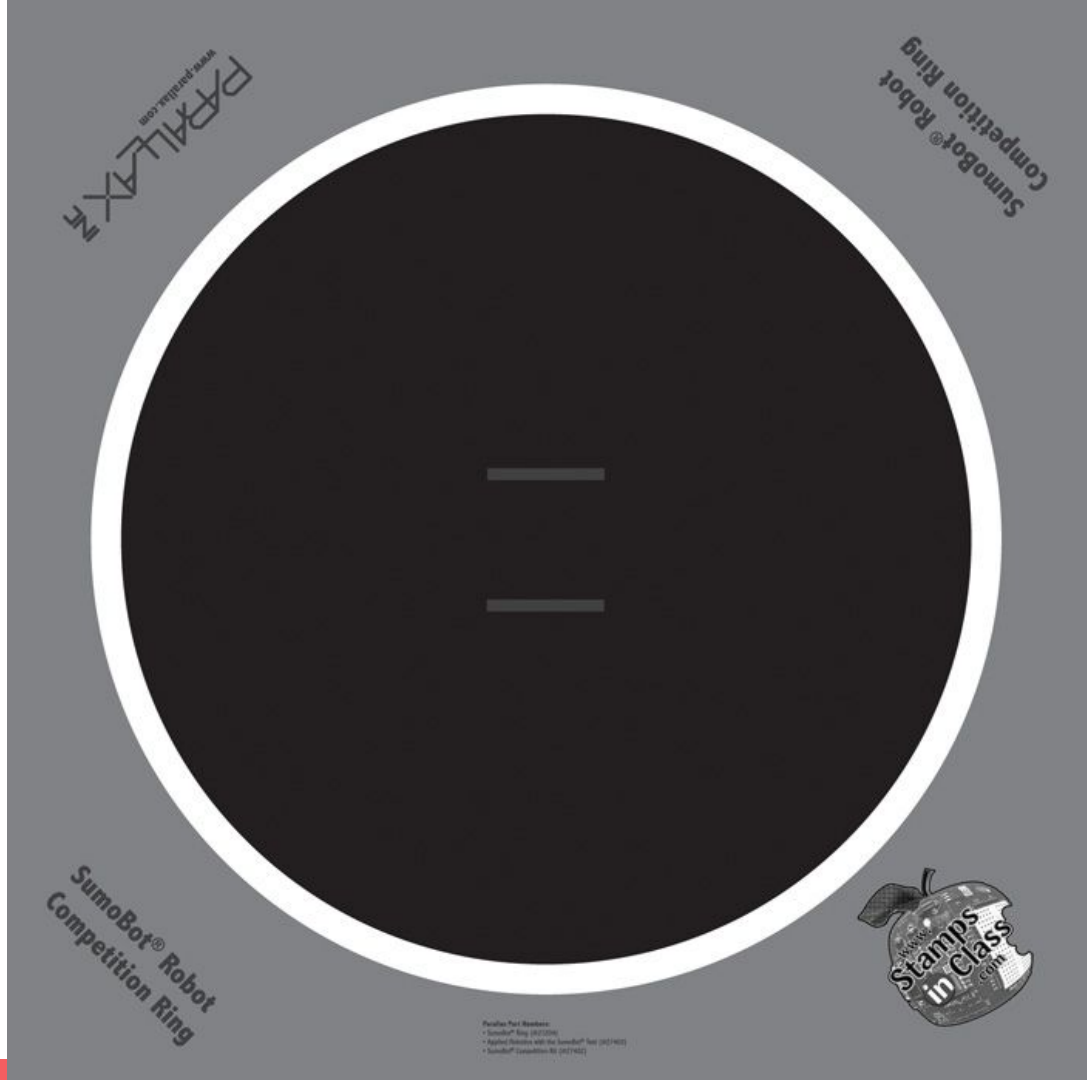
Robotics Workshop Day 3

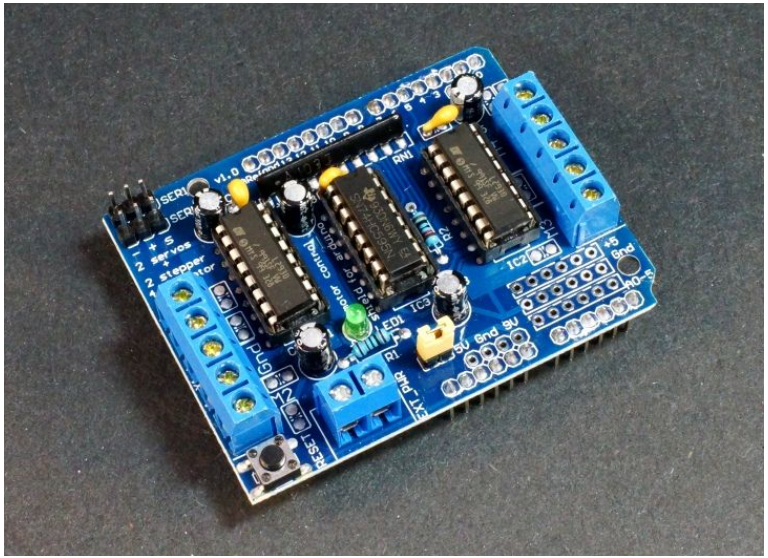
Katherine Kee Wan Ting
MAE Robotics Club

Sumo Challenge



Controlling Speed and Direction (Motor 4)

The following sketch will give you complete understanding on how to control speed and spinning direction of a DC motor with L293D motor driver shield and can serve as the basis for more practical experiments and projects.



E.g. Controlling speed of Motor Number 4

```
#include <AFMotor.h>
AF_DCMotor motor_4(4);
void setup() {
        //Set initial speed of the motor & stop
        motor_4.setSpeed(200);
        motor_4.run(RELEASE);
}

void loop() {
        uint8_t i;
        motor_4.run(FORWARD);
        // Choose motor direction
//Change “FORWARD” to “BACKWARD” to reverse

        // Accelerate from zero to maximum speed
        for (i=0; i<255; i++) {
                motor_4.setSpeed(i);
                delay(10);
        }

        // Decelerate from maximum speed to zero
        for (i=255; i!=0; i--) {
                motor_4.setSpeed(i);
                delay(10);
        }

        motor_4.run(RELEASE); // Now turn off motor
        delay(1000);
}
```

Changing Direction

```
motor_4.run(FORWARD);

motor_4.setSpeed(255);

motor_3.run(BACKWARD);

motor_4.setSpeed(255);

//turn on the spot
```

Ultrasonic Sensor: Obstacle Avoidance Part 1

Motor Shield Library:
https://github.com/adafruit/Adafruit-Motor-Shield-library

New Ping Library:
https://bitbucket.org/teckel12/arduino-new-ping/wiki/Home#!download-install

```
#include <AFMotor.h>
#include <NewPing.h>
#include <Servo.h>
```

Next, we declare the pins to which or ultrasonic sensor is connected to, and some variables which will be used to store info as the code runs then we set the speed of the motors. You can set any value up to 255.

```
#define TRIG_PIN A4
#define ECHO_PIN A5
#define MAX_DISTANCE 200
#define MAX_SPEED 190 // sets speed of DC  motors
#define MAX_SPEED_OFFSET 20
```

Next, we Initialize the servo motor by creating an object of the servo library. We Also initialize the Motors to which the wheels are connected using the AF library.

```
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(3, MOTOR12_1KHZ);

Servo myservo;
```

Ultrasonic Sensor: Obstacle Avoidance Part 2

We then move into the void setup() function where we initialize the servo motor and we set it to “look” straight. In my case, this was at 115 degrees angle, but you might need to do some maths or trial and error to get the degrees which signify the ultrasonic sensor is forward faced. Next, we read the distance a few times in order to get a valid distance measurement.

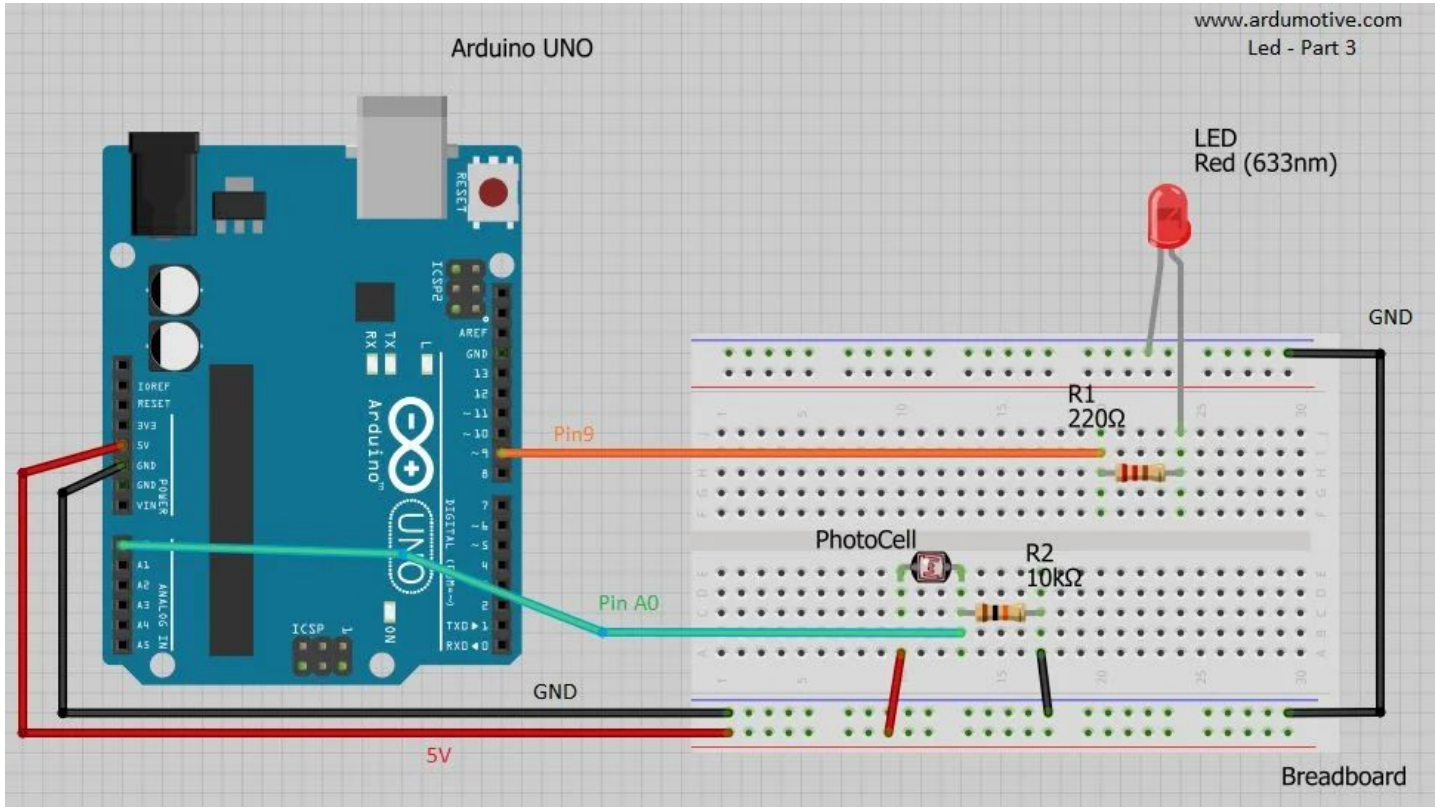
```
void setup() {
        myservo.attach(9);
        myservo.write(115);
        delay(2000);
        distance = readPing();
        delay(100);
        distance = readPing();
        delay(100);
        distance = readPing();
        delay(100);
        distance = readPing();
        delay(100);
}
```

Now we go to the loop function which executes every 40 ms. If the distance we measured is less than or equal to 15 cm, we stop the motors, reverse for 300ms, stop, look right and left and measure the distance in each direction. If the distance in one direction is greater than the other we turn the robot to the direction with the farthest/greatest distance and instruct it to move forward.

```
void loop() {
        int distanceR = 0;
        int distanceL =  0;
        delay(40);
        if(distance<=15) {
                moveStop();
                delay(100); //0.1sec
                moveBackward();
                delay(300);
                moveStop();
                delay(200);
                distanceR = lookRight();
                delay(200);
```

```
if(distanceR>=distanceL) {
        turnRight();
        moveStop();
}else {
        turnLeft();
        moveStop();
}

}else{ moveForward(); }

 distance = readPing();

}
```

Photoresistor Project Diagram

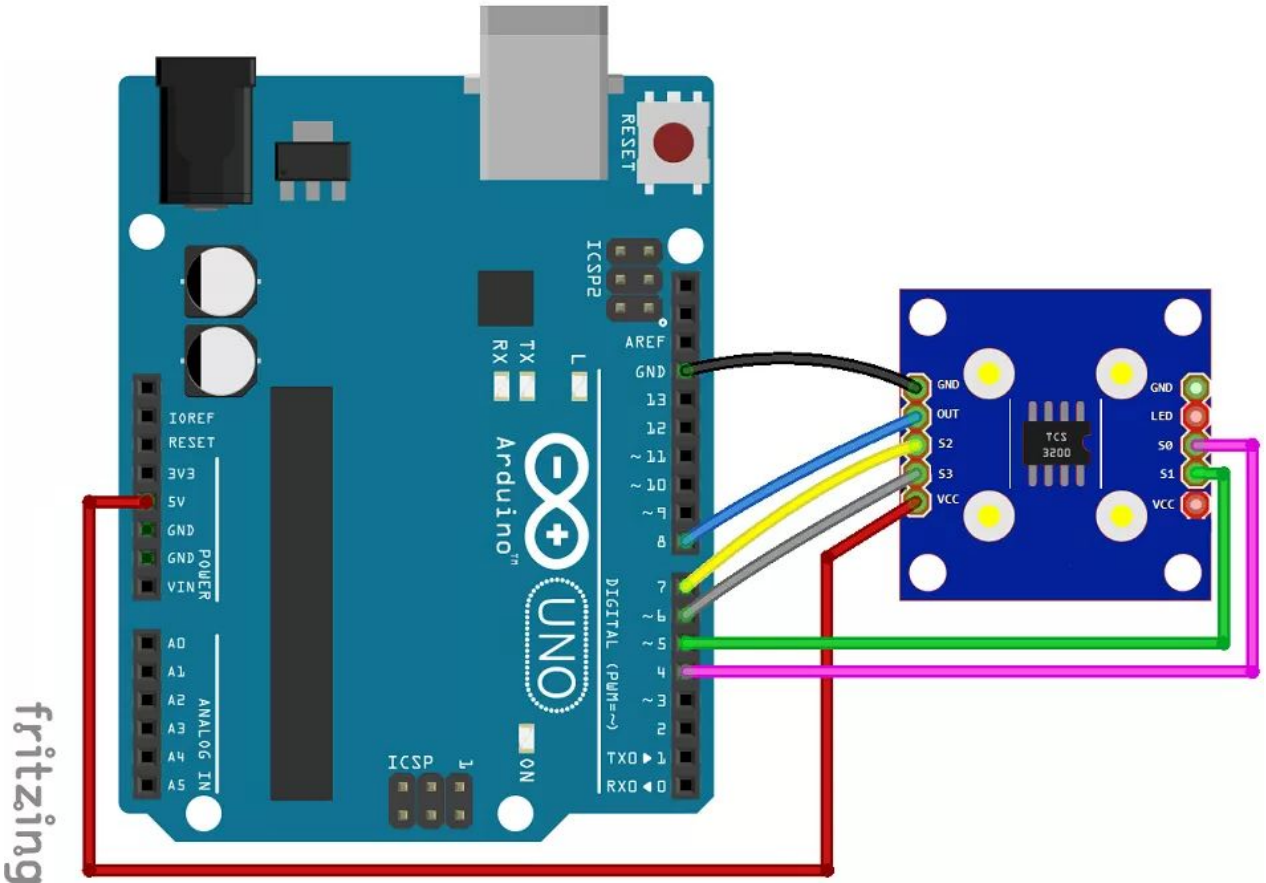


Photoresistor Code

```
//Constants
const int pResistor = A0; // Photoresistor at
Arduino analog pin A0

const int ledPin=9;      // Led pin at Arduino pin 9
int value;
// Store value from photoresistor (0-1023)

void setup(){
        pinMode(ledPin, OUTPUT);
        pinMode(pResistor, INPUT);
}
```

```
void loop(){
        value = analogRead(pResistor);
        //You can change value "25"

        if (value > 500){
        digitalWrite(ledPin, LOW);  //Turn led off
        }
         else{
        digitalWrite(ledPin, HIGH); //Turn led on
        }

        delay(500); //Small delay
}
```

TCS230/TCS3200 Color Sensor with Arduino



- **S0**: digital pin 4
- **S1**: digital pin 5
- **VCC**: 5V
- **S3**: digital pin 6
- **S4**: digital pin 7
- **OUT**: digital pin 8

What the code does - Colour Sensor

1. Reading and displaying the output frequency on the serial monitor. In this part you need to write down the frequency values when you place different colors in front of the sensor.

2. Distinguish between different colors. In this section you'll insert the frequency values picked previously on your code, so that your sensor can distinguish between different colors. We'll detect red, green and blue colors.

Colour Sensor Code part 1

```
// TCS230 or TCS3200 pins wiring to Arduino
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

// Stores frequency read by the photodiodes
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
```

```
void setup() {
      // Setting the outputs
      pinMode(S0, OUTPUT);
      pinMode(S1, OUTPUT);
      pinMode(S2, OUTPUT);
      pinMode(S3, OUTPUT);

  // Setting the sensorOut as an input
      pinMode(sensorOut, INPUT);

  // Setting frequency scaling to 20%
      digitalWrite(S0,HIGH);
      digitalWrite(S1,LOW);

  // Begins serial communication
      Serial.begin(9600);

}
```

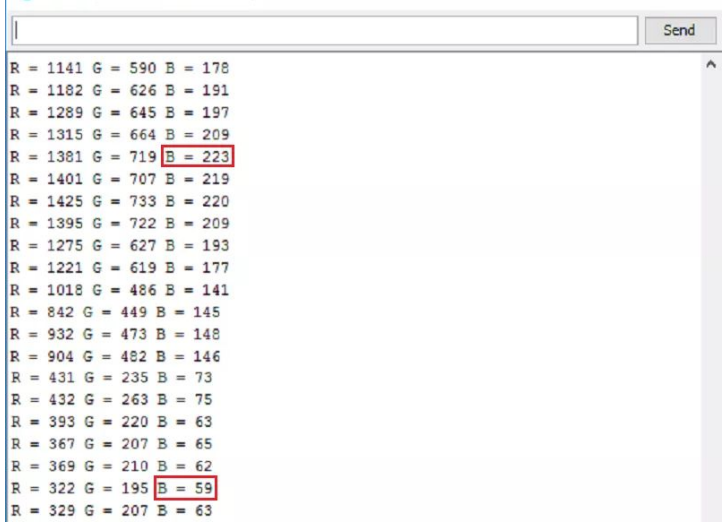
Code Part 2

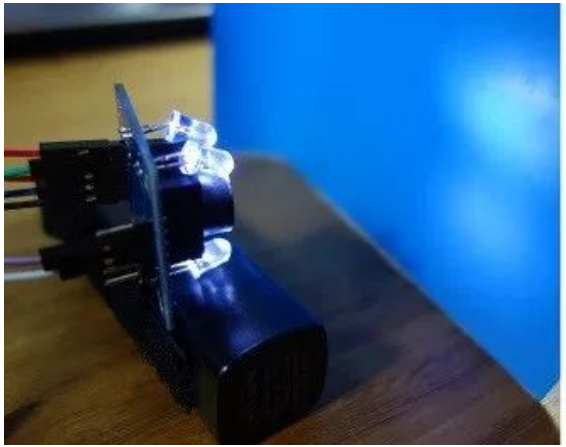
```
void loop() {
// Setting RED (R) filtered photodiodes to be read
      digitalWrite(S2,LOW);
      digitalWrite(S3,LOW);

// Reading the output frequency
      redFrequency = pulseIn(sensorOut, LOW);

 // Printing the RED (R) value
      Serial.print("R = ");
      Serial.print(redFrequency);
      delay(100);

// Setting GREEN (G) filtered photodiodes to be read
      digitalWrite(S2,HIGH);
      digitalWrite(S3,HIGH);

// Reading the output frequency
      greenFrequency = pulseIn(sensorOut, LOW);

// Printing the GREEN (G) value
      Serial.print(" G = ");
      Serial.print(greenFrequency);
      delay(100);

// Setting BLUE (B) filtered photodiodes to be read
      digitalWrite(S2,LOW);
      digitalWrite(S3,HIGH);

// Reading the output frequency
      blueFrequency = pulseIn(sensorOut, LOW);

// Printing the BLUE (B) value
      Serial.print(" B = ");
      Serial.println(blueFrequency);
      delay(100);

 }
```

Try Sensing Colours

Open the serial monitor at a baud rate of 9600.

Place a blue object in front of the sensor at different distances. You should save two measurements: when the object is placed far from the sensor and when the object is close to it.





Colour Tutorial 2 - Printing out colour

Distinguish between different colors

This next sketch maps the frequency values to RGB values (that are between 0 and 255).

In the previous step when we have maximum blue we obtained a frequency of 59 and when we have blue at a higher distance we obtained 223.

So, 59 in frequency corresponds to 255 (in RGB) and 223 in frequency to 0 (in RGB). We'll do this with the Arduino *map()* function. In the *map()*function you need to replace XX parameters with your own values.

Print Colour Code 1

```
// TCS230 or TCS3200 pins wiring to Arduino

#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

// Stores frequency read by the photodiodes
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;

// Stores the red. green and blue colors
int redColor = 0;
int greenColor = 0;
int blueColor = 0;
```

```
void setup() {
      pinMode(S0, OUTPUT);
      pinMode(S1, OUTPUT);
      pinMode(S2, OUTPUT);
      pinMode(S3, OUTPUT);

      pinMode(sensorOut, INPUT);

// Setting frequency scaling to 20%
      digitalWrite(S0,HIGH);
      digitalWrite(S1,LOW);

  // Begins serial communication
      Serial.begin(9600);

}
```

Print Colour Code 2

```
void loop() {
// Setting RED (R) filtered photodiodes to be read
      digitalWrite(S2,LOW);
      digitalWrite(S3,LOW);

// Reading the output frequencY
      redFrequency = pulseIn(sensorOut, LOW);

// Remaping the value of the RED (R) frequency from 0 to 255
// You must replace with your own values. Here's an example:
// redColor = map(redFrequency, 70, 120, 255,0);
      redColor = map(redFrequency, XX, XX, 255,0);

// Printing the RED (R) value
      Serial.print("R = ");
      Serial.print(redColor);
      delay(100);

// Setting GREEN (G) filtered photodiodes to be read
      digitalWrite(S2,HIGH);
      digitalWrite(S3,HIGH);

// Reading the output frequency
      greenFrequency = pulseIn(sensorOut, LOW);

// Remap Green Now
// greenColor = map(greenFrequency, 100, 199, 255, 0);
      greenColor = map(greenFrequency, XX, XX, 255, 0);

// Printing the GREEN (G) value
      Serial.print(" G = ");
      Serial.print(greenColor)
      delay(100);
```

Print Colour Code 3

```
// Setting BLUE (B) filtered photodiodes to be read
      digitalWrite(S2,LOW);
      digitalWrite(S3,HIGH);

 // Reading the output frequency
      blueFrequency = pulseIn(sensorOut, LOW);
// Remaping the value of the BLUE (B) frequency from 0 to 255
// Example: blueColor = map(blueFrequency, 38, 84, 255, 0);
      blueColor = map(blueFrequency, XX, XX, 255, 0);

  // Printing the BLUE (B) value
      Serial.print(" B = ");
      Serial.print(blueColor);
      delay(100);
```

```
// Checks the current detected color and prints a
message in the serial monitor

if(redColor > greenColor && redColor > blueColor){
      Serial.println(" - RED detected!");
}
if(greenColor > redColor && greenColor > blueColor){
      Serial.println(" - GREEN detected!");
}
if(blueColor > redColor && blueColor > greenColor){
      Serial.println(" - BLUE detected!");
}

}
```

Testing Code

To distinguish between different colors we have three conditions:

- When the R is the maximum value (in RGB parameters) we know we have a red object
- When G is the maximum value, we know we have a green object
- When B is the maximum value, we know we have a blue object

BLUE

GREEN

RED

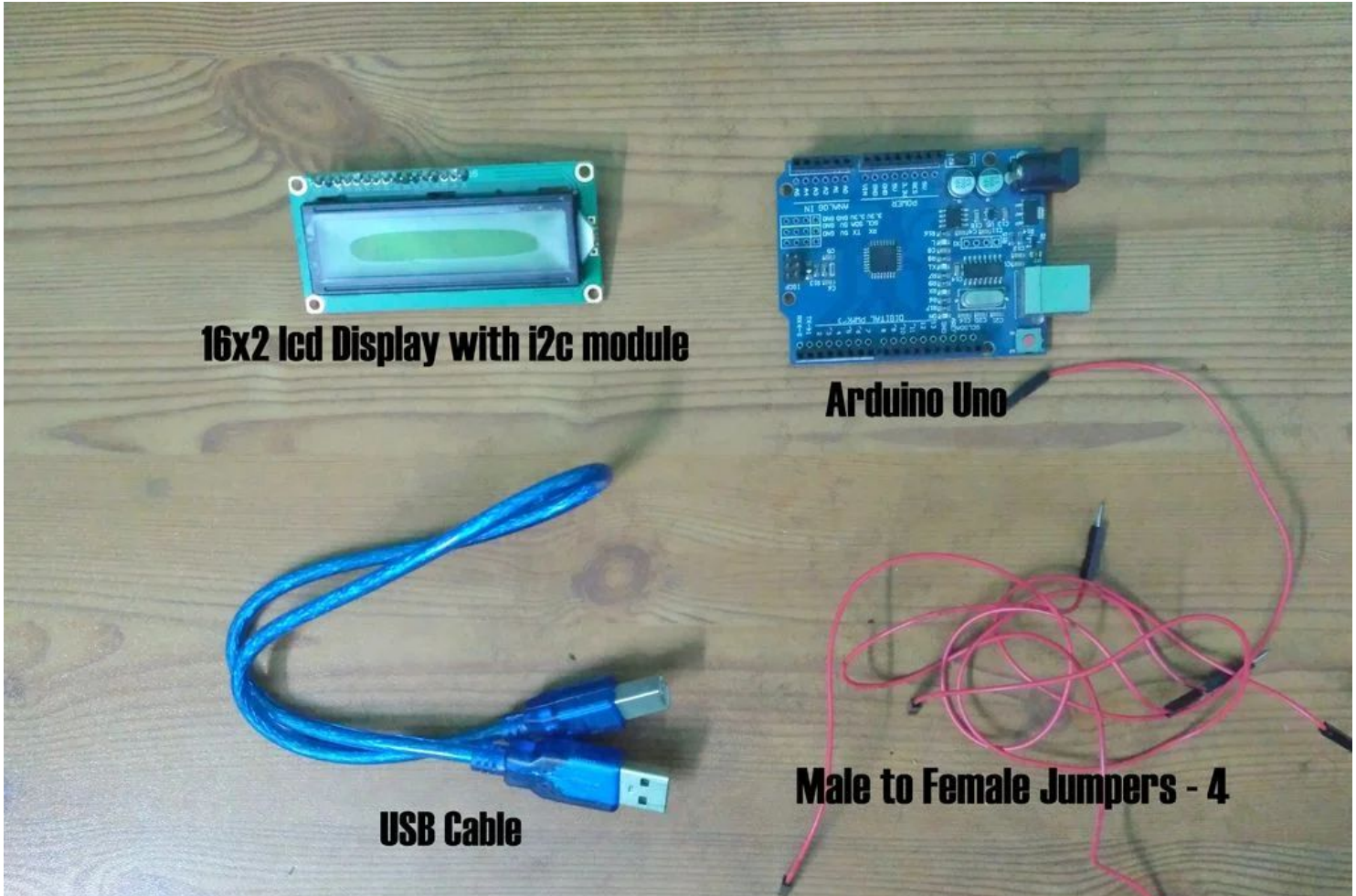
Now, place something in front of the sensor. It should print in your serial monitor the color detected: red, green or blue.

Items in the box

Joystick Shield Module for Arduino	HR0214-5 Joystick Shield Module for Arduino
Color sensor TCS230 TCS3200 Color Recognition Sensor	HR0214-59 Color sensor TCS230 TCS3200 Color Sensor
Infrared Obstacle Avoidance Sensor	HR0062 Infrared Obstacle Avoidance Sensor
Infrared IR Transmitter and Receiver Module	HR0097-1 Infrared IR Receiver Module
Tilt Sensor	HR0214-10 SW520D Ball switch Tilt sensor
Diffused 3mm Red LED - 10pcs	E CLL30001R 3mm Red LED - 10pcs
5mm Triple Output LED RGB - Common Cathode (5Pcs)	E CLC50350C 5mm LED RGB - Common Cathode (5Pcs)
I2C 16x2 Blue Backlight LCD Display Module For Arduino	HR0079 1602A IIC with blue light
5MM GL5528 Light Dependent Resistor Photoresistor LDR	HR0282 5MM GL5528 Photoresistor LDR
Trimpot 10K with Knob	HR0595 Trimpot 10K with Knob
High Speed Micro Continuous Rotation Servo	FT FS90R Micro Continuous Rotation Servo
Thermistor 10K - 1pc	SF00250Thermistor 10K
Robot Car Chassis Kit w/ Encoder Disc	HR0238 Robot Car Chassis Kit
0.96" White I2C OLED Display Module For Arduino	HR0088 0.96 White I2C OLED Display Module For Arduino
MAX7219 Dot Matrix Module	HR0122 MAX7219 Dot Matrix Module

Feel Free to Google search tutorials of the name of these items to find out more!

LCD Screen


16x2 lcd Display with i2c module
Arduino Uno
USB Cable
Male to Female Jumpers - 4

Scope for Today

LCD Work
-optional: test for I2C address
- Change Mega pin 20 SDA and 21 SCL
- Adjust the backlight contrast

Testing the Address (LCD)

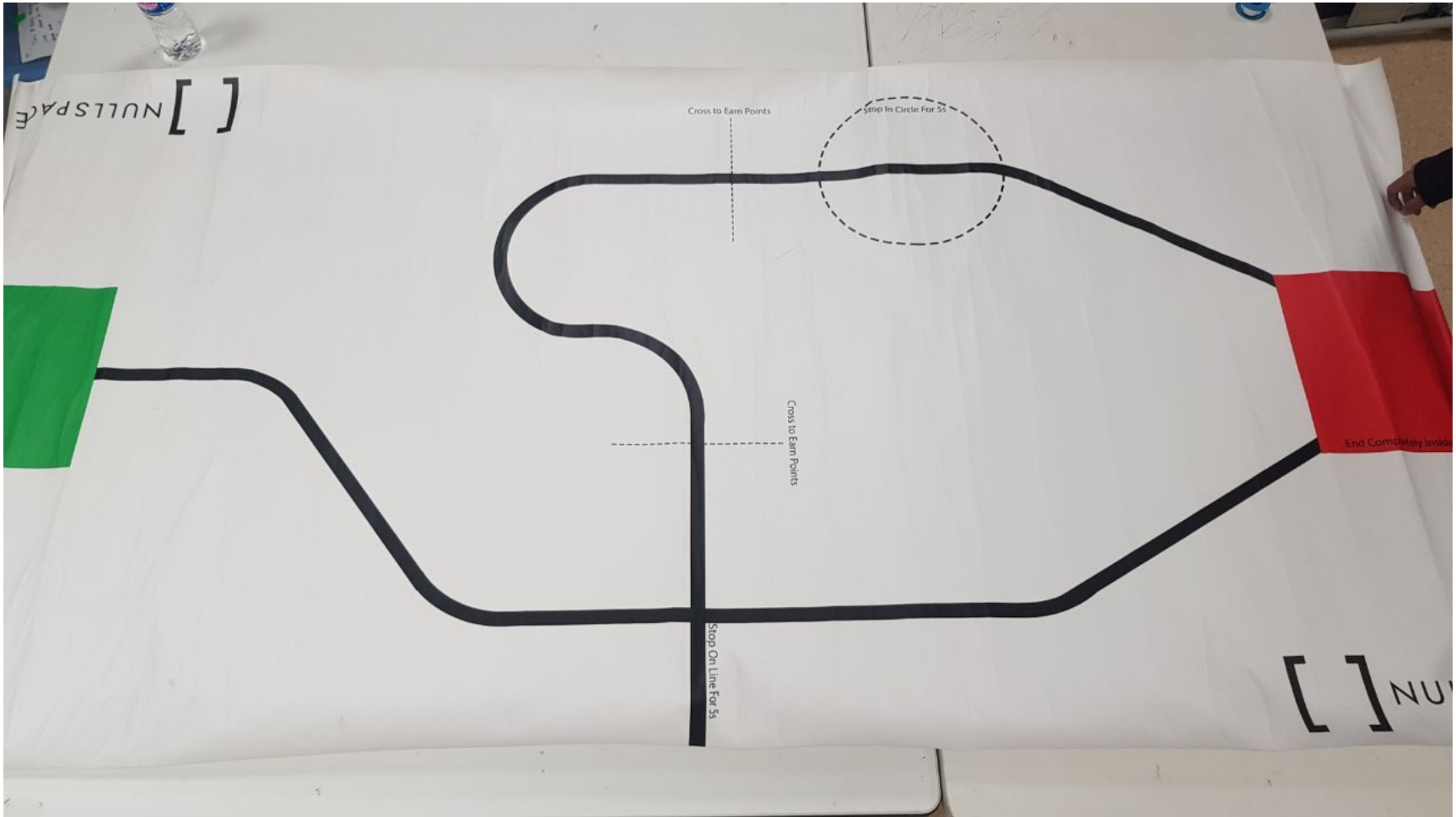
```
#include <Wire.h>
void setup() {
        Wire.begin();
        Serial.begin(9600);
        while (!Serial);          // Leonardo: wait for serial
monitor
        Serial.println("\nI2C Scanner");
}

 void loop() {
        byte error, address;
        int nDevices;
        Serial.println("Scanning...");
        nDevices = 0;

for(address = 1; address < 127; address++ ) {
        // The i2c_scanner uses the return value of the
Write.endTransmisstion to see if a device did acknowledge to
the address.
        Wire.beginTransmission(address);
        error = Wire.endTransmission();
        if (error == 0) {
                Serial.print("I2C device found at address 0x");
                if (address<16)
                        Serial.print("0");
                Serial.print(address,HEX);
                Serial.println("  !");
                nDevices++;
        }
else if (error==4) {
        Serial.print("Unknown error at address 0x");
        if (address<16)
                Serial.print("0");
        Serial.println(address,HEX); }
}

 if (nDevices == 0)
         Serial.println("No I2C devices found\n");
 else
        Serial.println("done\n");

 delay(5000);        // wait 5 seconds for next scan
}
```

Program to run

```
//Libraries
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

void setup(){
        lcd.begin(16,2);   // iInit the LCD for 16 chars 2 lines
        lcd.backlight();   // Turn on the backlight (try lcd.noBacklight() to
turn it off)
        lcd.setCursor(0,0); //First line
        lcd.print("I2C LCD Tutorial");
        lcd.setCursor(0,1); //Second line
        lcd.print("*Ardumotive.com*");
}

void loop(){
}
```

```
#include <IRremote.h> //include the library
#define Button_1 0xFF6897
#define Button_2 0xFF9867
int receiver = 13; //initialize pin 13 as recevier pin.
uint32_t Previous;
IRrecv irrecv(receiver); //create a new instance of receiver
decode_results results;

void setup() {
        Serial.begin(9600);
        irrecv.enableIRIn(); //start the receive
        pinMode(12, OUTPUT);
}

void loop() {
        if (irrecv.decode(&results)) { //if we have received an IR signal
        if (results.value==0xFFFFFFFF) {
                results.value=Previous;
        }
switch(results.value) {
        case Button_1 : digitalWrite(12, HIGH); break;
        case Button_2 : digitalWrite(12, LOW); break;
}
Serial.println (results.value, HEX); //display HEX results
irrecv.resume(); //next value
}
Previous=results.value;
}
```

Maze Challenge



Ultrasonic Sensor

VCC	to	5v
Gnd	to	Gnd
Trig	to	A4
Echo	to	A5

Ultrasonic Sensor + Arduino UNO + Motor Shield

```
#include <AFMotor.h>

const int MOTOR_2 = 2;
const int MOTOR_4 = 4;

//const int TRIG_PIN = A4;
//const int ECHO_PIN = A5;

AF_DCMotor motor2(MOTOR_2, MOTOR12_64KHZ);
AF_DCMotor motor4(MOTOR_4, MOTOR12_64KHZ);

void setup() {
  Serial.begin(9600);
  //pinMode(TRIG_PIN, OUTPUT);
  //pinMode(ECHO_PIN, INPUT);

  motor2.setSpeed(240);         // set the motor speed to 0-255
  motor4.setSpeed(240);

}
```

Ultrasonic Sensor + Arduino UNO + Motor Shield

```
void loop() {

 if (button up pressed) {     //move up
    motor2.run(FORWARD);
    motor4.run(FORWARD);
  }
  else if (button down pressed) {
    motor2.run(BACKWARD);
    motor4.run(BACKWARD);
  }

else if ( left button) {
    motor2.run(FORWARD);
    motor4.run(BACKWARD);
}
else if ( right button) {
    motor2.run(BACKWARD);  //might swap if wrong
    motor4.run(FORWARD);
}
```