SCHOOL OF ELECTRONICS & COMMUNICATION

FALL SEMESTER 2023 – 2024

ECE – 4003

EMBEDDED SYSTEM DESIGN

Professor: Sundar. S

LAB ASSESSMENT – 05

BUS INTERFACE TECHNIQUES

VINYAS A SHETTY

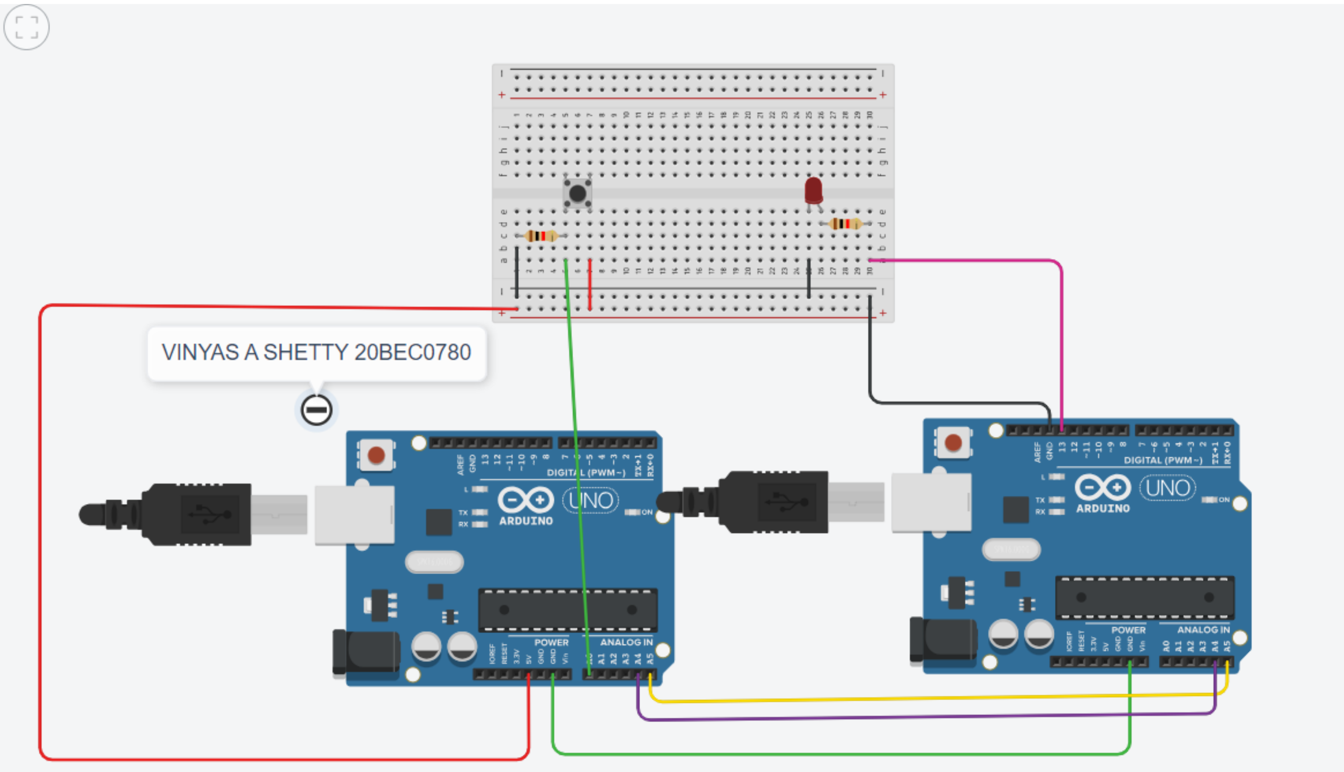
20BEC0780

L27+28

QUESTION 1:

Simulate a circuit for to demonstrate I2C protocol

CIRCUIT DIAGRAM:



CODE:

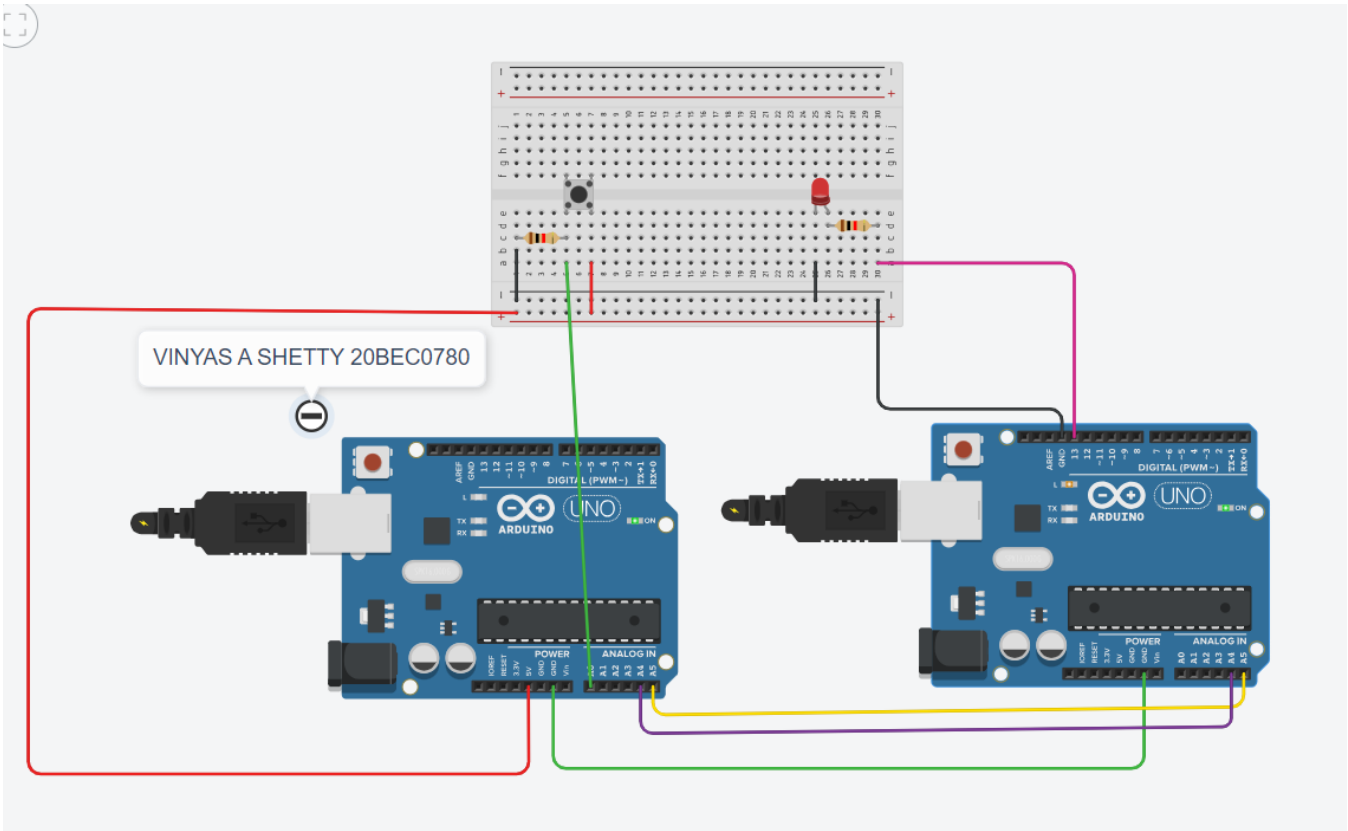
MASTER CODE:

```
#include <Wire.h>
int pushButton = A0;
int x = 0;
void setup()
{
  Wire.begin();
  pinMode(pushButton, INPUT);
}

void loop()
{
  Wire.beginTransmission(1);
  x = digitalRead(pushButton);
  Wire.write(x);
  Wire.endTransmission();
  delay(500);
}
```

SLAVE CODE:

```
#include <Wire.h>
int pinLed=13;
int x =0;
void setup()
{
 Wire.begin(1);
 Wire.onReceive(receiveEvent);
 pinMode(pinLed, OUTPUT);
}

void loop()
{
 delay(100);
}

void receiveEvent(int howMany){

x = Wire.read();
   if (x == 1){
        digitalWrite(pinLed,HIGH);
  }
 else{
      digitalWrite(pinLed,LOW);
  }
}
```

RESULTS SCREENSHOTS:



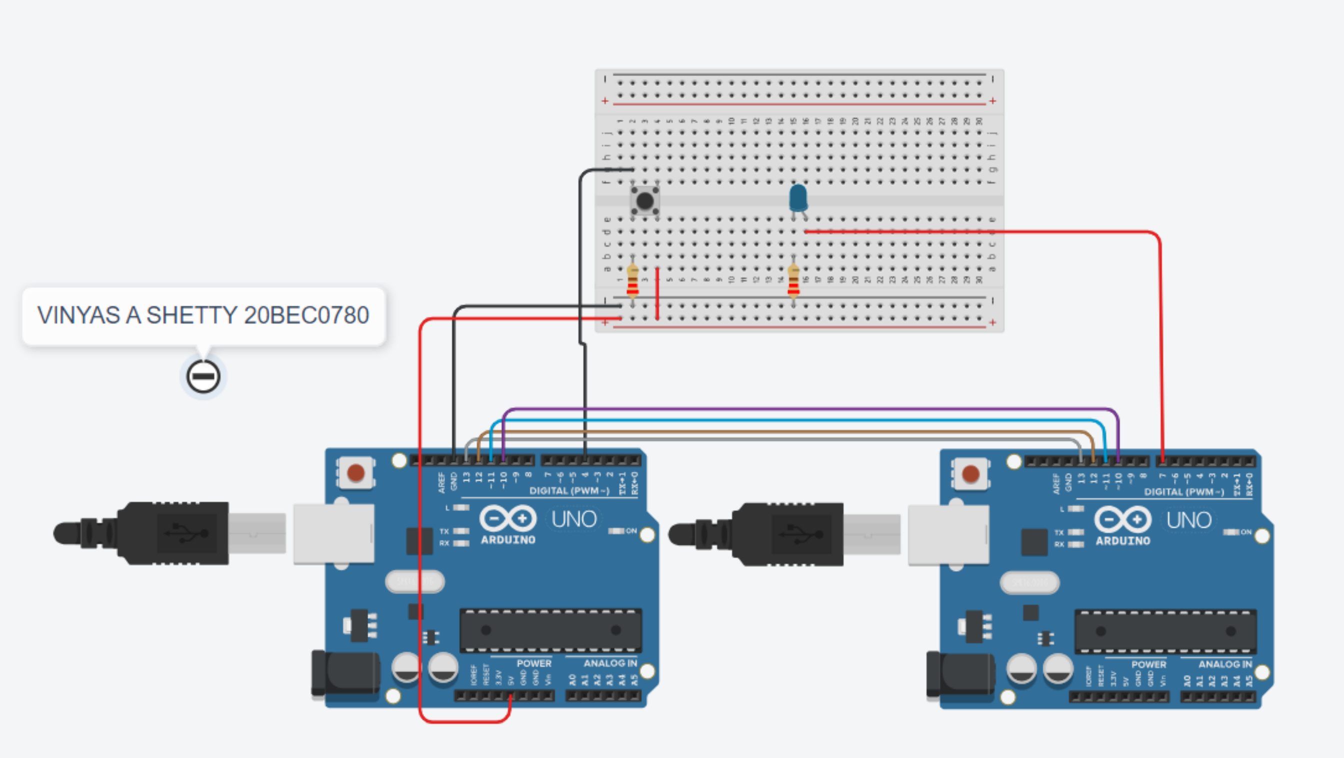
OBSERVATIONS:

We observed that the LED switches ON in the second Arduino only when you press the push button connected in the first Arduino.

QUESTION 2:

Simulate a circuit to demonstrate SPI protocol

CIRCUIT DIAGRAM:



CODE:

1ST ARDUINO CODE:

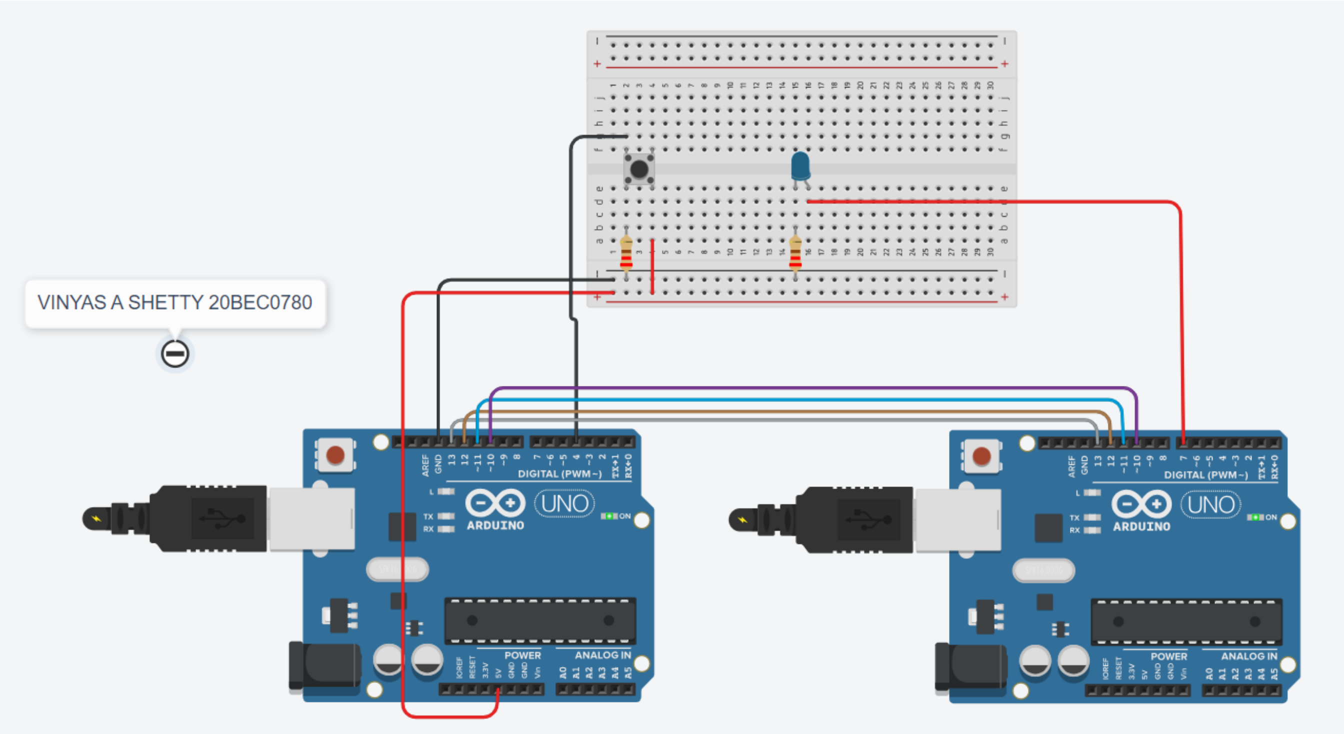
```
#include<SPI.h>
#define ipbutton 4
#define SS 10
int x;
int buttonvalue;

void setup() {
  Serial.begin(9600);
  pinMode(ipbutton, INPUT);
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV8);
  digitalWrite(SS,HIGH); //portas
}

void loop() {
  byte mastersend, mastereceive;
  buttonvalue = digitalRead(ipbutton);

  if(buttonvalue == HIGH){
    x = 1;
    Serial.println("Slave LED ON");
  }else{
    x = 0;
  }

  digitalWrite(SS, LOW);
  mastersend = x;
  mastereceive=SPI.transfer(mastersend);

  if(mastereceive == 1){
    Serial.println("LED ON");
  }else{
    Serial.println("LED OFF");
  }
  delay(1000);
}
```

2ND ARDUINO CODE:

```
#include<SPI.h>
bool received;
byte Slavereceived,Slavesend;
int buttonvalue;
int x;
```

```
void setup(){
  Serial.begin(9600);
  pinMode(7,OUTPUT);
  pinMode(MISO,OUTPUT);
  SPCR |= _BV(SPE);
  received = 0;
  SPI.attachInterrupt();
}

ISR (SPI_STC_vect){
  Slavereceived = SPDR;
  received = 1;
}
void loop(){
  if(received){
     if (Slavereceived==1){
          digitalWrite(7,HIGH);
          Serial.println("Slave LED ON");
        }else{
          digitalWrite(7,LOW);
          Serial.println("Slave LED OFF");
      }
  buttonvalue = digitalRead(Slavereceived);
  if(buttonvalue == HIGH){
     x=1;
  }else{
     x=0;
  }
  Slavesend=x;
  SPDR = Slavesend;
  delay(1000);
 }
}
```

RESULTS SCREENSHOTS:



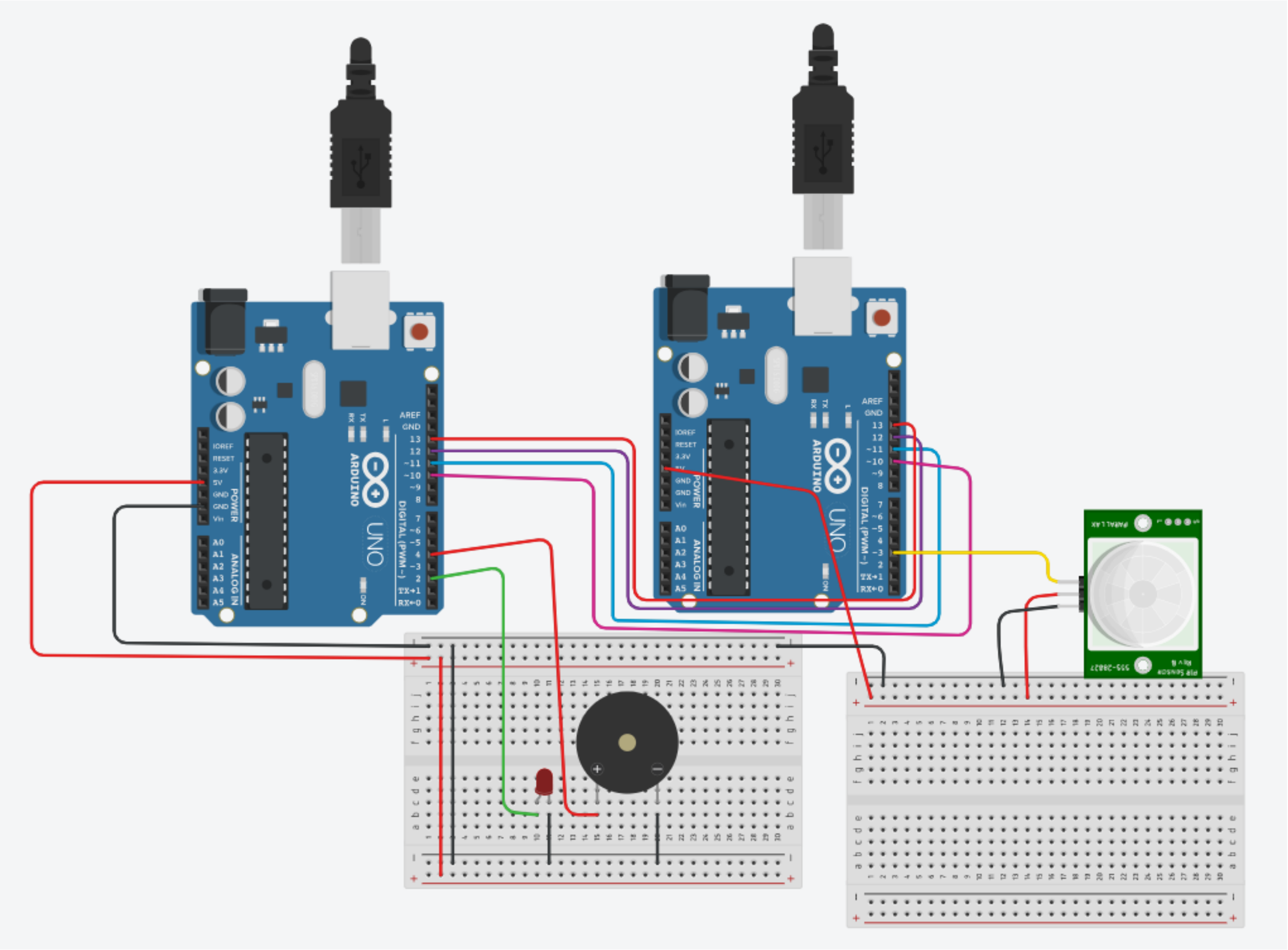
OBSERVATIONS:

We observed that the LED switches ON in the second Arduino only when you press the push button connected in the first Arduino

QUESTION 3:

Simulate a circuit to demonstrate SPI protocol USING any 2 sensors-buzzer and PIR sensor

CIRCUIT DIAGRAM:



CODE:

MASTER CODE:

```
// C++ code
//SPI MASTER
#include<SPI.h>
#include <pins_arduino.h>
int led =2;
int buzzer = 4;
int x=0;
void setup()
{ Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(buzzer, OUTPUT);
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV8);
  digitalWrite(SS,HIGH);

}

void loop()
{
    digitalWrite(SS,LOW);
 byte Masterecieve,Mastersend;
  Mastersend=0;
  Masterecieve= SPI.transfer(Mastersend);
  int v =  Masterecieve;
  Serial.println(v);
  delay(500);

   if(Masterecieve == 1)                   //Logic for setting the LED output depending upon
value received from slave
  {

    digitalWrite(led,HIGH);
    digitalWrite(buzzer,HIGH);
  }
  else
  {

   digitalWrite(led,LOW);
```

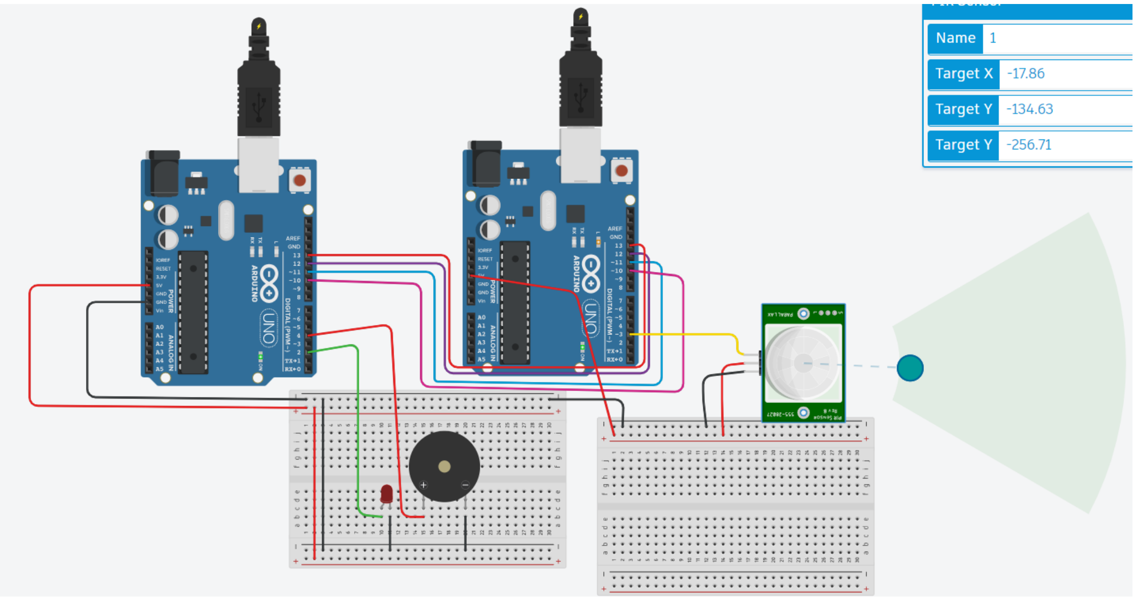
```
    digitalWrite(buzzer,LOW);
  }
digitalWrite(SS, HIGH); // disable Slave Select
  delay(1000);
}
```

SLAVE CODE:

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

volatile boolean received;
volatile byte Slavesend,Slavereceived;

int irs =3;
int irs_val;
int x;
void setup()
{
  Serial.begin(115200);
  pinMode(irs,INPUT);              // Setting pin 2 as INPUT
  pinMode(12,OUTPUT);                  //Sets MISO as OUTPUT (Have to Send data to
Master IN
  SPCR |= _BV(SPE);                    //Turn on SPI in Slave Mode

  received = false;
  SPI.attachInterrupt();                //Interuupt ON is set for SPI commnucation
}

ISR (SPI_STC_vect)                      //Inerrrput routine function
{

  Slavereceived = SPDR;        // Value received from master if store in variable
slavereceived
```

```
  received = true;                    //Sets received as True
}
void loop()
{
  irs_val = digitalRead(irs);
  Serial.println(irs_val);
  delay(500);
  if (irs_val == LOW)                //Logic to set the value of x to send to master
        x=0;
  else
   x=1;
  Slavesend=x;

  SPDR = Slavesend;                          //Sends the x value to master via SPDR
}
```

RESULTS SCREENSHOTS:



Name	1
Target X	-17.86
Target Y	-134.63
Target Y	-256.71

OBSERVATIONS:

We observe that buzzer and LED switches on when it detects an object near PIR sensor