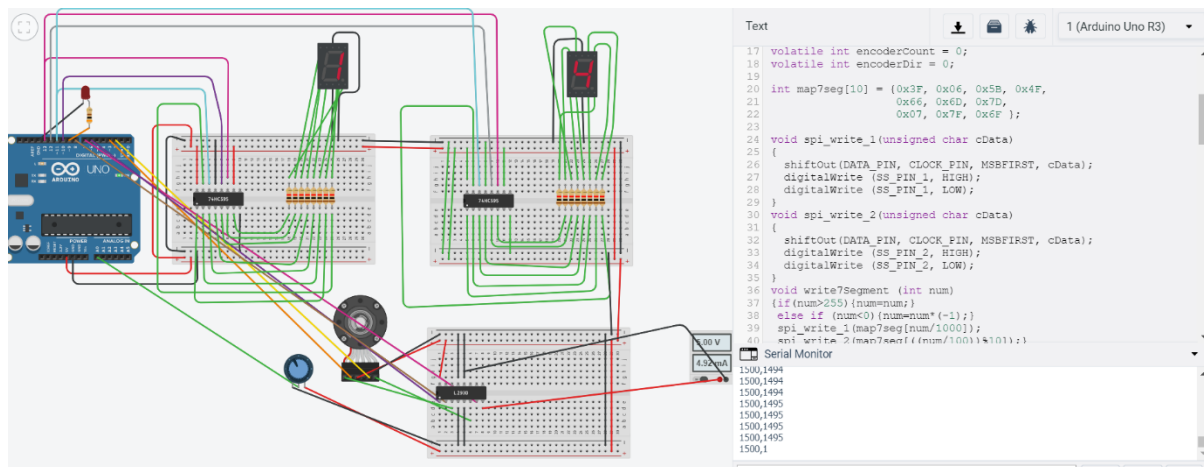
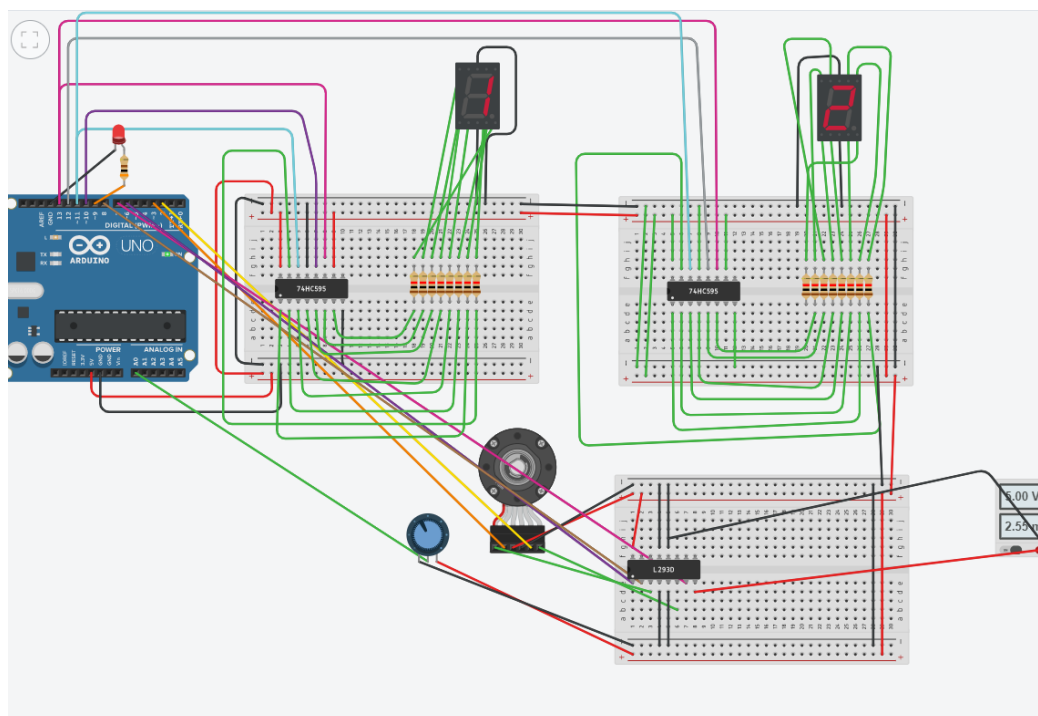


Q1

positive



Negative



```
#include <SPI.h>
```

```
#define DATA_PIN 11
```

```
#define CLOCK_PIN 13
```

```
#define SS_PIN_2 12
```

```
#define SS_PIN_1 10
```

```
#define MOTOR_D1_PIN 7
```

```
#define MOTOR_D2_PIN 8
```

```

#define MOTOR_PWM_PIN 6

#define LED_PIN 9

int interruptChannel1APin = 2;
int interruptChannel1BPin = 3;
int delay_count = 0;
int setpoint=0;
int toggle = 1;
volatile int Timer1_Counter;
volatile int num = 0;
volatile int encoderCount = 0;
volatile int encoderDir = 0;

int map7seg[10] = {0x3F, 0x06, 0x5B, 0x4F,
                  0x66, 0x6D, 0x7D,
                  0x07, 0x7F, 0x6F };

void spi_write_1(unsigned char cData)
{
    shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, cData);
    digitalWrite (SS_PIN_1, HIGH);
    digitalWrite (SS_PIN_1, LOW);
}

void spi_write_2(unsigned char cData)
{
    shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, cData);
    digitalWrite (SS_PIN_2, HIGH);
    digitalWrite (SS_PIN_2, LOW);
}

void write7Segment (int num)
{if(num>255){num=num;}
  else if (num<0){num=num*(-1);}
}

```

```
spi_write_1(map7seg[num/1000]);  
spi_write_2(map7seg[((num/100))%10]);}
```

```
void InitialInterruptTimer(int freq)  
{  
    noInterrupts();  
    TCCR1A = 0;  
    TCCR1B = 0;  
    Timer1_Counter = 62500/freq;  
    TCNT1 = Timer1_Counter;  
    TCCR1B |= (1 << CS12);  
    TIMSK1 |= (1 << TOIE1);  
    interrupts();  
}
```

```
void moveForward (int speed)  
{  
    digitalWrite(MOTOR_D1_PIN,HIGH);  
    digitalWrite(MOTOR_D2_PIN,LOW);  
    analogWrite(MOTOR_PWM_PIN,speed);  
}
```

```
void moveBackward (int speed)  
{  
    digitalWrite(MOTOR_D1_PIN,LOW);  
    digitalWrite(MOTOR_D2_PIN,HIGH);  
    analogWrite(MOTOR_PWM_PIN,speed);  
}
```

```
void setSpeed (int speed)  
{  
    if (speed>0)
```

```

    {if(speed>255){speed=255;}

      moveForward(speed);}
else if (speed<0)
{speed=speed*(-1);
  if (speed>255){speed=255;}

  moveBackward(speed);}
else
{moveForward(1);}
}

void setup()
{
  pinMode(MOTOR_D1_PIN,OUTPUT);
  pinMode(MOTOR_D2_PIN,OUTPUT);
  pinMode(MOTOR_PWM_PIN,OUTPUT);
  pinMode(SS_PIN_1, OUTPUT);
  pinMode(SS_PIN_2, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  pinMode(DATA_PIN, OUTPUT);
  pinMode(CLOCK_PIN, OUTPUT);
  pinMode(interruptChannel1APin,INPUT_PULLUP);
  pinMode(interruptChannel1BPin,INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(interruptChannel1APin),
    Channel1A_callback, RISING);
  attachInterrupt(digitalPinToInterrupt(interruptChannel1BPin),
    Channel1B_callback, RISING);
  Serial.begin(9600);
}

void loop()
{
  if(setpoint < 0)
    {digitalWrite(LED_PIN,HIGH);}

```

```

else if(setpoint > 0)
{digitalWrite(LED_PIN,LOW);}

int error = setpoint - encoderCount;

int p = 1.1*(float)(error);

write7Segment(encoderCount);

setSpeed(p);

Serial.print(setpoint);

Serial.print(",");

Serial.println(encoderCount);

delay(10);

}

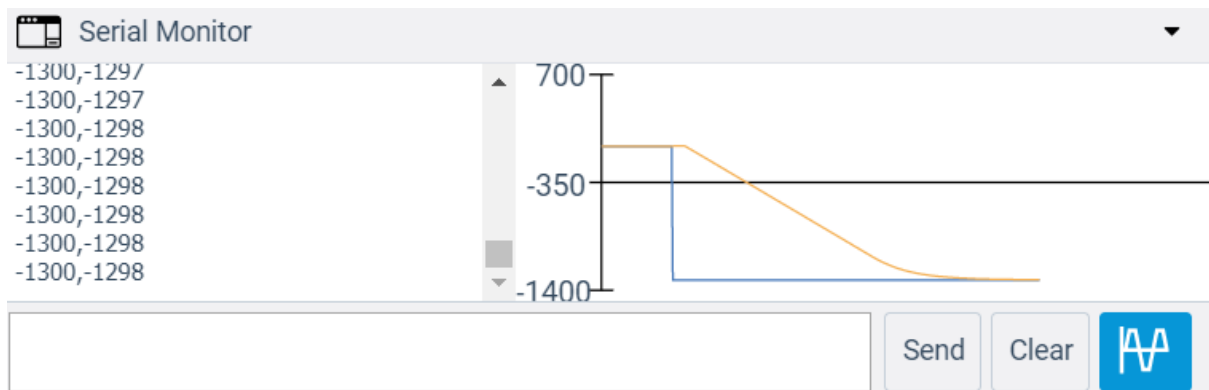
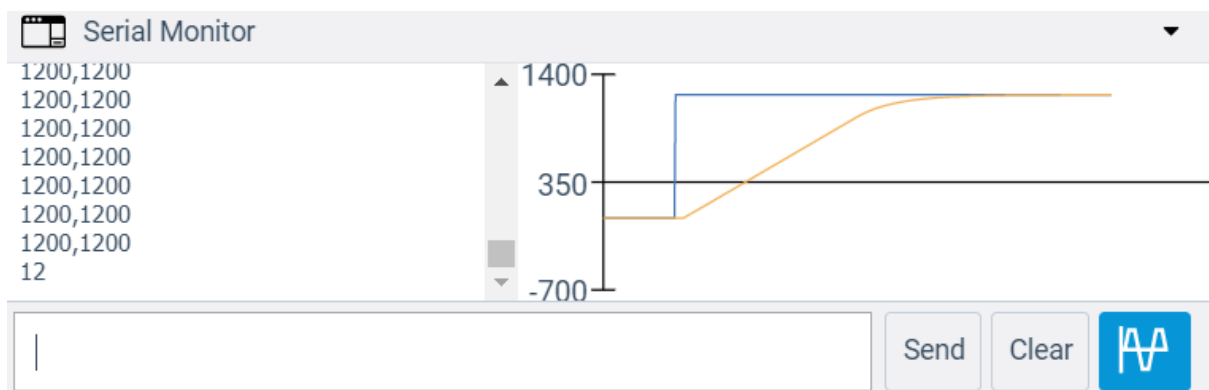
void Channel1A_callback()
{if (digitalRead(interruptChannel1APin)==1 &&
digitalRead(interruptChannel1BPin)==0)
{encoderCount++;
encoderDir = 1;}}

void Channel1B_callback()
{if (digitalRead(interruptChannel1APin)==0 &&
digitalRead(interruptChannel1BPin)==1)
{encoderCount--;
encoderDir = 0;}}

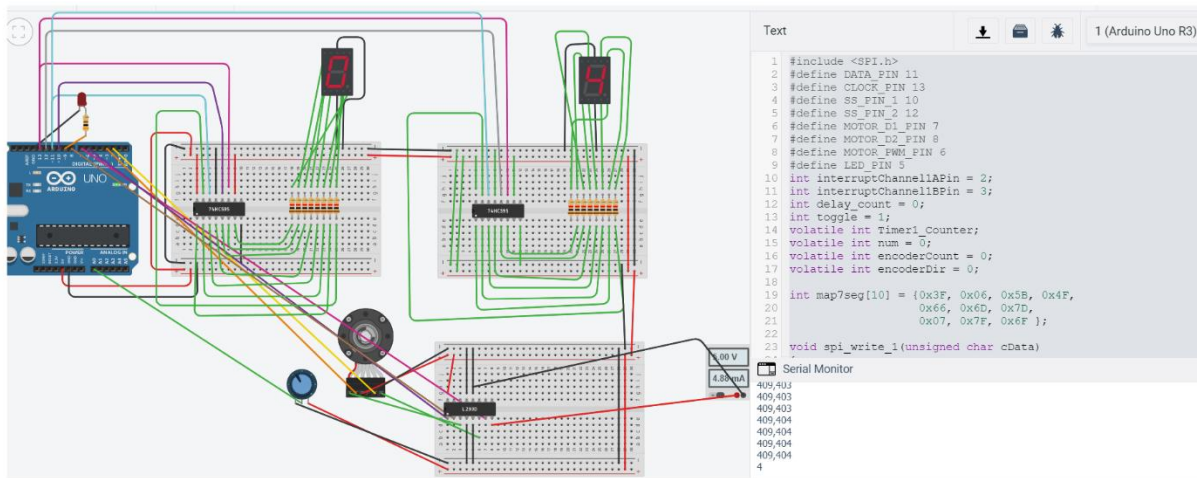
void serialEvent()
{setpoint = Serial.parseInt();}

```

Q2



Q3



```
#include <SPI.h>
```

```
#define DATA_PIN 11
```

```
#define CLOCK_PIN 13
```

```
#define SS_PIN_1 10
```

```
#define SS_PIN_2 12
```

```
#define MOTOR_D1_PIN 7
```

```
#define MOTOR_D2_PIN 8
```

```
#define MOTOR_PWM_PIN 6
```

```
#define LED_PIN 5
```

```
int interruptChannel1APin = 2;
```

```
int interruptChannel1BPin = 3;
```

```
int delay_count = 0;
```

```
int toggle = 1;
```

```
volatile int Timer1_Counter;
```

```
volatile int num = 0;
```

```
volatile int encoderCount = 0;
```

```
volatile int encoderDir = 0;
```

```
int map7seg[10] = {0x3F, 0x06, 0x5B, 0x4F,
```

```
0x66, 0x6D, 0x7D,
```

```
0x07, 0x7F, 0x6F };
```

```
void spi_write_1(unsigned char cData)
{
    shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, cData);
    digitalWrite (SS_PIN_1, HIGH);
    digitalWrite (SS_PIN_1, LOW);
}
```

```
void spi_write_2(unsigned char cData)
{
    shiftOut(DATA_PIN, CLOCK_PIN, MSBFIRST, cData);
    digitalWrite (SS_PIN_2, HIGH);
    digitalWrite (SS_PIN_2, LOW);
}
```

```
void write7Segment (int num)
{if(num>255){num=num;}
  else if (num<0){num=num*(-1);}
  spi_write_1(map7seg[num/1000]);
  spi_write_2(map7seg[((num/100))%10]);}
```

```
void InitialInterruptTimer(int freq)
{
    noInterrupts();
    TCCR1A = 0;
    TCCR1B = 0;
    Timer1_Counter = 62500/freq;
    TCNT1 = Timer1_Counter;
    TCCR1B |= (1 << CS12);
    TIMSK1 |= (1 << TOIE1);
    interrupts();
}
```



```
void moveForward (int speed)
{
    digitalWrite(MOTOR_D1_PIN,HIGH);
    digitalWrite(MOTOR_D2_PIN,LOW);
    analogWrite(MOTOR_PWM_PIN,speed);
}
```

```
void moveBackward (int speed)
{
    digitalWrite(MOTOR_D1_PIN,LOW);
    digitalWrite(MOTOR_D2_PIN,HIGH);
    analogWrite(MOTOR_PWM_PIN,speed);
}
```

```
void setSpeed (int speed)
{
    if (speed>0)
    {if(speed>255){speed=255;}
      moveForward(speed);}
    else if (speed<0)
    {speed=speed*(-1);
      if (speed>255){speed=255;}
      moveBackward(speed);}
    else
    {moveForward(1);}
}
```

```
void setup()
{
    pinMode(A0,INPUT);
    pinMode(MOTOR_D1_PIN,OUTPUT);
    pinMode(MOTOR_D2_PIN,OUTPUT);
}
```

```

pinMode(MOTOR_PWM_PIN,OUTPUT);

pinMode(SS_PIN_1, OUTPUT);

pinMode(SS_PIN_2, OUTPUT);

pinMode(LED_PIN, OUTPUT);

pinMode(DATA_PIN, OUTPUT);

pinMode(CLOCK_PIN, OUTPUT);

pinMode(interruptChannel1APin,INPUT_PULLUP);

pinMode(interruptChannel1BPin,INPUT_PULLUP);

attachInterrupt(digitalPinToInterrupt(interruptChannel1APin),
                Channel1A_callback, RISING);

attachInterrupt(digitalPinToInterrupt(interruptChannel1BPin),
                Channel1B_callback, RISING);

Serial.begin(9600);
}

void loop()
{
    int setpoint = analogRead(A0);
    if(setpoint < 0)
    {digitalWrite(LED_PIN,HIGH);}
    else if(setpoint > 0)
    {digitalWrite(LED_PIN,LOW);}

    int error = setpoint - encoderCount;
    int p = 1.1*(float)(error);
    write7Segment(encoderCount);
    setSpeed(p);
    Serial.print(setpoint);
    Serial.print(",");
    Serial.println(encoderCount);
    delay(10);
}

```

```
void Channel1A_callback()
{if (digitalRead(interruptChannel1APin)==1 &&
digitalRead(interruptChannel1BPin)==0)
{encoderCount++;
encoderDir = 1;}}
```

```
void Channel1B_callback()
{if (digitalRead(interruptChannel1APin)==0 &&
digitalRead(interruptChannel1BPin)==1)
{encoderCount--;
encoderDir = 0;}}
```

```
//void serialEvent()
//{setpoint = Serial.parseInt();}
```