MINI PROJECT, W2019

COURSE TITLE:

INTRODUCTION TO ELECTRIC VEHICLE

COURSE CODE:

EEL428

COURSE COORDINATOR:

DR. RITESH KESHRI, EED, VNIT Nagpur

GROUP MEMBERS:

Aniket Gujarathi	**BT16EEE005**
Dolly Jaiswal	**BT16EEE023**
Roma Gupta	**BT16EEE062**
Sumeet Kumar	**BT16EEE075**
Khalid Syed	**BT16EEE077**

STATEMENT OF PROJECT

Preparation a micro-controller for the generation of Gate-pulse a PM BLDC Motor Drive to be used in Light Electric Vehicle.

Apparatus Used:

1. Arduino Mega & Arduino UNO board(as per availability).
2. Potentiometer
3. Jumper wires
4. DSO
5. Connecting cables

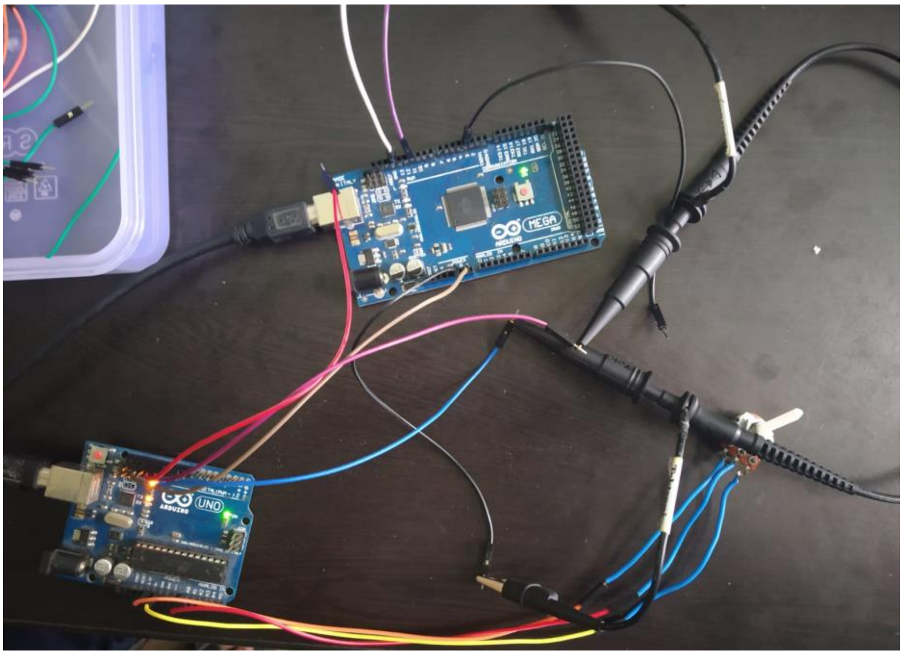
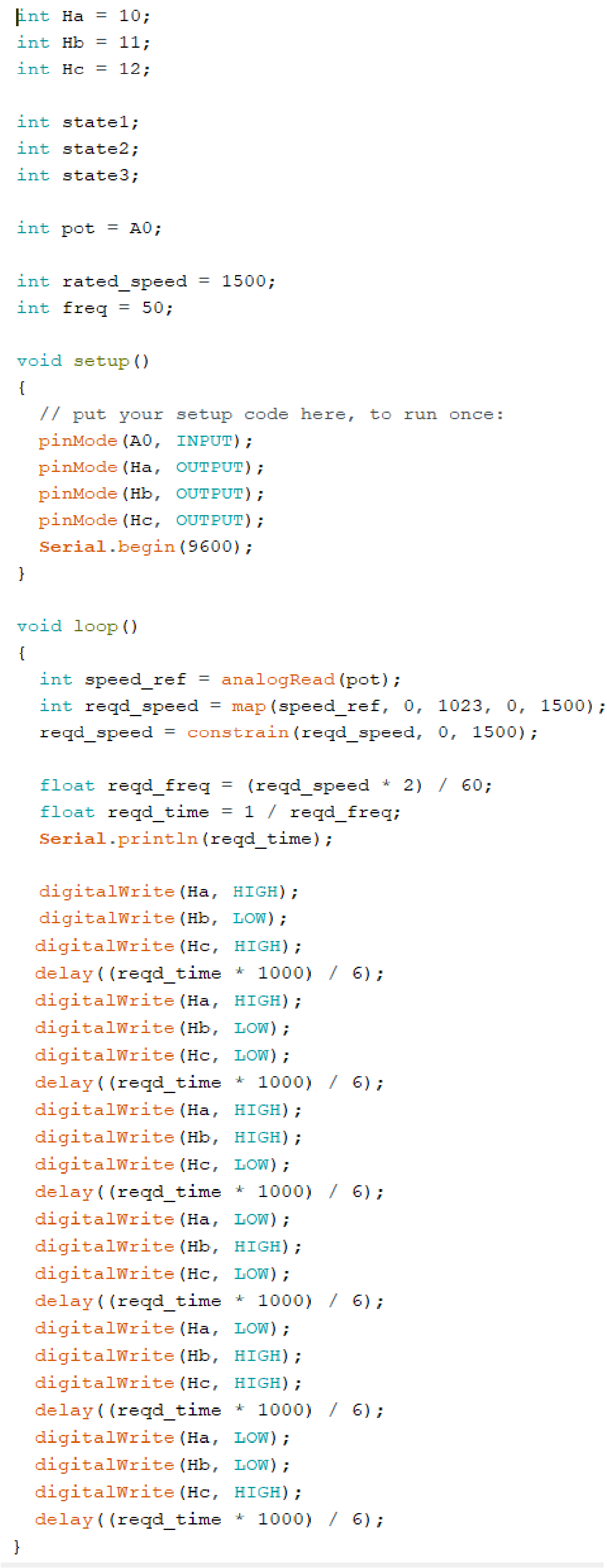


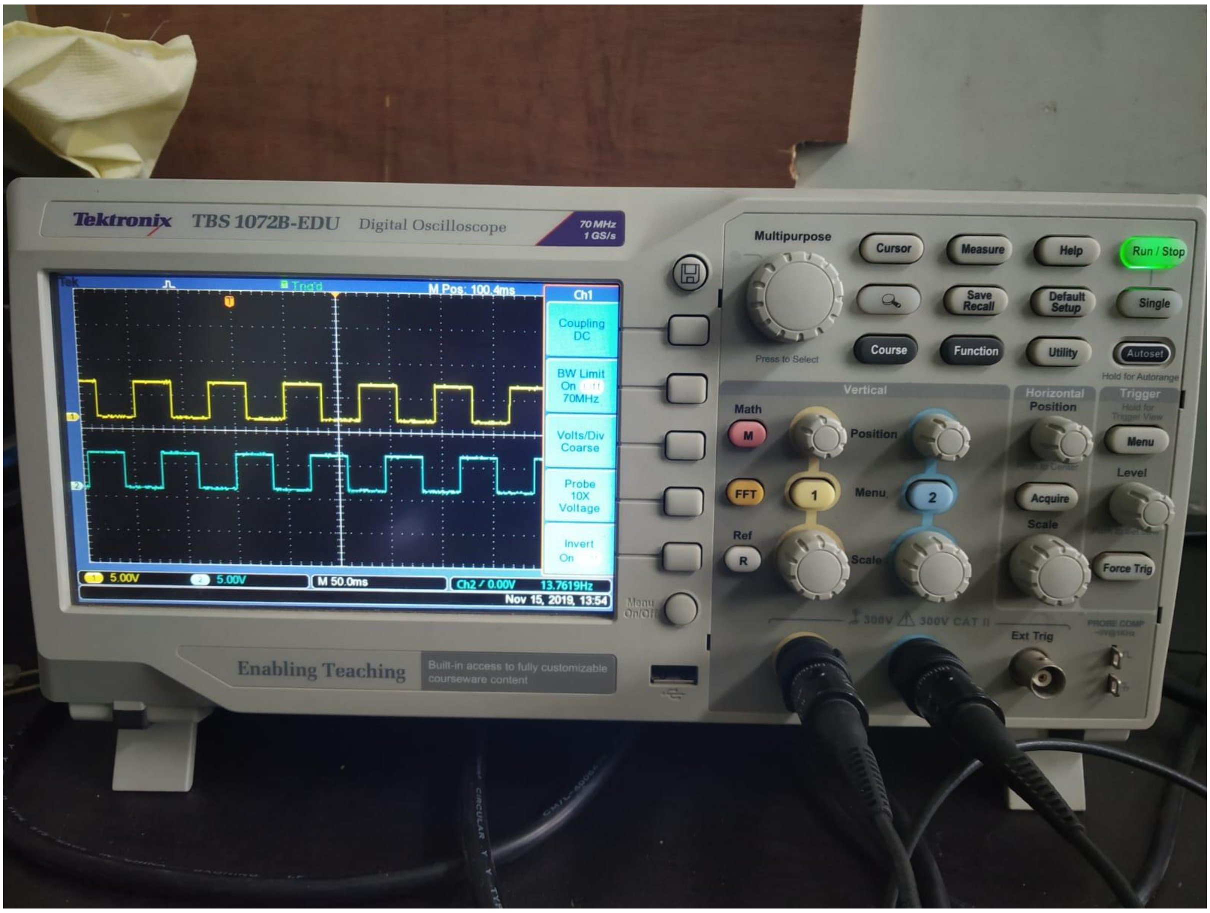
Figure 1: connection of two controllers with potentiometer to mimic speed control in PM BLDC

Hall Sensor Output Code:

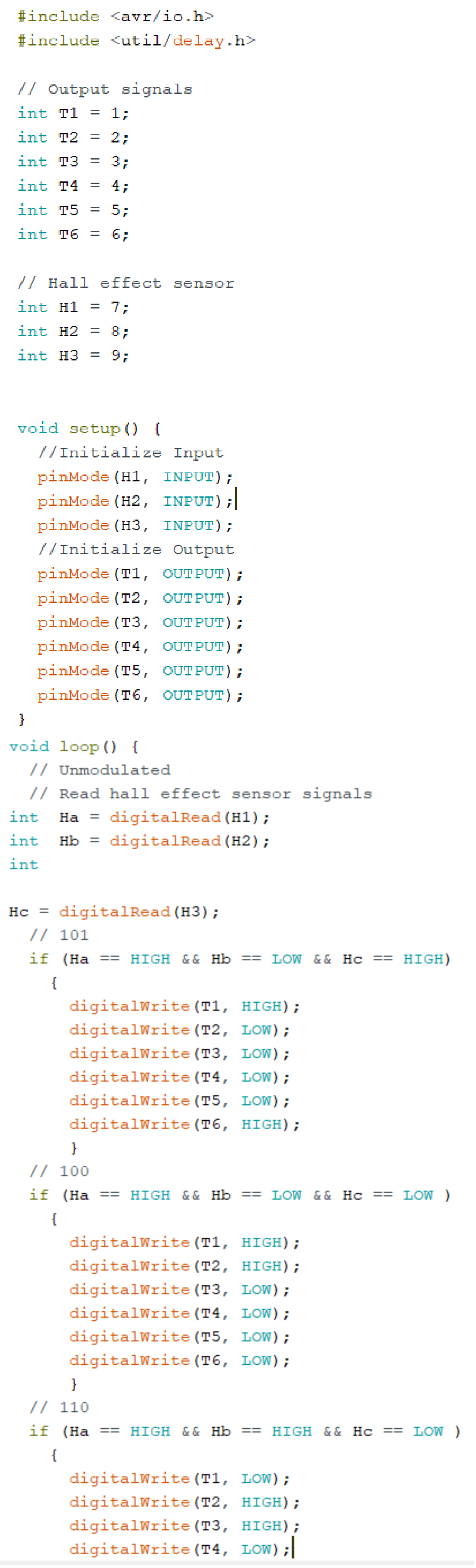


```
int Ha = 10;
int Hb = 11;
int Hc = 12;

int state1;
int state2;
int state3;

int pot = A0;

int rated_speed = 1500;
int freq = 50;

void setup()
{
  // put your setup code here, to run once:
  pinMode(A0, INPUT);
  pinMode(Ha, OUTPUT);
  pinMode(Hb, OUTPUT);
  pinMode(Hc, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int speed_ref = analogRead(pot);
  int reqd_speed = map(speed_ref, 0, 1023, 0, 1500);
  reqd_speed = constrain(reqd_speed, 0, 1500);

  float reqd_freq = (reqd_speed * 2) / 60;
  float reqd_time = 1 / reqd_freq;
  Serial.println(reqd_time);

  digitalWrite(Ha, HIGH);
  digitalWrite(Hb, LOW);
  digitalWrite(Hc, HIGH);
  delay((reqd_time * 1000) / 6);
  digitalWrite(Ha, HIGH);
  digitalWrite(Hb, LOW);
  digitalWrite(Hc, LOW);
  delay((reqd_time * 1000) / 6);
  digitalWrite(Ha, HIGH);
  digitalWrite(Hb, HIGH);
  digitalWrite(Hc, LOW);
  delay((reqd_time * 1000) / 6);
  digitalWrite(Ha, LOW);
  digitalWrite(Hb, HIGH);
  digitalWrite(Hc, LOW);
  delay((reqd_time * 1000) / 6);
  digitalWrite(Ha, LOW);
  digitalWrite(Hb, HIGH);
  digitalWrite(Hc, HIGH);
  delay((reqd_time * 1000) / 6);
  digitalWrite(Ha, LOW);
  digitalWrite(Hb, LOW);
  digitalWrite(Hc, HIGH);
  delay((reqd_time * 1000) / 6);
}
```

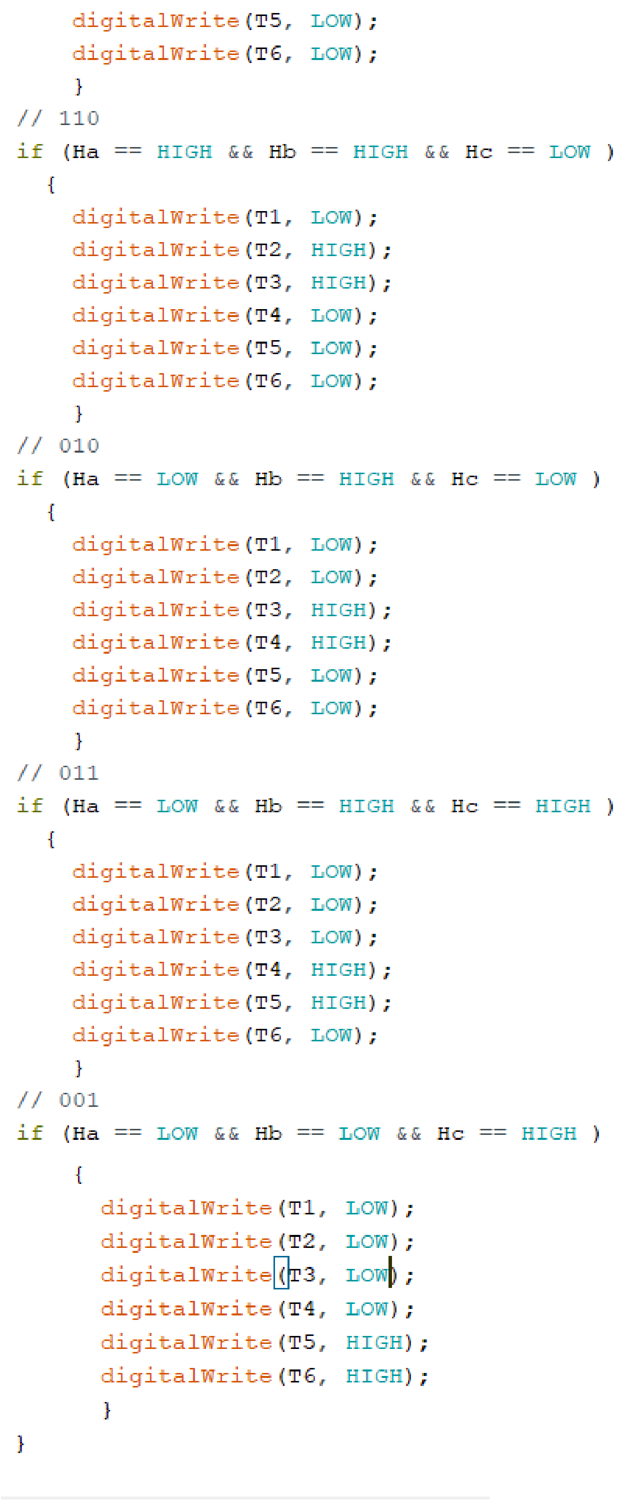
Hall sensor output:



Code for Unmodulated output:

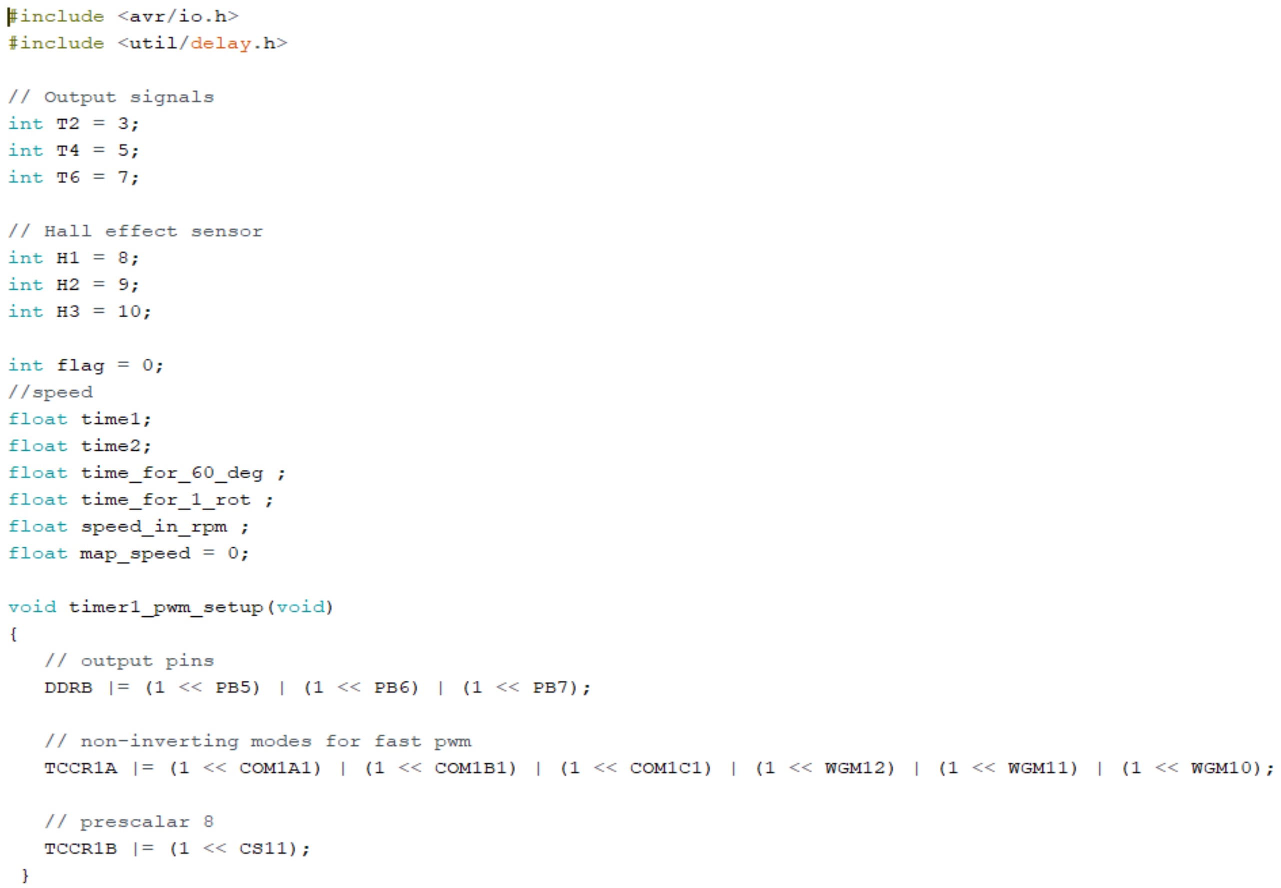


```
#include <avr/io.h>
#include <util/delay.h>

// Output signals
int T1 = 1;
int T2 = 2;
int T3 = 3;
int T4 = 4;
int T5 = 5;
int T6 = 6;

// Hall effect sensor
int H1 = 7;
int H2 = 8;
int H3 = 9;


void setup() {
  //Initialize Input
  pinMode(H1, INPUT);
  pinMode(H2, INPUT);
  pinMode(H3, INPUT);
  //Initialize Output
  pinMode(T1, OUTPUT);
  pinMode(T2, OUTPUT);
  pinMode(T3, OUTPUT);
  pinMode(T4, OUTPUT);
  pinMode(T5, OUTPUT);
  pinMode(T6, OUTPUT);
}
void loop() {
  // Unmodulated
  // Read hall effect sensor signals
int  Ha = digitalRead(H1);
int  Hb = digitalRead(H2);
int

Hc = digitalRead(H3);
  // 101
  if (Ha == HIGH && Hb == LOW && Hc == HIGH)
    {
      digitalWrite(T1, HIGH);
      digitalWrite(T2, LOW);
      digitalWrite(T3, LOW);
      digitalWrite(T4, LOW);
      digitalWrite(T5, LOW);
      digitalWrite(T6, HIGH);
      }
  // 100
  if (Ha == HIGH && Hb == LOW && Hc == LOW )
    {
      digitalWrite(T1, HIGH);
      digitalWrite(T2, HIGH);
      digitalWrite(T3, LOW);
      digitalWrite(T4, LOW);
      digitalWrite(T5, LOW);
      digitalWrite(T6, LOW);
      }
  // 110
  if (Ha == HIGH && Hb == HIGH && Hc == LOW )
    {
      digitalWrite(T1, LOW);
      digitalWrite(T2, HIGH);
      digitalWrite(T3, HIGH);
      digitalWrite(T4, LOW);
```

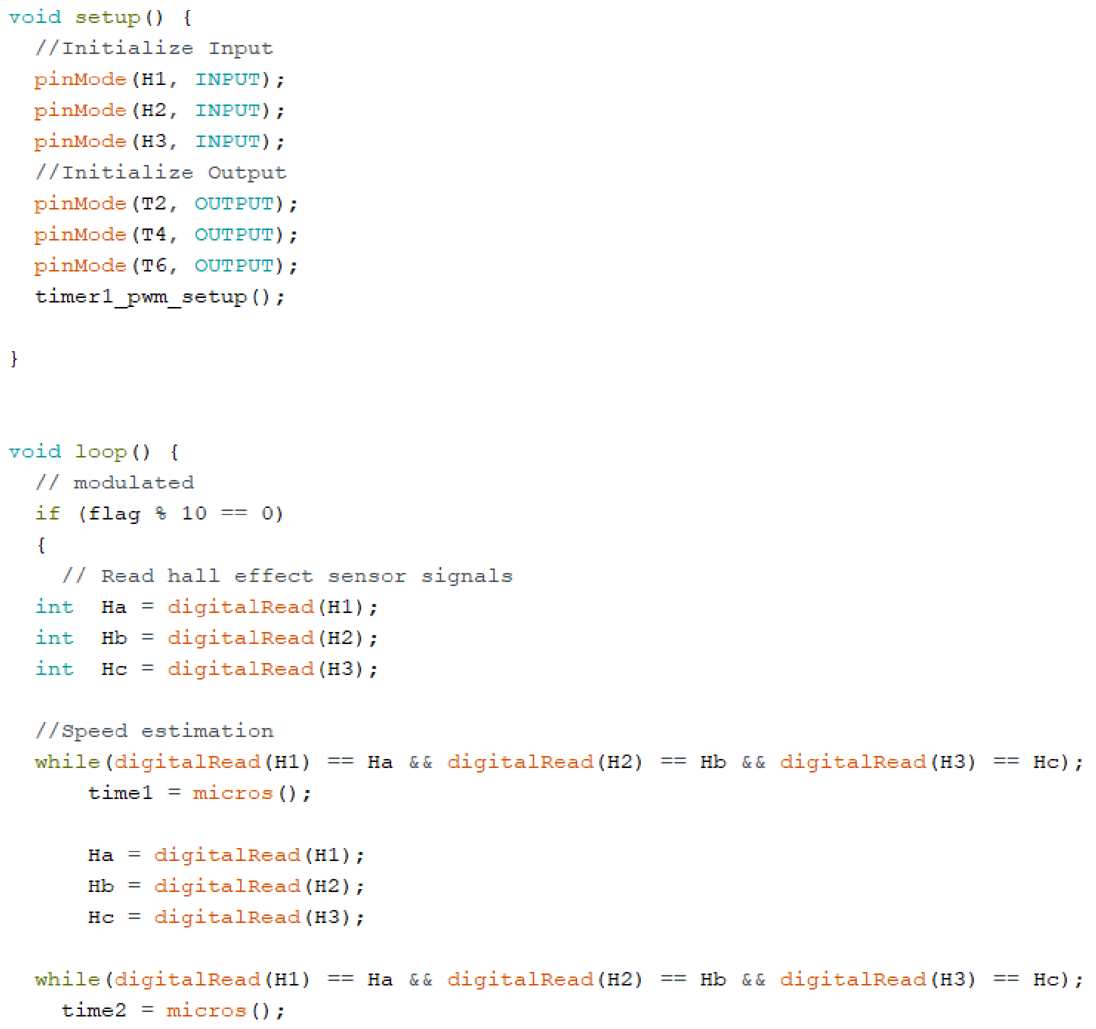


```
    digitalWrite(T5, LOW);
    digitalWrite(T6, LOW);
    }
// 110
if (Ha == HIGH && Hb == HIGH && Hc == LOW )
  {
    digitalWrite(T1, LOW);
    digitalWrite(T2, HIGH);
    digitalWrite(T3, HIGH);
    digitalWrite(T4, LOW);
    digitalWrite(T5, LOW);
    digitalWrite(T6, LOW);
    }
// 010
if (Ha == LOW && Hb == HIGH && Hc == LOW )
  {
    digitalWrite(T1, LOW);
    digitalWrite(T2, LOW);
    digitalWrite(T3, HIGH);
    digitalWrite(T4, HIGH);
    digitalWrite(T5, LOW);
    digitalWrite(T6, LOW);
    }
// 011
if (Ha == LOW && Hb == HIGH && Hc == HIGH )
  {
    digitalWrite(T1, LOW);
    digitalWrite(T2, LOW);
    digitalWrite(T3, LOW);
    digitalWrite(T4, HIGH);
    digitalWrite(T5, HIGH);
    digitalWrite(T6, LOW);
    }
// 001
if (Ha == LOW && Hb == LOW && Hc == HIGH )
    {
      digitalWrite(T1, LOW);
      digitalWrite(T2, LOW);
      digitalWrite(T3, LOW);
      digitalWrite(T4, LOW);
      digitalWrite(T5, HIGH);
      digitalWrite(T6, HIGH);
      }
}
```

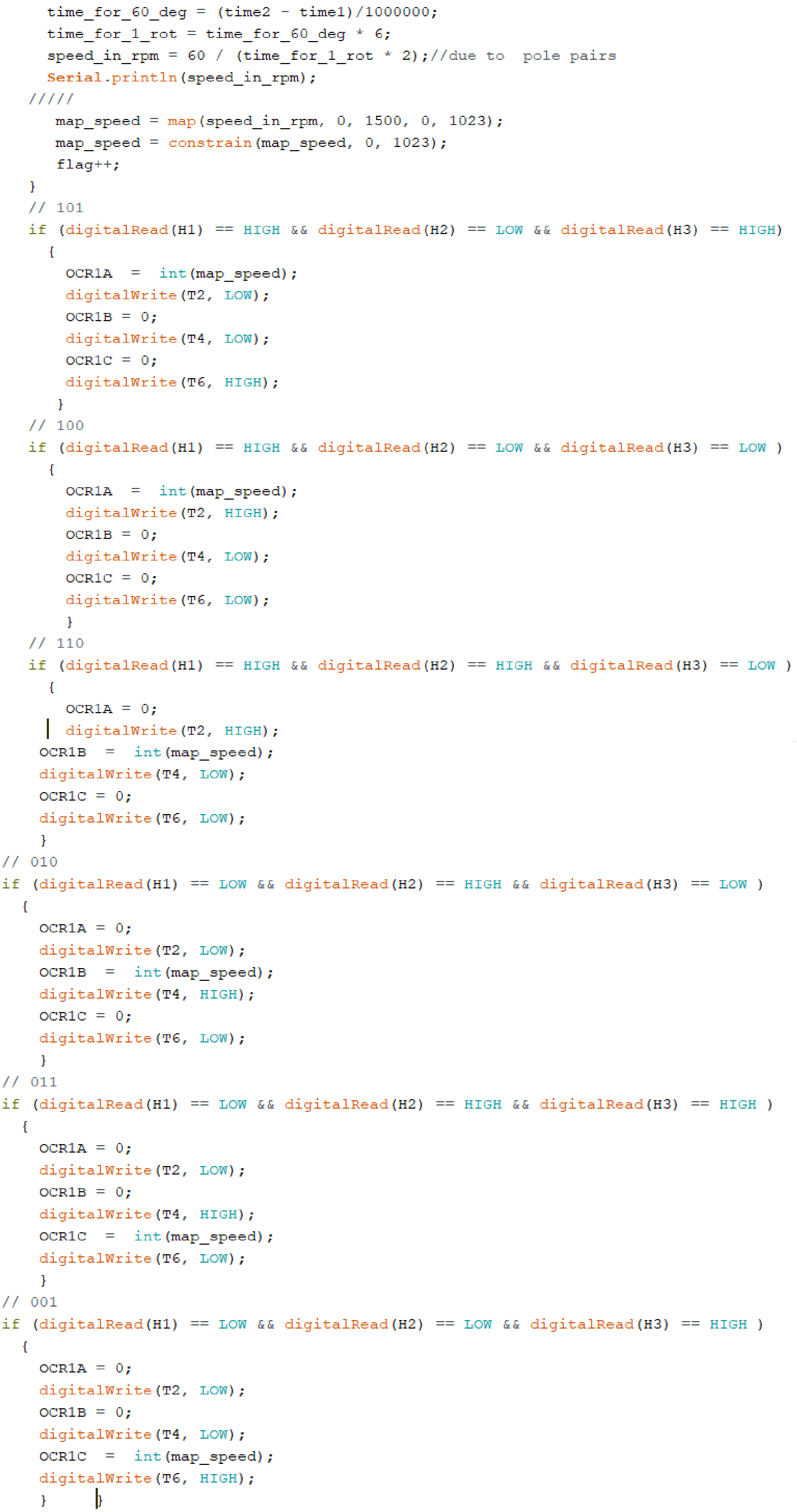
Code for Modulated output and speed estimation:



```
#include <avr/io.h>
#include <util/delay.h>

// Output signals
int T2 = 3;
int T4 = 5;
int T6 = 7;

// Hall effect sensor
int H1 = 8;
int H2 = 9;
int H3 = 10;

int flag = 0;
//speed
float time1;
float time2;
float time_for_60_deg ;
float time_for_1_rot ;
float speed_in_rpm ;
float map_speed = 0;

void timer1_pwm_setup(void)
{
   // output pins
   DDRB  |= (1 << PB5) | (1 << PB6) | (1 << PB7);

   // non-inverting modes for fast pwm
   TCCR1A |= (1 << COM1A1) | (1 << COM1B1) | (1 << COM1C1) | (1 << WGM12) | (1 << WGM11) | (1 << WGM10);

   // prescalar 8
   TCCR1B |= (1 << CS11);
 }
```

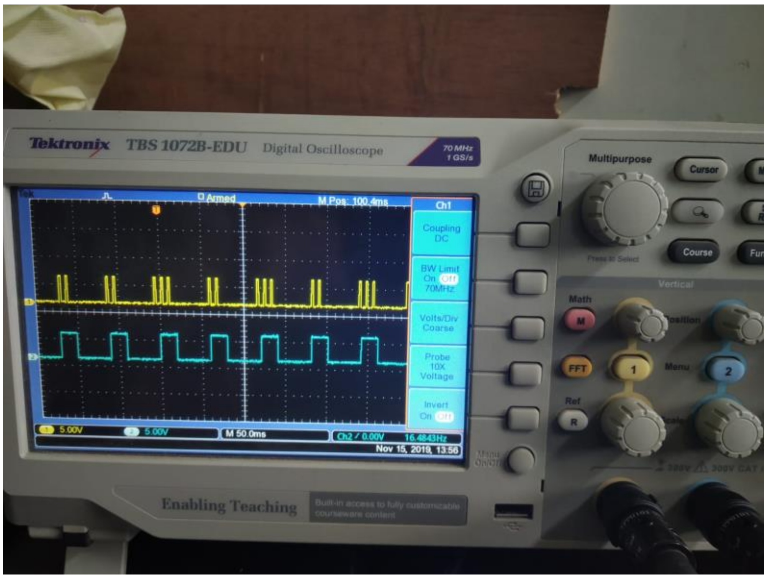


```
void setup() {
  //Initialize Input
  pinMode(H1, INPUT);
  pinMode(H2, INPUT);
  pinMode(H3, INPUT);
  //Initialize Output
  pinMode(T2, OUTPUT);
  pinMode(T4, OUTPUT);
  pinMode(T6, OUTPUT);
  timer1_pwm_setup();

}


void loop() {
  // modulated
  if (flag % 10 == 0)
  {
    // Read hall effect sensor signals
  int  Ha = digitalRead(H1);
  int  Hb = digitalRead(H2);
  int  Hc = digitalRead(H3);

  //Speed estimation
  while(digitalRead(H1) == Ha && digitalRead(H2) == Hb && digitalRead(H3) == Hc);
      time1 = micros();

      Ha = digitalRead(H1);
      Hb = digitalRead(H2);
      Hc = digitalRead(H3);

  while(digitalRead(H1) == Ha && digitalRead(H2) == Hb && digitalRead(H3) == Hc);
    time2 = micros();
```



```
     time_for_60_deg = (time2 - time1)/1000000;
     time_for_1_rot = time_for_60_deg * 6;
     speed_in_rpm = 60 / (time_for_1_rot * 2);//due to  pole pairs
     Serial.println(speed_in_rpm);
   /////
      map_speed = map(speed_in_rpm, 0, 1500, 0, 1023);
      map_speed = constrain(map_speed, 0, 1023);
      flag++;
   }
   // 101
   if (digitalRead(H1) == HIGH && digitalRead(H2) == LOW && digitalRead(H3) == HIGH)
     {
       OCR1A   =  int(map_speed);
       digitalWrite(T2, LOW);
       OCR1B = 0;
       digitalWrite(T4, LOW);
       OCR1C = 0;
       digitalWrite(T6, HIGH);
      }
   // 100
   if (digitalRead(H1) == HIGH && digitalRead(H2) == LOW && digitalRead(H3) == LOW )
     {
       OCR1A   =  int(map_speed);
       digitalWrite(T2, HIGH);
       OCR1B = 0;
       digitalWrite(T4, LOW);
       OCR1C = 0;
       digitalWrite(T6, LOW);
       }
   // 110
   if (digitalRead(H1) == HIGH && digitalRead(H2) == HIGH && digitalRead(H3) == LOW )
     {
       OCR1A = 0;
       digitalWrite(T2, HIGH);
    OCR1B   =  int(map_speed);
    digitalWrite(T4, LOW);
    OCR1C = 0;
    digitalWrite(T6, LOW);
    }
// 010
if (digitalRead(H1) == LOW && digitalRead(H2) == HIGH && digitalRead(H3) == LOW )
  {
    OCR1A = 0;
    digitalWrite(T2, LOW);
    OCR1B   =  int(map_speed);
    digitalWrite(T4, HIGH);
    OCR1C = 0;
    digitalWrite(T6, LOW);
    }
// 011
if (digitalRead(H1) == LOW && digitalRead(H2) == HIGH && digitalRead(H3) == HIGH )
  {
    OCR1A = 0;
    digitalWrite(T2, LOW);
    OCR1B = 0;
    digitalWrite(T4, HIGH);
    OCR1C   =  int(map_speed);
    digitalWrite(T6, LOW);
    }
// 001
if (digitalRead(H1) == LOW && digitalRead(H2) == LOW && digitalRead(H3) == HIGH )
  {
    OCR1A = 0;
    digitalWrite(T2, LOW);
    OCR1B = 0;
    digitalWrite(T4, LOW);
    OCR1C   =  int(map_speed);
    digitalWrite(T6, HIGH);
    }      }
```

Output on Modulated switch:



Output by varying speed by through potentiometer: