ECGR 4161/5196 LAB 5

GROUP: 29

NAME: Anyaegbu Somtochukwu C. & Ajay Sankar Chundi

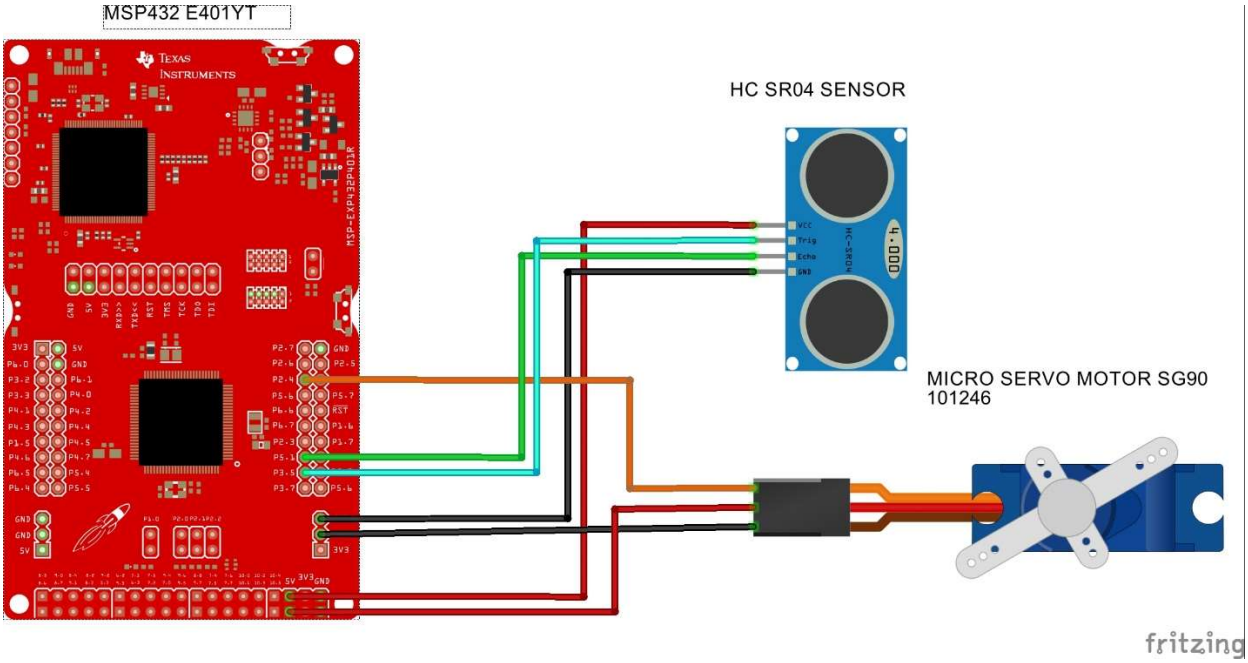
VIDEO LINK:

https://drive.google.com/drive/folders/1hOmAlyJv4Dqr4Dn0_VzIgB__iLi_xvaE?usp=sharing

https://drive.google.com/file/d/1hSBUXXZwSqzrXqqkCHdcsUuycpsmQan4/view?usp=sharing

OBJECTIVE: The main objective is to attach the ultrasonic sensor onto the servo, and then report three ultrasonic measurements at the servo's 0, 90 and 180-degree position.

PICTORIAL CIRCUIT DIAGRAM:



COMMENTARY:

- **Introduction**

  This is a one-part lab which is aimed at enabling students gain a practical understanding of how ultrasound sensors interfaced with a micro servo motor (SG90) and microcontroller board (MSP432) work.

- **Materials Required**
  - MSP432
  - TI-RSLK
  - HC-SR04 UltraSound Sensor
  - Micro Servo Motor SG90
  - F-M and F-F Jumper Wires
  - Energia 1.8.11E23
- **Theory**

  The HC-SR04 works by sending an ultrasonic pulse and then sensing it as it returns. A microcontroller () and the use of the pulse In function is used to measure the time the pulse took to travel out, reflect off an object, and return. Knowing the speed of sound, the distance to the object can be calculated.

  The servo motor uses a feedback to control its motion and direction. It's a closed loop electro-mechanical system.

  A simple code is written to turn the motor to 0 degrees, take 5 readings using the ultrasound sensor and find the median and repeat the process for 90 degrees and 180 degrees turn positions. All readings are monitored using a Serial Monitor via the Energia IDE.

- **Results**

  Learned to use the special micro servo functions on the energia IDE to control the motor. Lab 5 was completed without any issues.

CODE:

```
//*****************************************************************

// ServoExample - Run an inexpensive Servo Motor
```

```
// James Conrad, 2020-06-10
//  Modified by Somto Anyaegbu & Ajay Sankar Chundi 03/19/2021
//****************************************************************
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
              // a maximum of eight servo objects can be created

const int trigPin = 32;              //This is Port Pin 3.5 on the MSP432 Launchpad
const int echoPin = 33;             //This is Port Pin 5.1 on the MSP432 Launchpad



void setup() {  // put your setup code here, to run once:
   pinMode(75, OUTPUT);  //RGB LED - RED_LED
   pinMode(76, OUTPUT);  //RGB LED - GREEN_LED
   pinMode(77, OUTPUT);  //RGB LED - BLUE_LED
   pinMode(trigPin, OUTPUT);  //Trig Signal Pin Set To Output
   pinMode(echoPin, INPUT);   //Echo Signal Pin Set To Output

   myservo.attach(38);  //Attaches the servo on Port 2.4 to the servo object
   myservo.write(0);  // Set to the default position
   Serial.begin(9600); //Set baudrate and initialize Rx & Tx
   delay(5000);

}

void loop() {               // put your main code here, to run repeatedly:
  Serial.println(" ");
```

```
Serial.println("Start Servo-uSound Test");
int pos = 0;              // variable to store the servo position
digitalWrite(75, HIGH);     //turn red LED on
delay(15);                  //waits 15ms for the servo to reach the position
Serial.println("_____________________________________________");
Serial.println("Measuring Distance at 0 deg");
float avgDistA = usound();  //call ultraSound function and print return value
Serial.print("Median Distance at 0 deg = ");
Serial.print(avgDistA);
Serial.println(" cm");
Serial.println("_____________________________________________");


digitalWrite(75, LOW);       //turn red LED off
delay(500);


digitalWrite(76, HIGH);      //turn green LED on
for(pos = 0; pos < 90; pos += 1) { // goes from 0 degrees to 90 degrees
  myservo.write(pos);              // tell servo to go to position in variable 'pos'
  delay(15);                   // waits 15ms for the servo to reach the position
}


  Serial.println("_____________________________________________");
  Serial.println("Measuring Distance at 90 deg");
  float avgDistB = usound(); //call ultraSound function and print return value
  Serial.print("Median Distance at 90 deg = ");
  Serial.print(avgDistB);
  Serial.println(" cm");
```

```
  Serial.println("_____________________________________________");


  digitalWrite(76, LOW);          //turn green LED off
  delay(500);


  digitalWrite(77, HIGH);         //turn blue LED on
  for(pos = 90; pos < 180; pos += 1)  {   // goes from 90 degrees to 180 degrees
    myservo.write(pos);            // tell servo to go to position in variable 'pos'
    delay(15);                   // waits 15ms for the servo to reach the position
}


  Serial.println("_____________________________________________");
  Serial.println("Measuring Distance at 180 deg");
  float avgDistC = usound();        //call uSound function and print return value
  Serial.print("Median Distance at 180 deg = ");
  Serial.print(avgDistC);
  Serial.println(" cm");
  Serial.println("_____________________________________________");
  Serial.println(" ");
  digitalWrite(77, LOW);          //turn blue LED off
  delay(500);                       // waits 500ms


  Serial.println("End Servo-uSound Test.");
  Serial.println("_____________________________________________");


  digitalWrite(76, HIGH);           //turn green LED on
  digitalWrite(75, LOW);            //turn red LED on
```

```
    for(pos = 180; pos>=1; pos-=1)  {   // goes from 180 degrees to 0 degrees
      myservo.write(pos);         // tell servo to go to position in variable 'pos'
      delay(15);               // waits 15ms for the servo to reach the position
  }

  digitalWrite(76, LOW);        //turn green LED off
  digitalWrite(75, LOW);        //turn red LED off
  delay(500);

}



float usound(){     //perception function

  long Midcm;           //declare median val variable
  long centimeters;       //declare centimeter variable

  int pulseLength[5],x,i,j,tmp;   //declare variables

  /* Sort five readings */

  for (i=0; i<5; i++) {                   //Loop for ascending ordering

      digitalWrite(trigPin, LOW);               // send low to get a clean pulse
      delayMicroseconds(2);                     // let it settle
      digitalWrite(trigPin, HIGH);              // send high to trigger device
```

```
    delayMicroseconds(10);                    // let it settle

    pulseLength[x] = pulseIn(echoPin, HIGH);  // measure pulse coming back
    centimeters = pulseLength[x] / 58;        //convert pulse to centimeters

    Serial.print("Distance = ");              //Print distance
    Serial.print(centimeters);
    Serial.println(" cm");
    delay(1000);                              //delay for 1000ms

    for (int j = 0; j < 5; j++) {         //Loop for comparing other values
      if (pulseLength[j] > pulseLength[i]) { //Comparing other array elements
        tmp = pulseLength[i];             //Using temp var for storing last value
        pulseLength[i] = pulseLength[j];  //replacing value
        pulseLength[j] = tmp;             //storing last value
      }
    }
  }

  /* Print middle one */
  Midcm = pulseLength[2] / 58;        //convert pulselength to cm
  return Midcm;                       //Return median

}
```