

Pranav Bhounsule

Arduino sensors and servo

1. If-else statement and DigitalRead()

Equipment: 10K ohm resistor, 1 Button switch, 1 bread board

<https://www.arduino.cc/en/Tutorial/Button>

2. Ultrasonic sensor

Equipment: 1 ultra sonic sensor, 1 breadboard

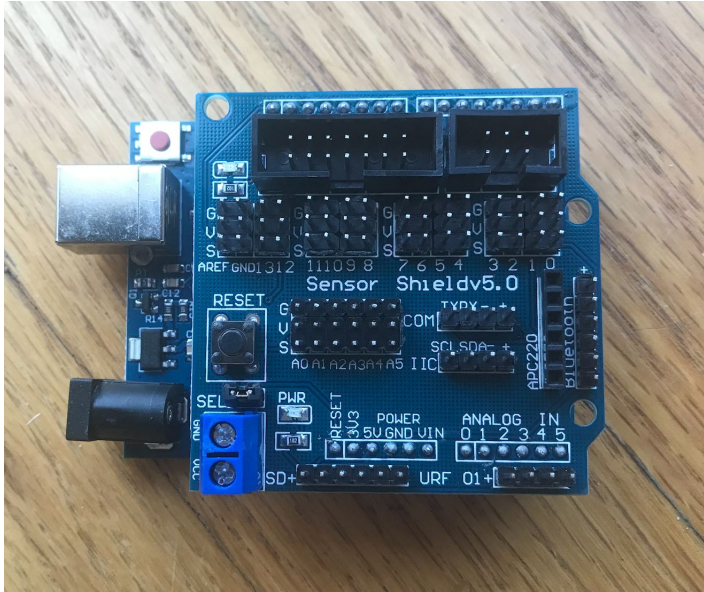
Caution: Ensure that you insert the ultrasonic sensor pins normal to the breadboard else you might break the pins.

<https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

What is the minimum and maximum distance detected by the sensor?

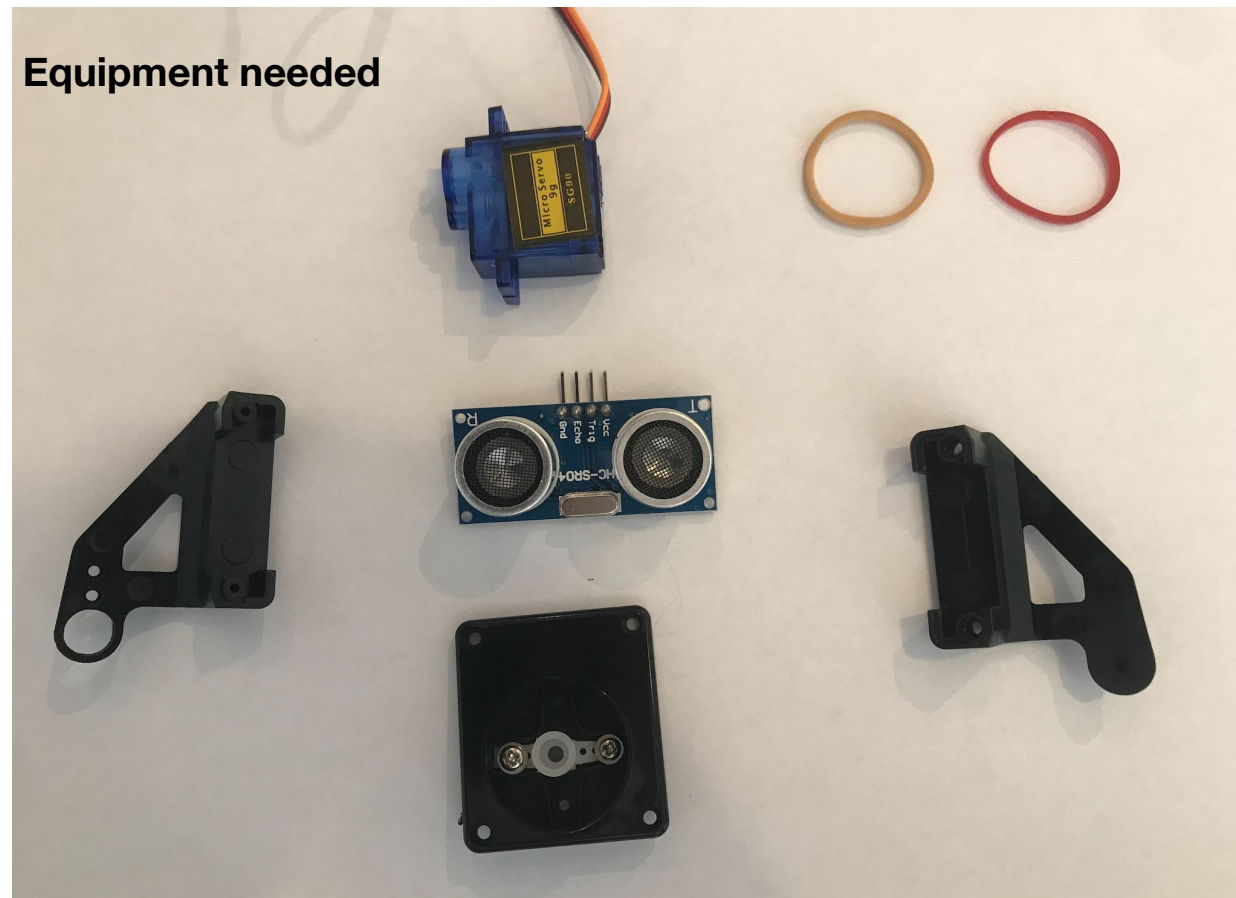
3. Arduino Sensor Shield v5.0 with servo and ultrasonic sensor running simultaneously.

An Arduino sensor shield goes on top of the Arduino and helps to increase the number of pins available and additional connectors not possible with just the Arduino board.

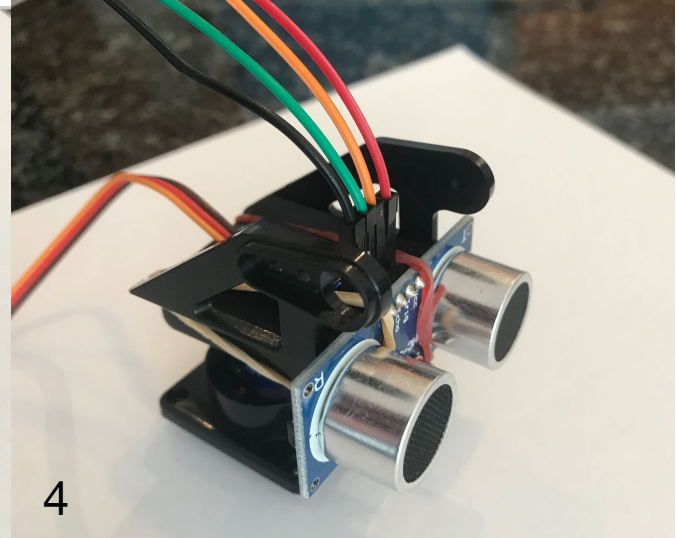
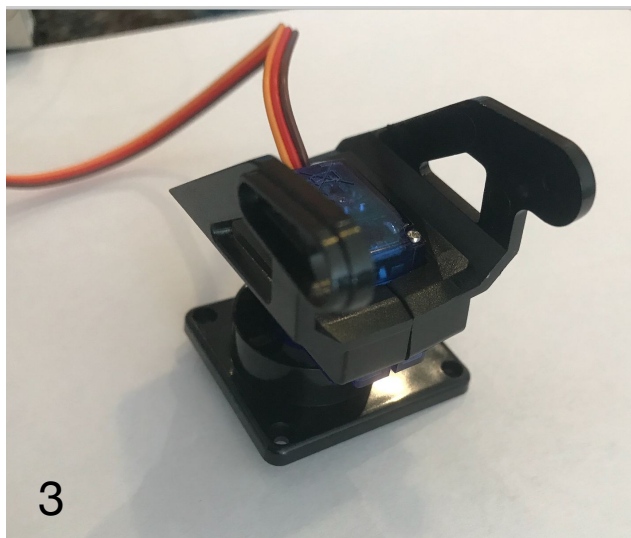
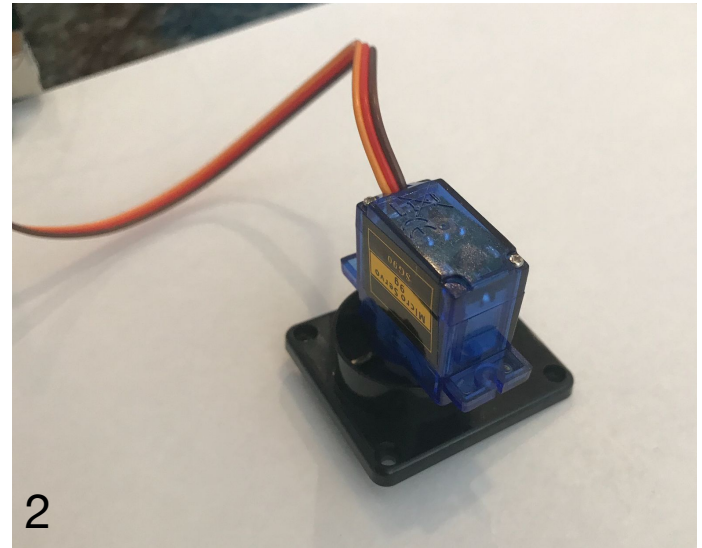
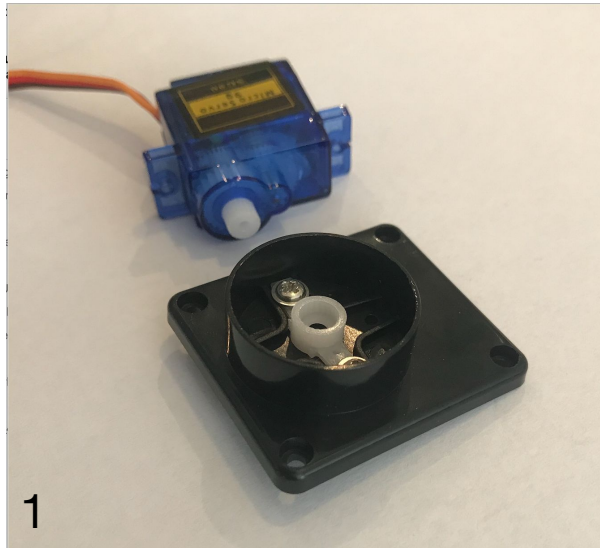


Here we will mount the ultrasonic sensor and servo on the chassis support system and write a code to pan the ultrasonic sensor controlled by servo and relay the distance information.

Equipment: 1 servo motor, 1 ultrasonic sensor, and 1 Arduino sensor shield (note we will not use the breadboard anymore), black connectors to mount the sensor and servo motor, 2 rubber bands. See photo below



A. Next we connect the mechanical elements together as shown in the Figure below. Also, don't forget to attach screw (its inside the bag that had the servo) to fasten the motor the base (after 2) and then use 2 screws with sharp ends to attach the connectors together (after 3). Finally, use 2 rubber bands to attach the ultrasonic sensor with its pin up as shown in 4.

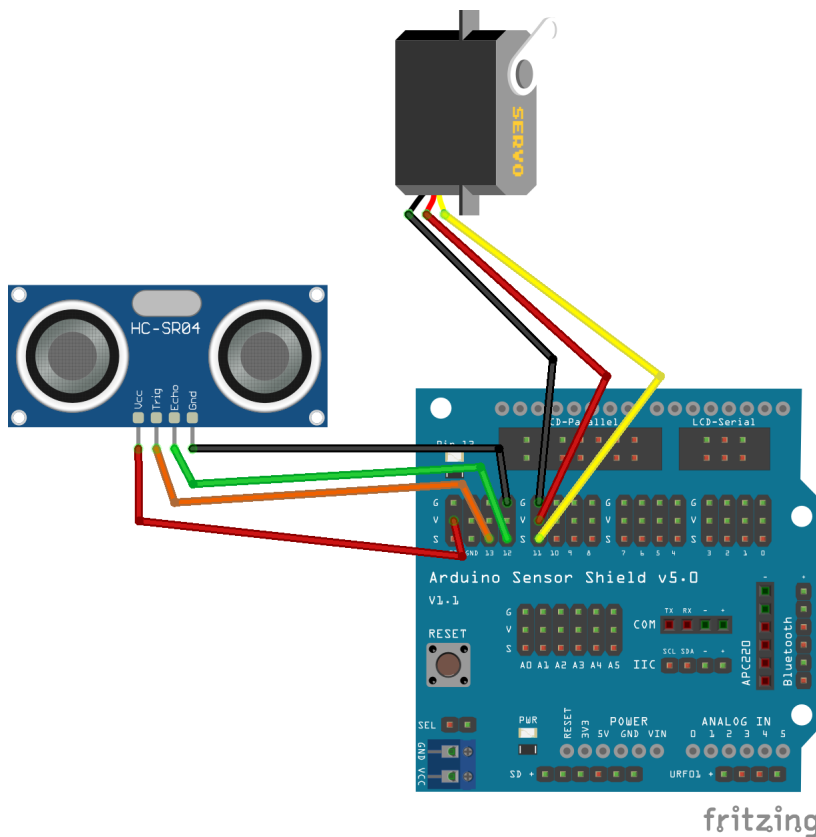


B. Next is wiring

Place the arduino sensor shield on top of the arduino by carefully aligning the pins. Note that the Analog IN pins on the shield should line up with the analog IN pins on the arduino. Be careful in the placement (pin enters normal to the board surface) else the pins can get bent and break.

Next, wire the Ultrasonic sensor and servo motor based on the wiring diagram shown below.

NOTE: Orange is to +5V, Brown is to GND, and Yellow or Light orange is to Digital input pin.



C. Finally, let's program. Open the Arduino IDE, and copy-paste the code below, save as sensor_servo.ino, connect the arduino to the USB and upload the code to the Arduino

```
#include <Servo.h>

// servo stuff
int servoPin = 11;
int pos = 90; // variable to store the servo position
int dpos = 1; // increments
int servoDir = 1; //increment
Servo myservo; // create servo object to control a servo

//ultrasonic stuff
const int trigPin = 13;
const int echoPin = 12;
long duration;
int distance;

void check_obstacles()
{
    pos+=servoDir*dpos;
    myservo.write(pos);
    delay(15);
    if (pos>=180)
    {
        servoDir = -1;
    }
    else if(pos<=0)
    {
        servoDir = +1;
    }
    distance = read_distance();
    Serial.print("Distance: ");
    Serial.println(distance);
}

long read_distance()
{
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
```

```

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration*0.034/2;

return distance; //returns the distance
}

void setup() {
  //servo stuff
  myservo.attach(servoPin);
  myservo.write(pos); //might need calibration

  //ultrasonic stuff
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  Serial.begin(9600); // Starts the serial communication
}

void loop() {
  check_obstacles();
}

```