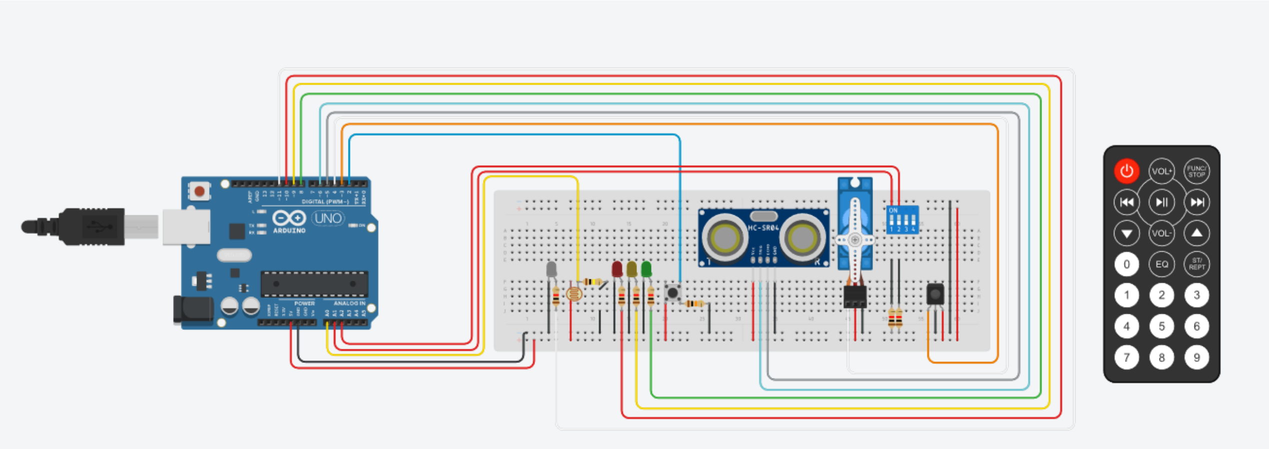
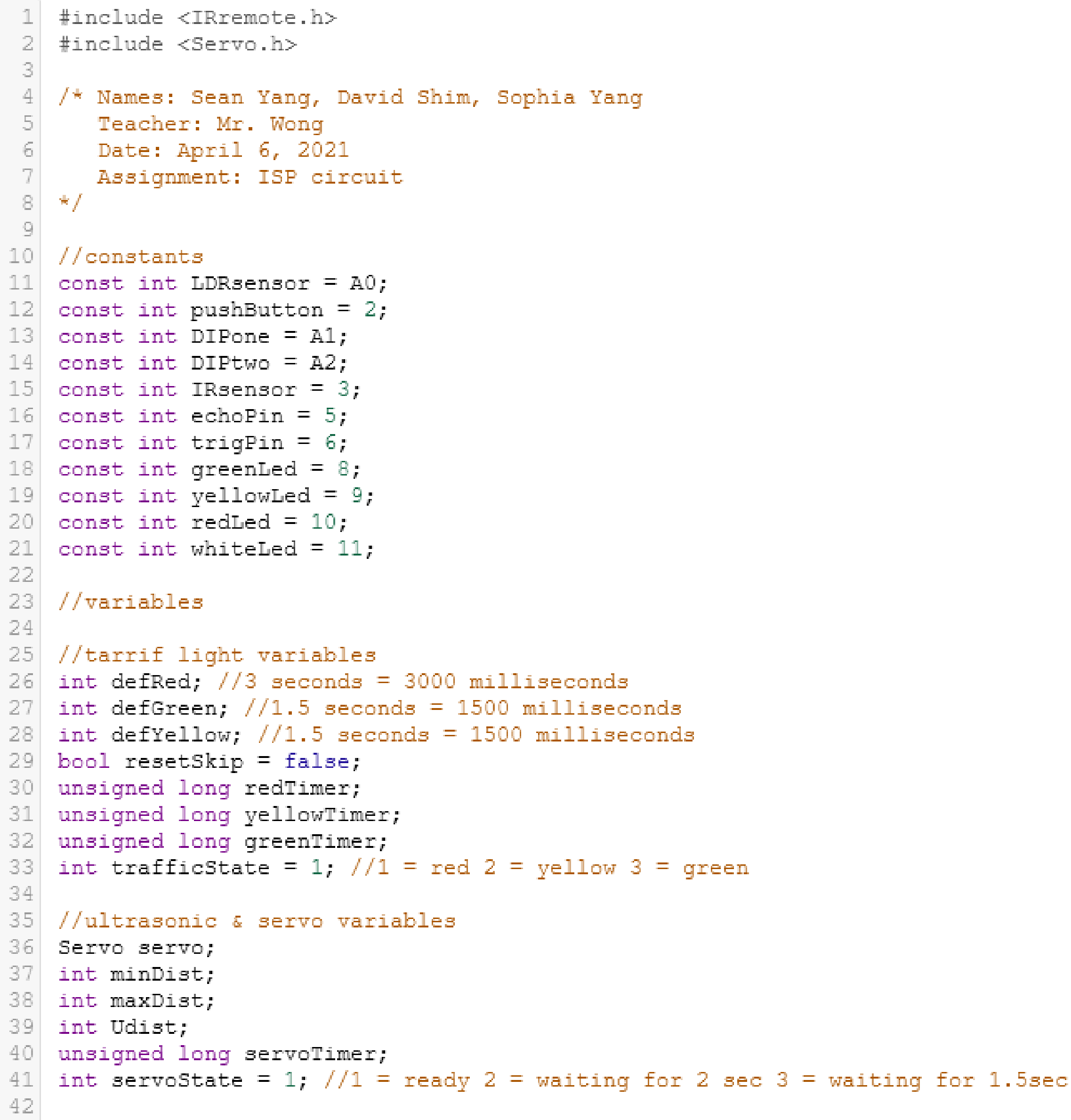
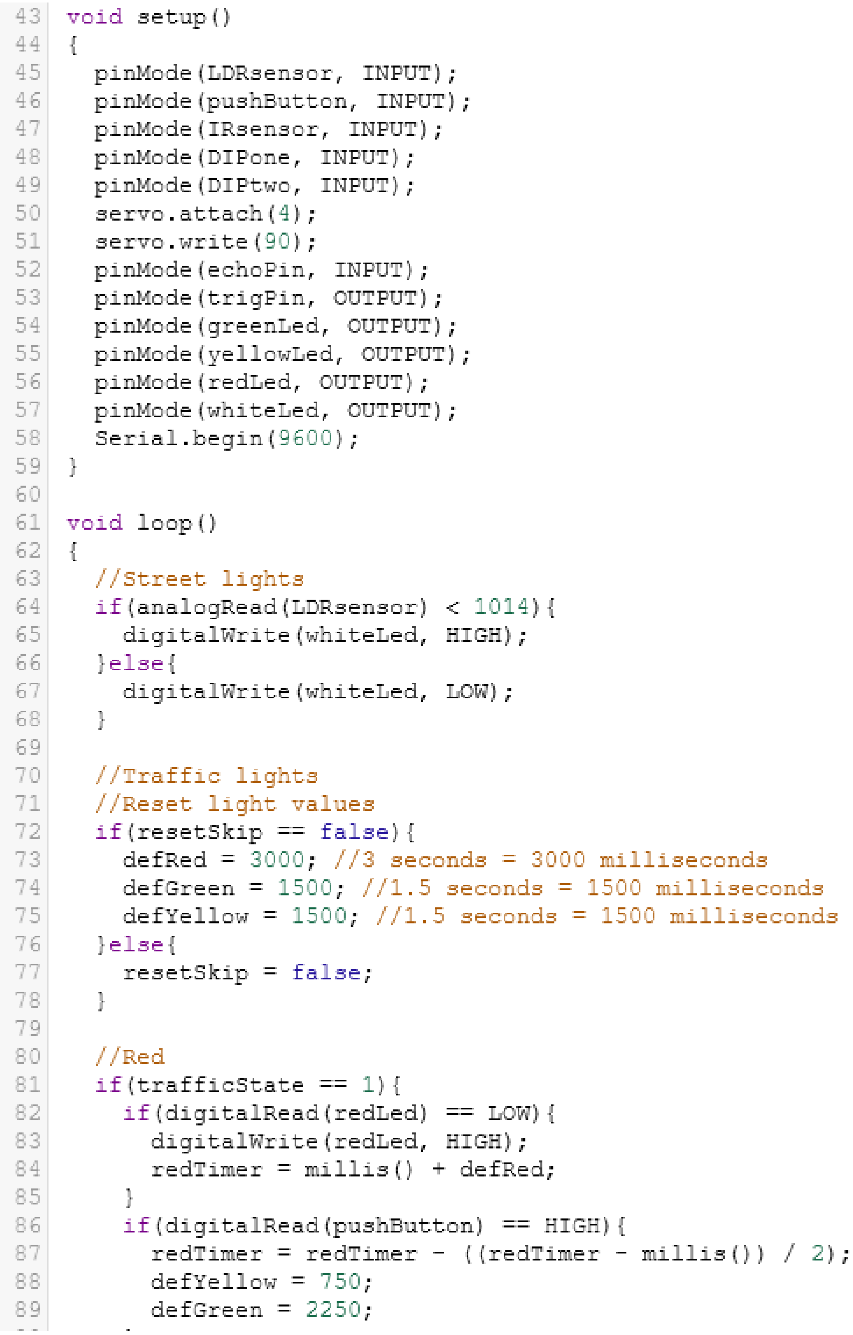
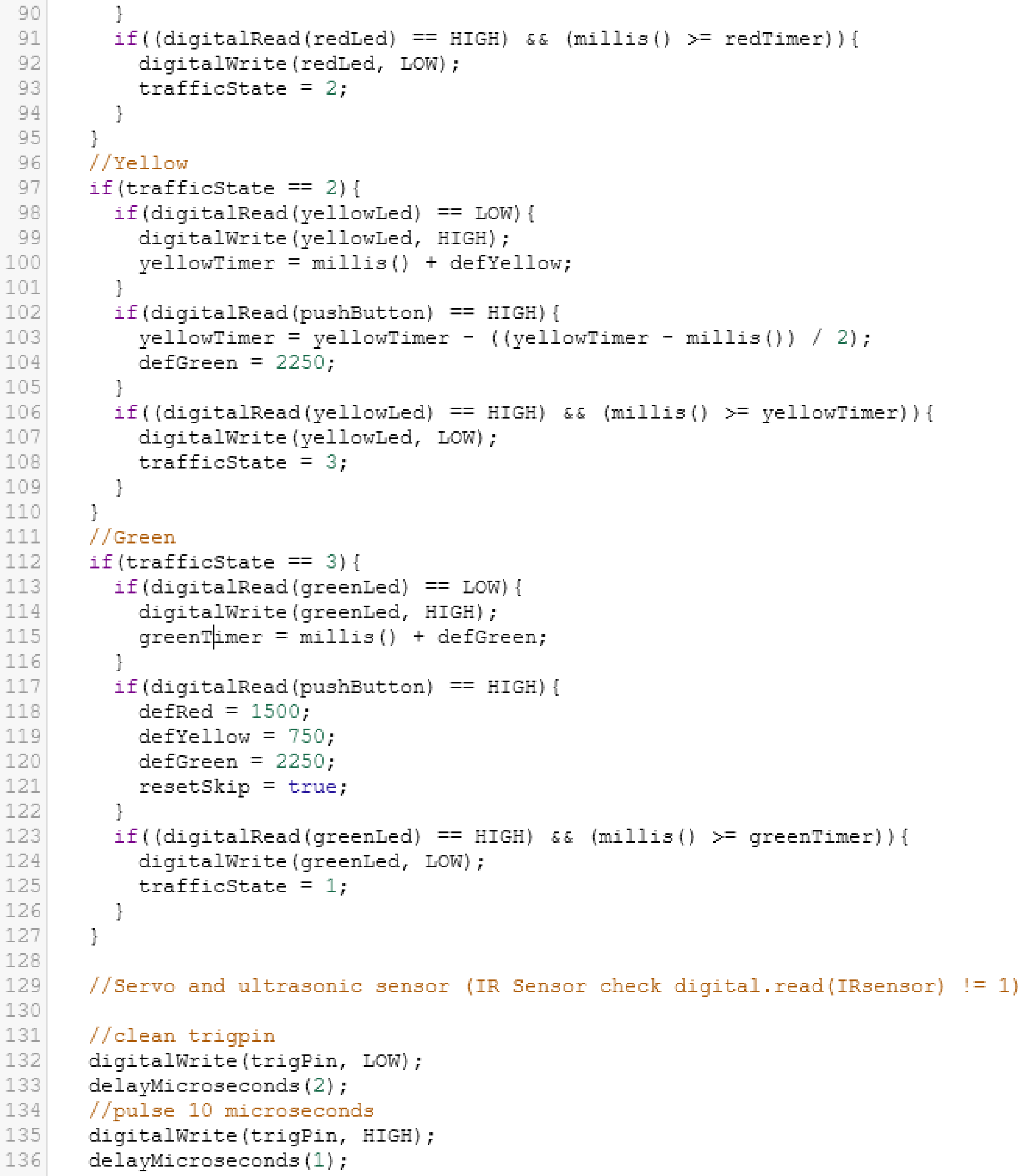




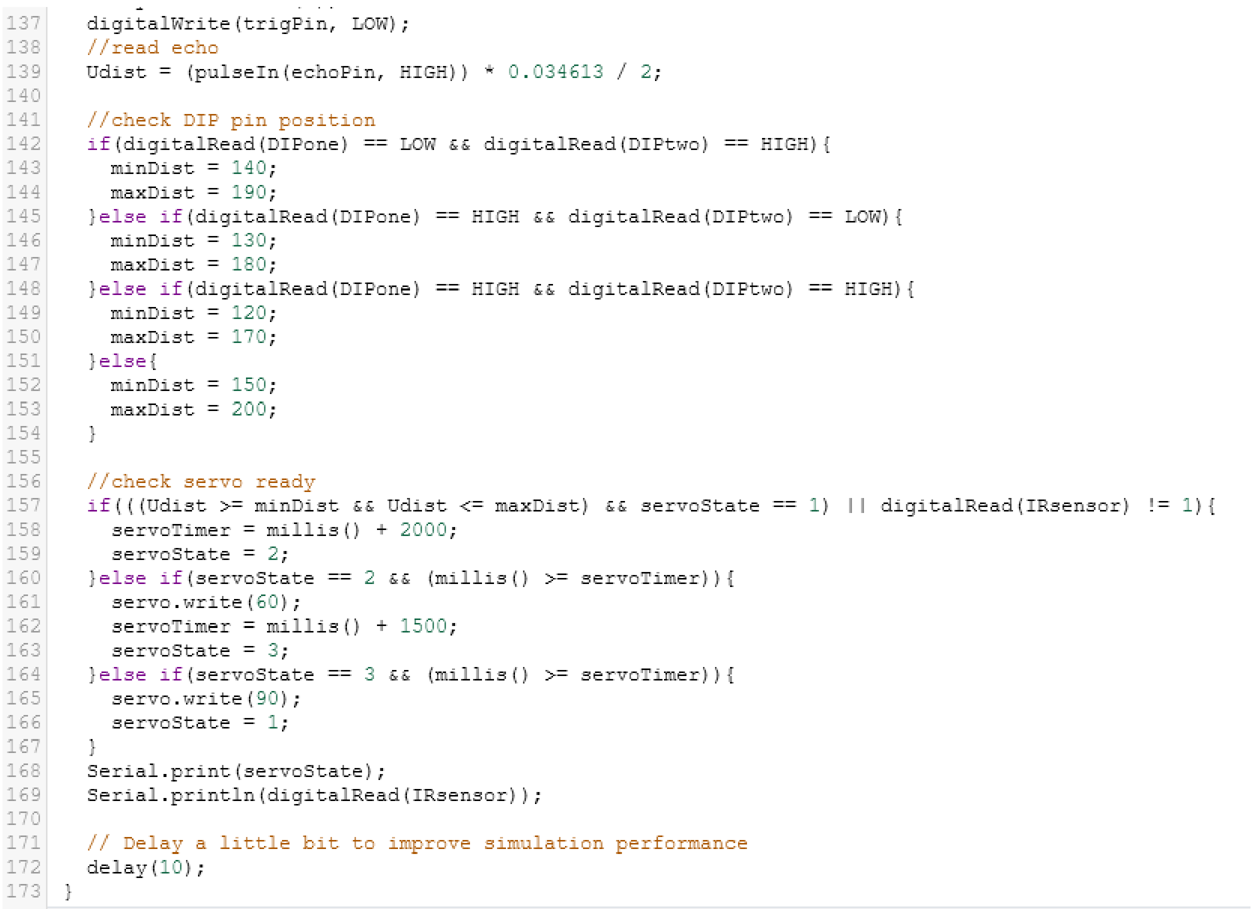
```
#include <IRremote.h>
#include <Servo.h>

/* Names: Sean Yang, David Shim, Sophia Yang
   Teacher: Mr. Wong
   Date: April 6, 2021
   Assignment: ISP circuit
*/

//constants
const int LDRsensor = A0;
const int pushButton = 2;
const int DIPone = A1;
const int DIPtwo = A2;
const int IRsensor = 3;
const int echoPin = 5;
const int trigPin = 6;
const int greenLed = 8;
const int yellowLed = 9;
const int redLed = 10;
const int whiteLed = 11;

//variables

//tarrif light variables
int defRed; //3 seconds = 3000 milliseconds
int defGreen; //1.5 seconds = 1500 milliseconds
int defYellow; //1.5 seconds = 1500 milliseconds
bool resetSkip = false;
unsigned long redTimer;
unsigned long yellowTimer;
unsigned long greenTimer;
int trafficState = 1; //1 = red 2 = yellow 3 = green

//ultrasonic & servo variables
Servo servo;
int minDist;
int maxDist;
int Udist;
unsigned long servoTimer;
int servoState = 1; //1 = ready 2 = waiting for 2 sec 3 = waiting for 1.5sec

```



```
void setup()
{
  pinMode(LDRsensor, INPUT);
  pinMode(pushButton, INPUT);
  pinMode(IRsensor, INPUT);
  pinMode(DIPone, INPUT);
  pinMode(DIPtwo, INPUT);
  servo.attach(4);
  servo.write(90);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(redLed, OUTPUT);
  pinMode(whiteLed, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  //Street lights
  if(analogRead(LDRsensor) < 1014){
    digitalWrite(whiteLed, HIGH);
  }else{
    digitalWrite(whiteLed, LOW);
  }

  //Traffic lights
  //Reset light values
  if(resetSkip == false){
    defRed = 3000; //3 seconds = 3000 milliseconds
    defGreen = 1500; //1.5 seconds = 1500 milliseconds
    defYellow = 1500; //1.5 seconds = 1500 milliseconds
  }else{
    resetSkip = false;
  }

  //Red
  if(trafficState == 1){
    if(digitalRead(redLed) == LOW){
      digitalWrite(redLed, HIGH);
      redTimer = millis() + defRed;
    }
    if(digitalRead(pushButton) == HIGH){
      redTimer = redTimer - ((redTimer - millis()) / 2);
      defYellow = 750;
      defGreen = 2250;
```



```
    }
    if((digitalRead(redLed) == HIGH) && (millis() >= redTimer)){
      digitalWrite(redLed, LOW);
      trafficState = 2;
    }
  }
  //Yellow
  if(trafficState == 2){
    if(digitalRead(yellowLed) == LOW){
      digitalWrite(yellowLed, HIGH);
      yellowTimer = millis() + defYellow;
    }
    if(digitalRead(pushButton) == HIGH){
      yellowTimer = yellowTimer - ((yellowTimer - millis()) / 2);
      defGreen = 2250;
    }
    if((digitalRead(yellowLed) == HIGH) && (millis() >= yellowTimer)){
      digitalWrite(yellowLed, LOW);
      trafficState = 3;
    }
  }
  //Green
  if(trafficState == 3){
    if(digitalRead(greenLed) == LOW){
      digitalWrite(greenLed, HIGH);
      greenTimer = millis() + defGreen;
    }
    if(digitalRead(pushButton) == HIGH){
      defRed = 1500;
      defYellow = 750;
      defGreen = 2250;
      resetSkip = true;
    }
    if((digitalRead(greenLed) == HIGH) && (millis() >= greenTimer)){
      digitalWrite(greenLed, LOW);
      trafficState = 1;
    }
  }

  //Servo and ultrasonic sensor (IR Sensor check digital.read(IRsensor) != 1)

  //clean trigpin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  //pulse 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(1);
```



```
  digitalWrite(trigPin, LOW);
  //read echo
  Udist = (pulseIn(echoPin, HIGH)) * 0.034613 / 2;

  //check DIP pin position
  if(digitalRead(DIPone) == LOW && digitalRead(DIPtwo) == HIGH){
    minDist = 140;
    maxDist = 190;
  }else if(digitalRead(DIPone) == HIGH && digitalRead(DIPtwo) == LOW){
    minDist = 130;
    maxDist = 180;
  }else if(digitalRead(DIPone) == HIGH && digitalRead(DIPtwo) == HIGH){
    minDist = 120;
    maxDist = 170;
  }else{
    minDist = 150;
    maxDist = 200;
  }

  //check servo ready
  if(((Udist >= minDist && Udist <= maxDist) && servoState == 1) || digitalRead(IRsensor) != 1){
    servoTimer = millis() + 2000;
    servoState = 2;
  }else if(servoState == 2 && (millis() >= servoTimer)){
    servo.write(60);
    servoTimer = millis() + 1500;
    servoState = 3;
  }else if(servoState == 3 && (millis() >= servoTimer)){
    servo.write(90);
    servoState = 1;
  }
  Serial.print(servoState);
  Serial.println(digitalRead(IRsensor));

  // Delay a little bit to improve simulation performance
  delay(10);
}
```

Link to circuit: https://www.tinkercad.com/things/drJhdThpKro-dazzling-sango/editel?sharecode=qnJa1HuIs73v_GOb4n_wRqi5N5tRcm-Klw-V-VDgoT0

Traffic Lights:

Goal: red light should stay on for default 3 seconds, yellow light should stay on for default 1.5 seconds, green light should stay on for default 1.5 seconds. Lights should repeat in a loop of red light, yellow light, green light.

Testing: use a stopwatch to record the duration of each light. With no modifiers, red light be on for 3 seconds, then switch to yellow light for 1.5 seconds, then switch to green light for 1.5 seconds, and then go back to red light.

Street Lights:

Goal: if the light sensor doesn’t detect light, the street light should be turned on. If the light sensor does detect light, the street light should be turned off.

Testing: slide the light sensor slider to the right until just after half. White LED should turn on. Slide the light sensor to the left until just after half. White LED should turn off.

Button:

Goal: if the button is pressed when the traffic light that is on is red, the amount of time left until the light changes to yellow is -50%, yellow light on time is also -50%, and the green light on time is +50%. If the button is pressed when the traffic light that is on is yellow, the amount of time left until the light changes to green is -50% and the green light on time is +50%. If the button is pressed while the green light is on, for the next cycle of red, yellow, green, it implements the same changes as when the button is pressed during the red light (red -50%, yellow -50%, green +50%).

Testing: use as stopwatch to measure the on times of each light colour and then compare the times of each button press case against the default times. The time differentials should match up with the differentials described in the goal.

Ultrasonic Sensor/Servo:

Goal: when the ultrasonic sensor detects an object within the ranges of 150cm - 200cm, the servo should turn to 60 degrees after 2 seconds, remain in that position for 1.5 seconds, and then return to the horizontal (90 degrees) position. If only pin2 is HIGH on the DIP switch, the detection ranges change to 140cm - 190cm. If only pin1 is HIGH on the DIP switch, the detection ranges change to 130cm - 180cm. If pin1 and pin2 are HIGH on the DIP switch, the detection ranges change to 120cm - 170cm.

Testing: drag the ultrasonic sensor ball between 150cm - 200cm on the ultrasonic sensor then use a stopwatch to check the timing of the servo changes. Then, repeat the process while checking different pin positions on the DIP switch. The detection distances and servo timings should match up with the goal.

IR sensor:

Goal: when the IR sensor detects a button press, the servo should conduct its movement as described in the goal of the ultrasonic sensor/servo while disregarding the state of the ultrasonic sensor.

Testing: drag the ultrasonic sensor ball outside of its current range of detection and press a button on the IR remote control. The servo should still move to 60 degrees after 2 seconds, remain in the position for 1.5 seconds, then return back to the 90 degree position.