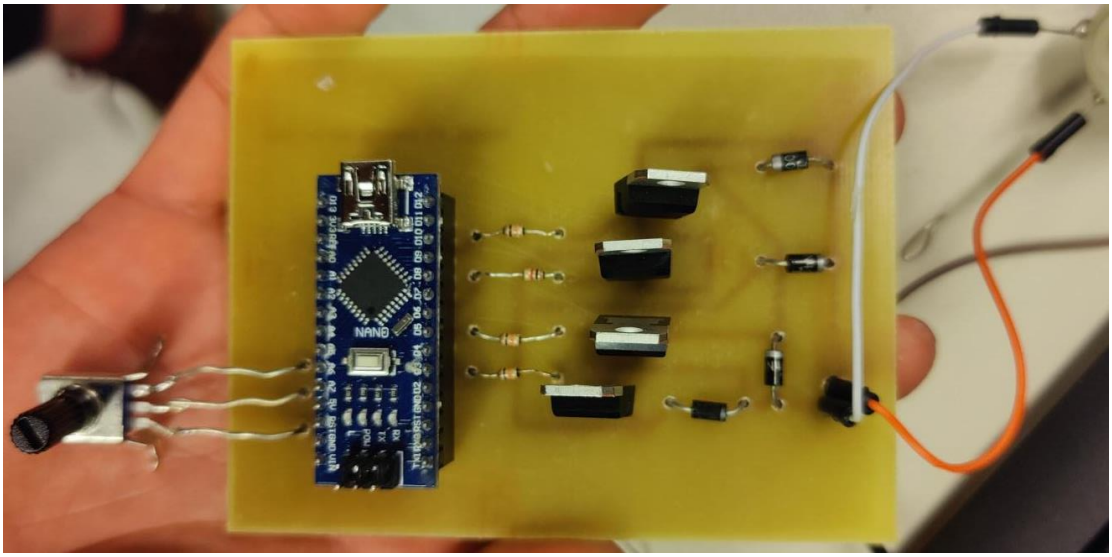
2.4 Components soldering

We used this machine to solder and stick components to PCB board with tin

This was our result, after soldering, as you can see we had to fix a lot of broken wires and replaced them with tin and added a missing wire.



This was the front page of PCB board



2.5 Testing

After soldering, we connected the Arduino to the computer via a USB cable and loaded this code. But the motor didn't run the first time as there were some connection issues so we had to adjust the wiring and re-soldered some areas.

Then, we could adjust the speed and make the motor rotate forward, backward, or stop, using the Arduino. The motor worked as intended.

```
#define pinr1 3 //resistor 1
#define pinr2 5
#define pinr3 6
#define pinr4 9 //resistor 4

void setup() {
pinMode(pinr1, OUTPUT);
pinMode(pinr2, OUTPUT);
pinMode(pinr3, OUTPUT);
pinMode(pinr4, OUTPUT);
digitalWrite(pinr1, LOW);
digitalWrite(pinr2, LOW);
digitalWrite(pinr3, LOW);
digitalWrite(pinr4, LOW);
Serial.begin(9600);
}
// the loop function runs over and over again forever
void loop()
{
  int potentiometer=analogRead(A0);
  if (potentiometer<400){
```

```
    forward(potentiometer);
  }
  else if (potentiometer>450){
    reverse(potentiometer);
  }
  else
  {
    off();
  }
}
void off(){
  digitalWrite(pinr1, LOW);
  digitalWrite(pinr2, LOW);
  digitalWrite(pinr3, LOW);
  digitalWrite(pinr4, LOW);
}
void forward(int potentiometer){
  Serial.print("forward ");
  Serial.print(potentiometer);
  potentiometer=map(potentiometer,400,0,50,254); //scale potentiometer value to
the pwm output
  Serial.print(" ");
  Serial.println(potentiometer);
  analogWrite(pinr3,  potentiometer);
  digitalWrite(pinr4, HIGH);
  digitalWrite(pinr2, LOW);
  digitalWrite(pinr1, LOW);
  }

  void reverse(int potentiometer)
  {
    Serial.print("reverse ");
    Serial.print(potentiometer);
    potentiometer=map(potentiometer,450,740,50,254); //scale potentiometer value
to to the PWM output
    if (potentiometer>254)
    {
      potentiometer=254;
    }
    Serial.print(" ");
    Serial.println(potentiometer);
    analogWrite(pinr1,  potentiometer);
    digitalWrite(pinr2, HIGH);
    digitalWrite(pinr3, LOW);
    digitalWrite(pinr4, LOW);
  }
```