NOISE POLLUTION MONITORING

Phase 4 Project Submission

Development part2

**Problem statement**

The noise pollution is a growing issue these days . As an engineer what is low-cost and effective ways to measure this pollution and encourage community to take certain steps to reduce it to reasonable limit.

In this IOT project, we will create an 10T based noise pollution monitoring system using ESP32, noise module, and Blynk app. We will use the KY-038 microphone sensorto detect noise in decibels and display noise in decibel on an OLED. Additionally, we will link our noise pollution monitor with Blynk application. To make our noise pollution monitor even more practical, users can monitor the sound decibels on the Blynk app. In other words, it becomes extremely handy to monitor the sound from anywhere through a mobile application. We will program our ESP32 board using Arduino IDE.

**KY-038 Noise Sensor Module**

The KY-038 noise sensor module consists of capacitance sensitive microphone (50Hz-10kHz) and an ampl ification circuit. The module converts noise waves to electrical signals.

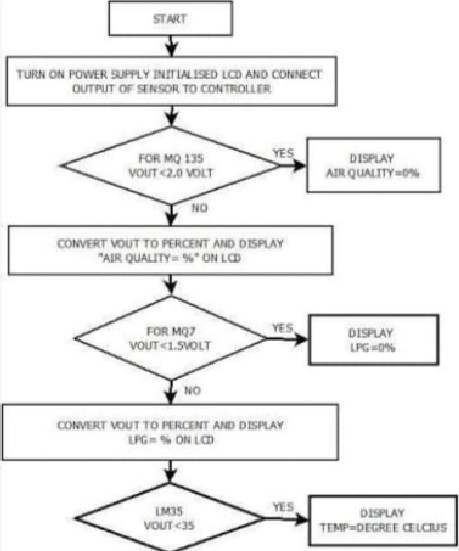
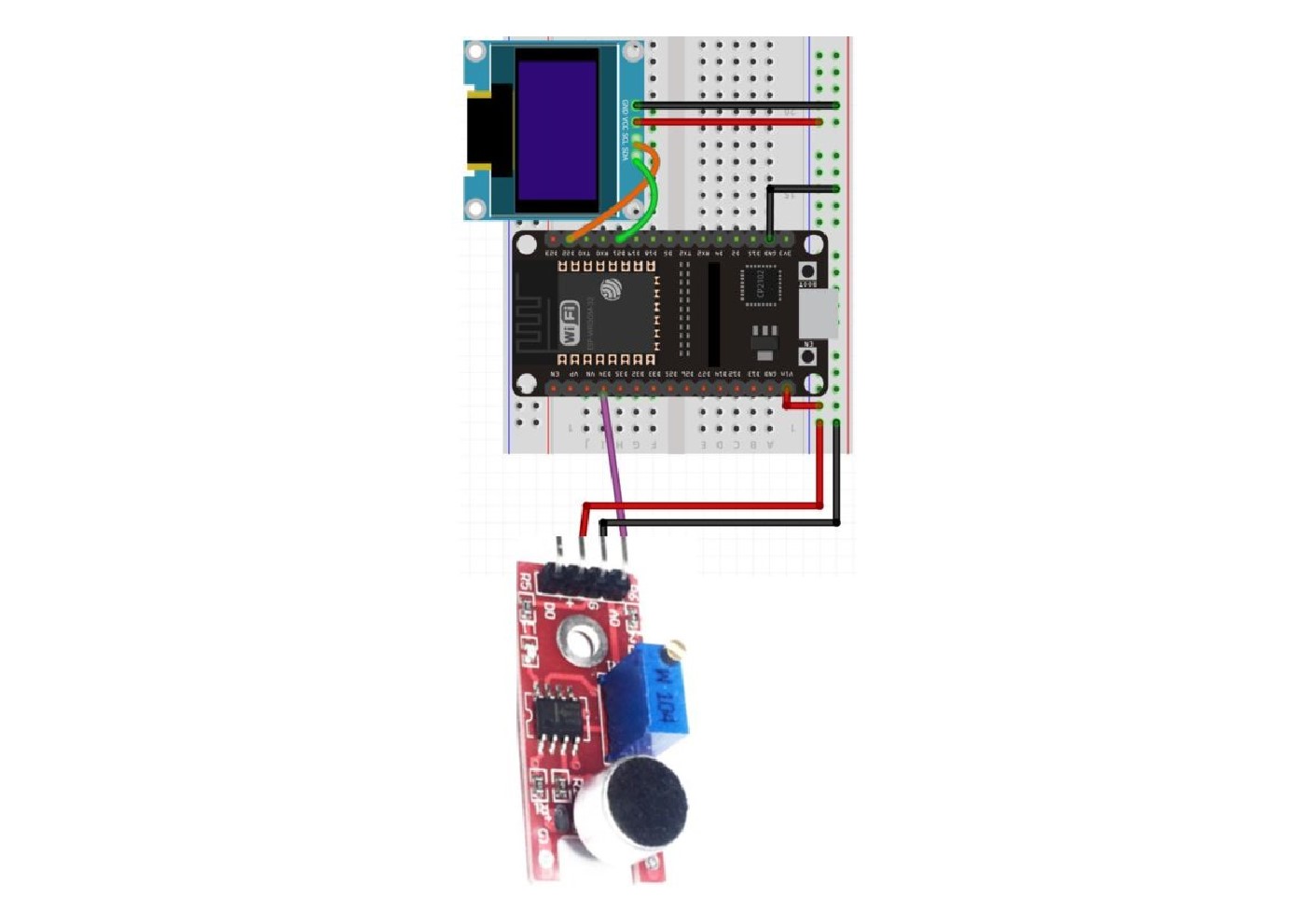
It detects the noise with the help of a microphone and then feeds this noise to processing circuitry which consists of an operational amplifier LM393. It also consists of a potentiometer which is used for setting the noise level and by setting this noise level the output of this noise sensor module could be easily controlled. Similarly, the output of this sensor could be checked by connecting the LED or any other device at output pins.

The OLED display has 4 terminals which we will connect with the ESP32 board. As the OLED display requires an operating voltage in the range of 3.3-5V hence we will connect the VCC terminal with 3,3V which will be in common with the ESP32 board. SCL of the display will be connected with the SCL pin of the module and the SDA of the display will be connected with the SDA of the module. By default, the 12C pin in ESP32 for SDA is GP1021, and for SCL is GP1022,

Installing Required Arduino Libraries for IOT noise

Pollution Monitoring System

We will use Arduino IDE to program our ESP32 development board. Make sure your Arduino IDE already has the ESP32 plugin installed. To program our ESP32 board for this sound monitor we will be required to install three libraries: BlynkSimpIeEsp32.h,



|  |  |
| --- | --- |
|  |  |
| 2194.” | |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | |  |
| ΦΑΤΑ wOlKE  ΤΟ | | |
|  | |  |
|  | | |
|  | |  |
|  | | |
|  | |  |
|  |  |  |

#define BLYNK PRINT Serial #include #include <BlynkSimpleEsp32.h>

#include

#include <Adafruit GFX.h> #include <Adafruit\_SSDI 306.h>

#define AO 34

Adafruit\_SSD1306 display = Adafruit\_SSD1306(128, 64,

&Wire,

unsigned int output; int Decibels;

char auth[] char ssid[l= "YOUR\_SSID"; char pass[] = "YOUR PASSWORD";



Blynk.virtualWrite(VO, Decibels);

void setup() {

Serial .begin( 1 15200) pinMode (AO, INPUT); ox3C); ! displ ay.begin(SS DI 306\_SWlTCHCAPVCC, ox3C)) {

Serial SSDI 306 allocation failed " ));

for(;;);

delay(2000); display.clearDisplay(); display.setTextCoIor(WHITE);

Blynk.begin(auth, ssid, pass);

void loop() {

Blynk,run(); unsigned long start\_time = millis(); float PeakToPeak = 0;

unsigned int maximum\_signal = O; //minimumvalue unsigned int minimum\_signal = 4095; //maximum value

while (millis() - start\_time < 50)

output = analogRead(AO);

if (output < 4095)

if (output > maximum\_signal)

maximum\_signal = output;

else if (output < minimum\_signal)

minimum\_signal = output;



void loop() {

Blynk.run(); unsigned long start\_time = millis(); float PeakToPeak = O,'

unsigned int maximum\_signal -O; //minimumvalue unsigned int minimum\_signal = 4095; //maximum value wh ile (millis() - start\_time < 50)

output = analogRead(AO);

if (output < 4095)

if (output > maximum\_signal)

maximum\_signal = output;

else if (output minimum\_signal)

minimum\_signal = output;

}dispIay.setTextSize(2); display.setCursor(O,10);



display,setTextSize(2); display.setCursor(40, 1 0); display.print("db");



if (Decibels 50)

{ display.setTextSizeC2); display.setCursor(Or30); display.print( "LOW"); display.display();

else if (Decibels > 50 && Decibels < 75)

display.setTextSize(2); display.setCursor(O,30); display.print( " Moderate" ); display.display();

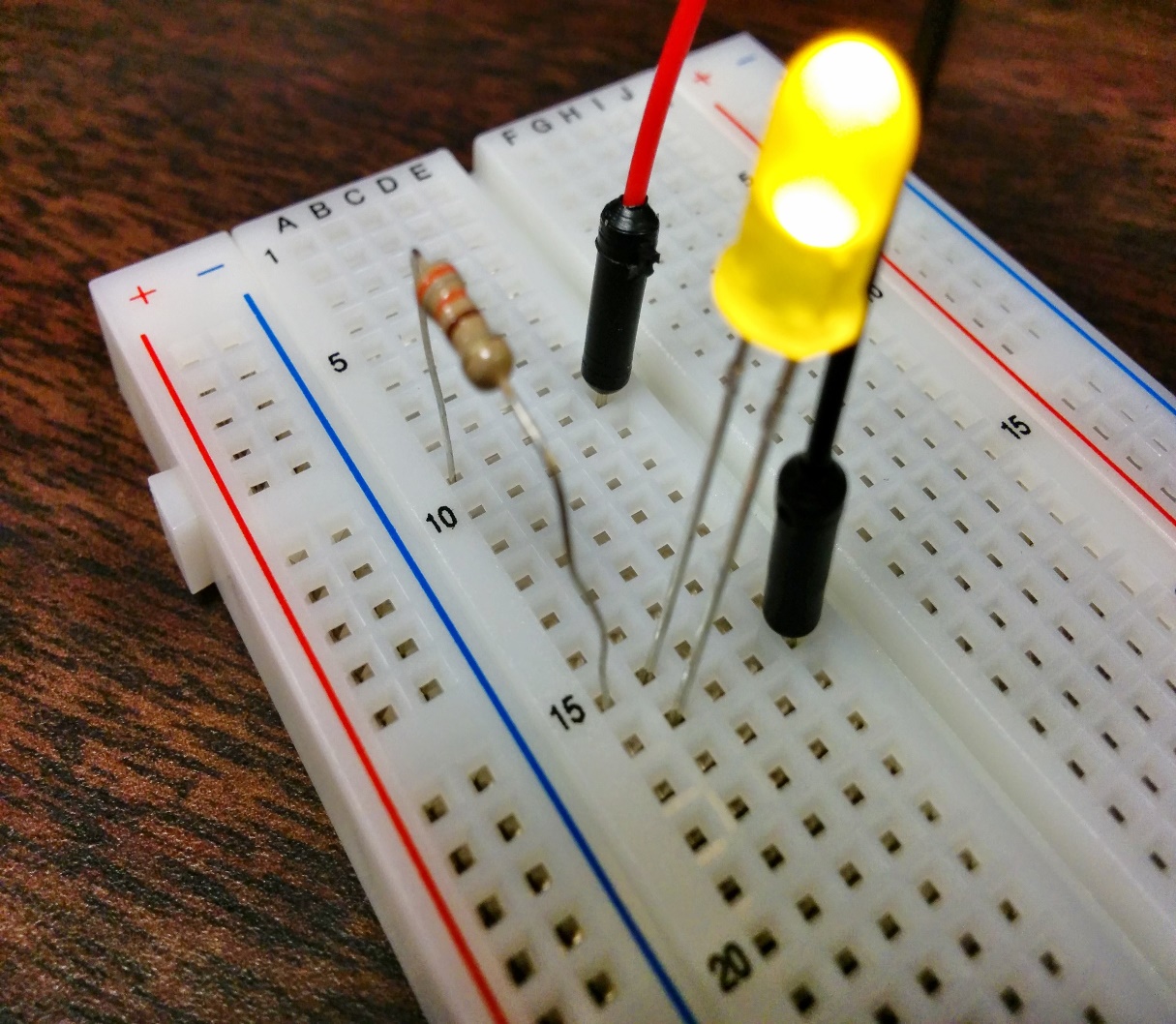
else if (Decibels 75)

display.setTextSize(2); display.setCursor(O,30);



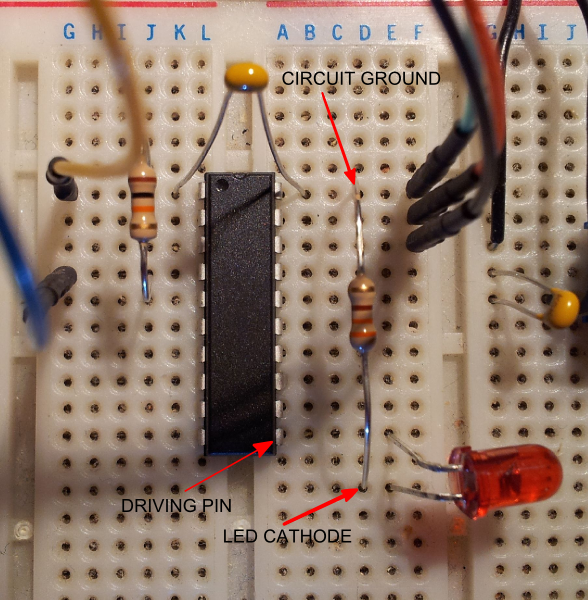
display.display();

delay(1000); display.cIearDispIay();



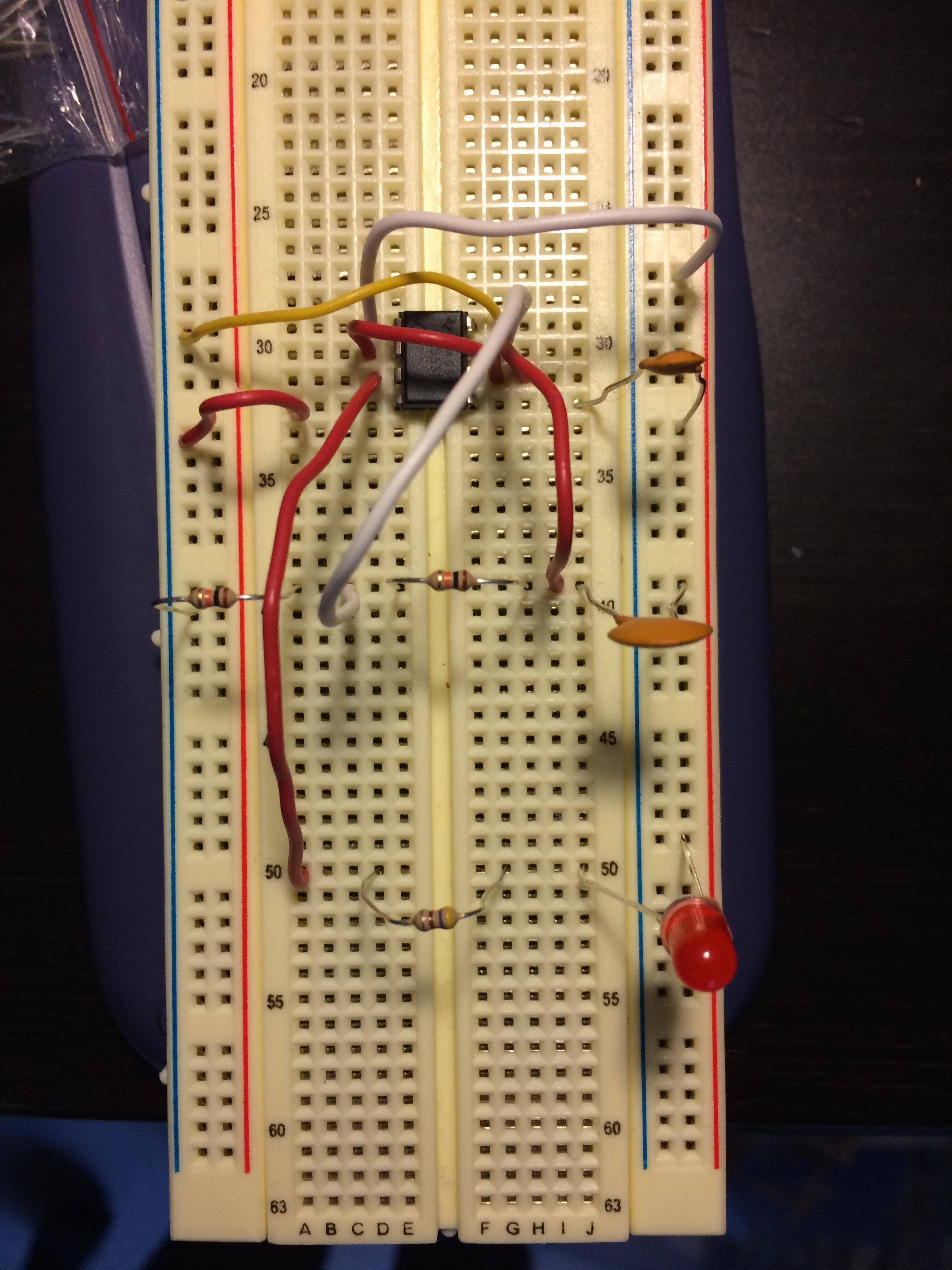
[This Photo](http://maxembedded.com/2014/07/using-raspberry-pi-gpio-using-python/) by Unknown Author is licensed under [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/3.0/)

A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a perfboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable.



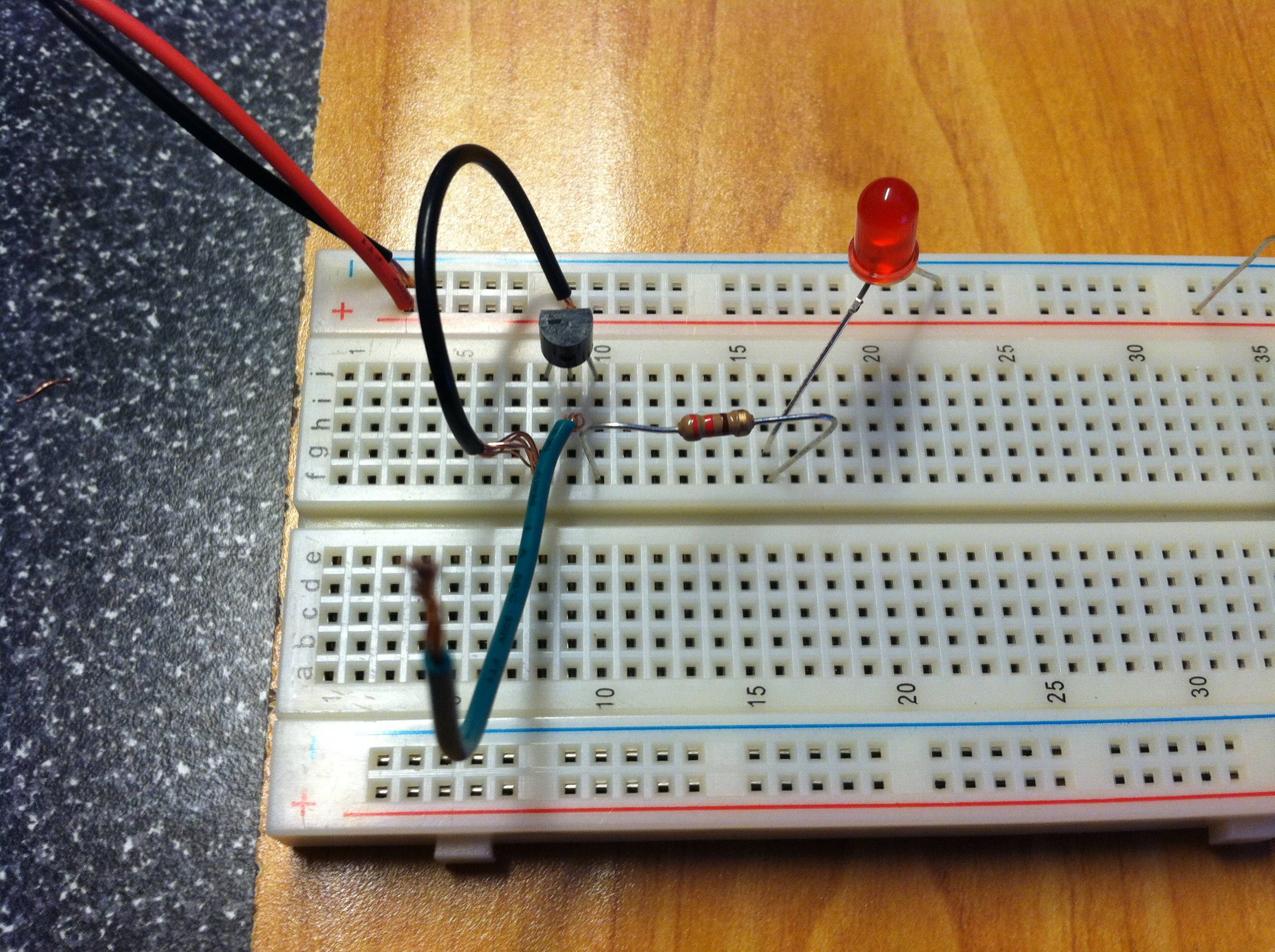
[This Photo](http://electronics.stackexchange.com/questions/52887/new-to-breadboarding-and-ic) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode).



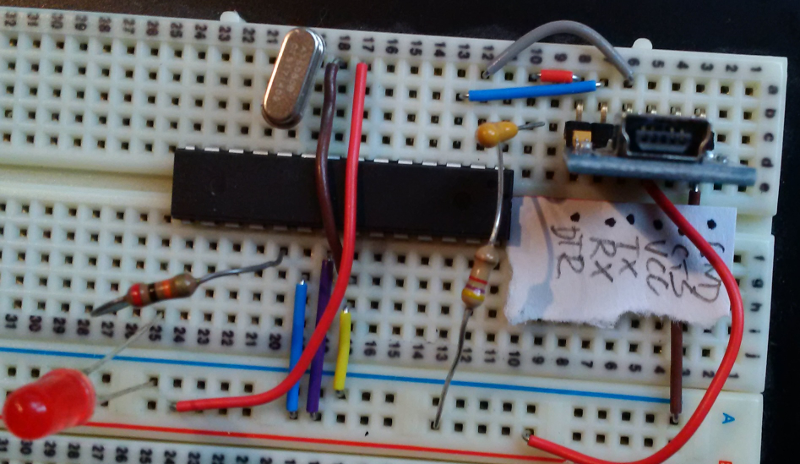
[This Photo](http://electronics.stackexchange.com/questions/140928/555-timer-ciruit-errors) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

The breadboard allows you to test circuits without having to solder components into a circuit and without having to use loads of connecting wires! It has little connection holes with stips of conducting metal connecting them underneath (either as 'power rails' or 'terminal strips').



[This Photo](http://electronics.stackexchange.com/questions/181003/simple-transistor-circuit-with-unconnected-gate-pin-acts-strangely) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Basically, there are three functional areas on a breadboard, the power rails near the long sides, the terminal strips (in some breadboards, these are the holes from rows a to j), and the middle groove. And remember that the inside of the breadboard (under the holes) is made up of sets of five metal clips.



[This Photo](http://electronics.stackexchange.com/questions/162641/garbled-serial-output-from-breadboard-atmega328-via-ftdi-breakout) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Diodes can be used as rectifiers, signal limiters, voltage regulators, switches, signal modulators, signal mixers, signal demodulators, and oscillators. The fundamental property of a diode is its tendency to conduct electric current in only one direction.// C program to append the contents of

// source file to the destination file

// including header files

#include <stdio.h>

// Function that appends the contents

**void** appendFiles(**char** source[],

**char** destination[])

{

    // declaring file pointers

**FILE** \*fp1, \*fp2;

    // opening files

    fp1 = **fopen**(source, "a+");

    fp2 = **fopen**(destination, "a+");

    // If file is not found then return.

**if** (!fp1 && !fp2) {

**printf**("Unable to open/"

               "detect file(s)\n");

**return**;

    }

**char** buf[100];

    // explicitly writing "\n"

    // to the destination file

    // so to enhance readability.

**fprintf**(fp2, "\n");

    // writing the contents of

    // source file to destination file.

**while** (!**feof**(fp1)) {

**fgets**(buf, **sizeof**(buf), fp1);

**fprintf**(fp2, "%s", buf);

    }

**rewind**(fp2);

    // printing contents of

    // destination file to stdout.

**while** (!**feof**(fp2)) {

**fgets**(buf, **sizeof**(buf), fp2);

**printf**("%s", buf);

    }

}

// Driver Code

**int** main()

{

**char** source[] = "file1.txt",

         destination[] = "file2.txt";

    // calling Function with file names.

    appendFiles(source, destination);

**return** 0;

}

Thanking you