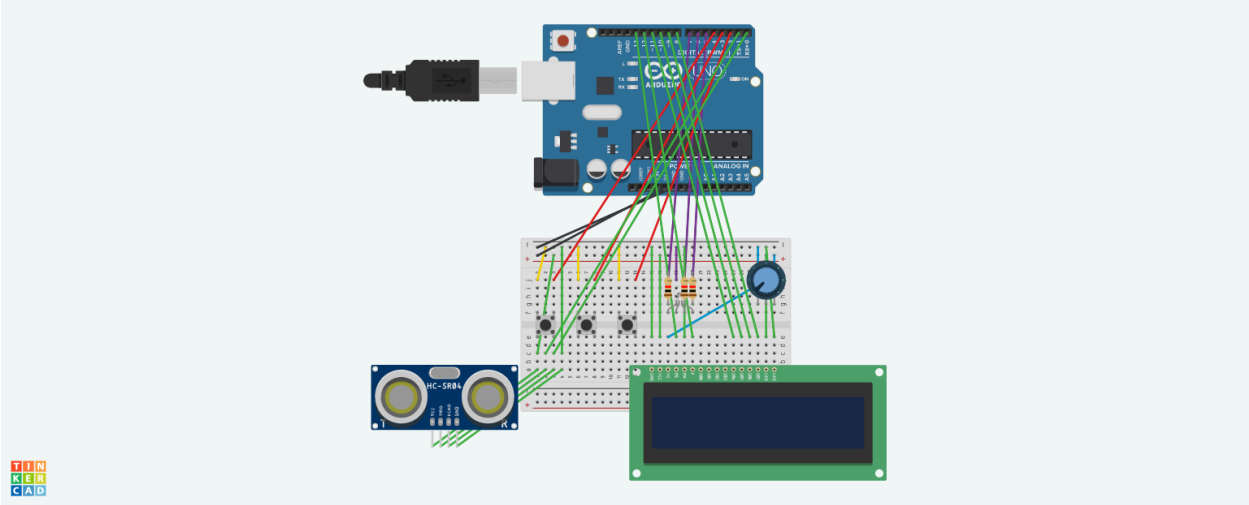
UNPLUG WIRES FROM PINS 0 & 1 WHILE UPLOADING

```
const int trigPin = 0;          //connects to the trigger pin on the distance sensor
const int echoPin = 1;          //connects to the echo pin on the distance sensor

const int redPin = 5;           //pin to control the red LED inside the RGB LED
const int greenPin = 6;         //pin to control the green LED inside the RGB LED
const int bluePin = 7;          //pin to control the blue LED inside the RGB LED

float distance = 0;             //stores the distance measured by the distance sensor

void setup()
{
  Serial.begin (9600);       //set up a serial connection with the computer

  pinMode(trigPin, OUTPUT);   //the trigger pin will output pulses of electricity
  pinMode(echoPin, INPUT);    //the echo pin will measure the duration of pulses coming back from the
distance sensor

  //set the RGB LED pins to output
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  distance = getDistance();   //variable to store the distance measured by the sensor

  Serial.print(distance);     //print the distance that was measured
  Serial.println(" in");      //print units after the distance

  if (distance >= 5 && distance <= 10) {

    //make the RGB LED red
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);

  } else if (distance >= 20 && distance <= 25 ) {

    //make the RGB LED yellow
    analogWrite(redPin, 200);
    analogWrite(greenPin, 200);
    analogWrite(bluePin, 0);

  } else if (distance >= 32 && distance <= 37)  {

    //make the RGB LED green
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);

  } else if (distance >= 40 && distance <= 45) {
```

```
    //make the RGB LED blue
    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 255);

   } else if (distance >= 14 && distance <= 19) {

    //make the RGB LED purple
    analogWrite(redPin, 128);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 128);

  } else if (distance >= 26 && distance <= 31) {

    //make the RGB LED white
    analogWrite(redPin, 255);
    analogWrite(greenPin, 204);
    analogWrite(bluePin, 229);

  }
  else {
    //make the RGB LED turn off
    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);
  }



  delay(50);      //delay 50ms between each reading
}

//------------------FUNCTIONS-------------------------------

//RETURNS THE DISTANCE MEASURED BY THE HC-SR04 DISTANCE SENSOR
float getDistance()
{
  float echoTime;                   //variable to store the time it takes for a ping to bounce off an object
  float calculatedDistance;         //variable to store the distance calculated from the echo time

  //send out an ultrasonic pulse that's 10ms long
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  echoTime = pulseIn(echoPin, HIGH);      //use the pulsein command to see how long it takes for the
                                          //pulse to bounce back to the sensor

  calculatedDistance = echoTime / 148.0;  //calculate the distance of the object that reflected the pulse (half the
bounce time multiplied by the speed of sound)

  return calculatedDistance;              //send back the distance that was calculated
```

}



Wiring Guide In Text:

- 330 Ohm resistors: g17 - i17, g19 - i19, g20 - i20
  RGB LED: f17-20 (longest leg in f18)
- Potentiometer: g28-30
- LCD: a15-30
- Ultrasonic Sensor: a1-4
- Button 1: e1, e3, f1, f3
  Button 2: e6, e8, f6, f8
  Button 3: e11, e13, f11, f13
- Connector Wires: c1 - (+), c2 - tx1, c3 - rx0, c4 - (-), j1 - (-), j3 - 4, j6 - (-), j8 - 3, j11 - (-), j13 - 2, e15 - (-), e16 - (+), e17 - i29, e18 - 13, e19 - (-), e20 - 12, e25 - 11, e26 - 10, e27 - 9, e28 - 8, e29 - (+), e30 - (-), i28 - (-), i30 - (+), j18 - (-), j17 - 7, j19 - 6, j20 - 5, GND - (-), 5V - (+)