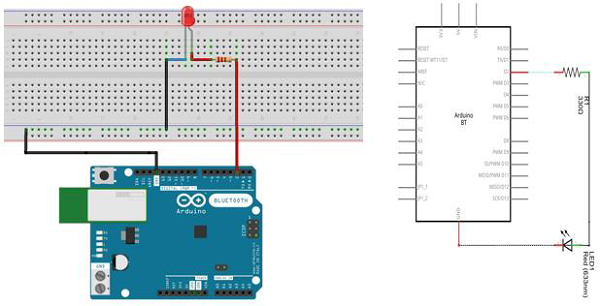
Expt1: To interface LED with Arduino and write a program to blink LED.

CIRCUIT:



CODE:

/\*

Blink

Turns on an LED on for one second, then off for one second, repeatedly.

\*/

// the setup function runs once when you press reset or power the board

void setup() { // initialize digital pin 13 as an output.

pinMode(2, OUTPUT);}

// the loop function runs over and over again forever

void loop() {

digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)

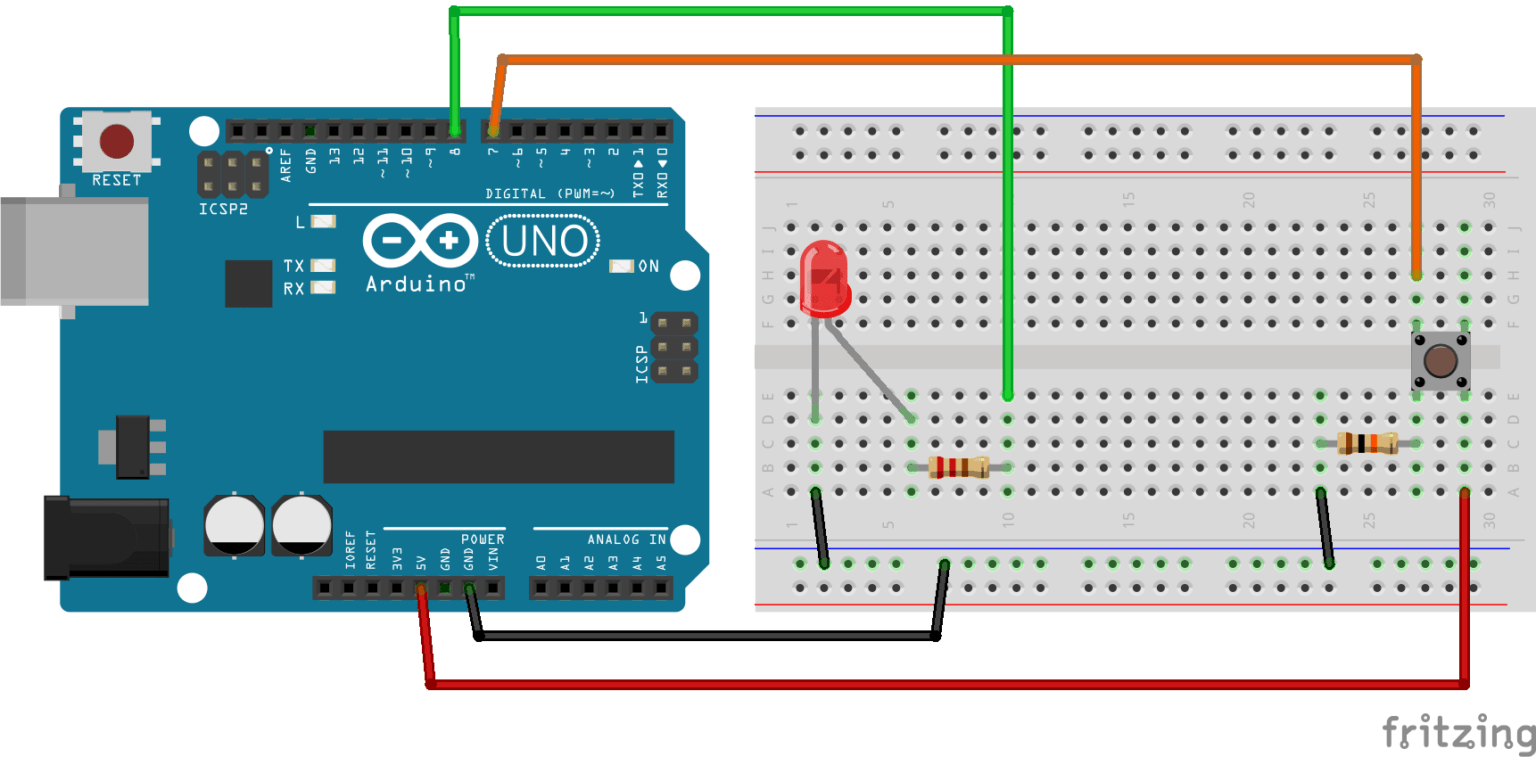
delay(1000); // wait for a second

digitalWrite(2, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second}

Expt2: To interface push button with Arduino and write a code to turn ON LED when push button is pressed.

CIRCUIT:



CODE:

#define LED\_PIN 8

#define BUTTON\_PIN 7

void setup() {

pinMode(LED\_PIN, OUTPUT);

pinMode(BUTTON\_PIN, INPUT);

}

void loop() {

if (digitalRead(BUTTON\_PIN) == HIGH) {

digitalWrite(LED\_PIN, HIGH);

}

else {

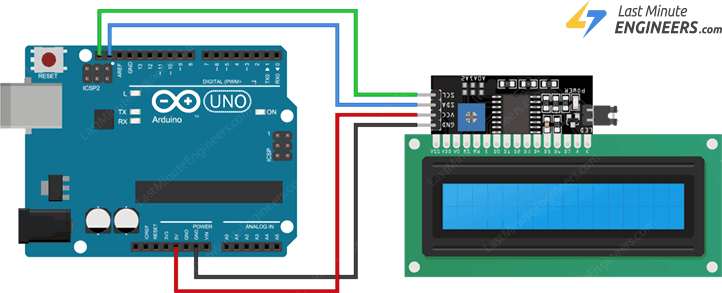
digitalWrite(LED\_PIN, LOW);

}

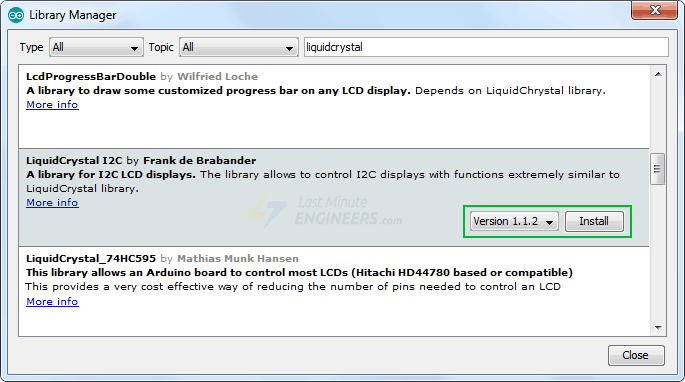
}

Expt3: To interface LCD with Arduino/Raspberry Pi and write a program to display a message .

Circuit:



Librabry installation:



Code to get the address:

#include <Wire.h>

void setup() {

Wire.begin();

Serial.begin(9600);

while (!Serial); // Leonardo: wait for serial monitor

Serial.println("\nI2C Scanner");

}

void loop() {

int nDevices = 0;

Serial.println("Scanning...");

for (byte 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);

byte 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

}

Code to print on LCD screen :

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3F,16,2); // set the LCD address to 0x3F for a 16 chars and 2 line display

void setup() {

lcd.init();

lcd.clear();

lcd.backlight(); // Make sure backlight is on

// Print a message on both lines of the LCD.

lcd.setCursor(2,0); //Set cursor to character 2 on line 0

lcd.print("Hello world!");

lcd.setCursor(2,1); //Move cursor to character 2 on line 1

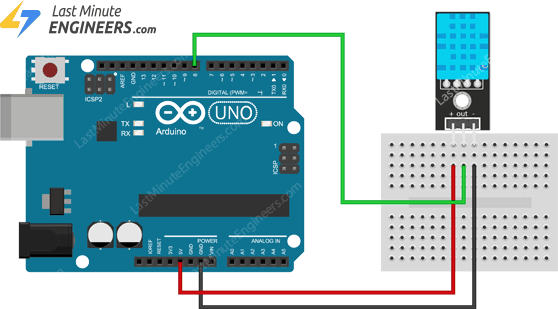
lcd.print("LCD Tutorial");

}

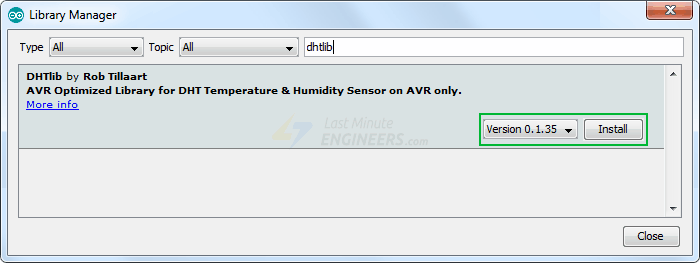
void loop() {}

Expt 4: To interface DHT11/ DHT22 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.

Circuit:



Library Installation:



Code:

#include <dht.h> // Include library

#define outPin 8 // Defines pin number to which the sensor is connected

dht DHT; // Creates a DHT object

void setup() {

Serial.begin(9600);

}

void loop() {

int readData = DHT.read11(outPin);

float t = DHT.temperature; // Read temperature

float h = DHT.humidity; // Read humidity

Serial.print("Temperature = ");

Serial.print(t);

Serial.print("°C | ");

Serial.print((t\*9.0)/5.0+32.0); // Convert celsius to fahrenheit

Serial.println("°F ");

Serial.print("Humidity = ");

Serial.print(h);

Serial.println("% ");

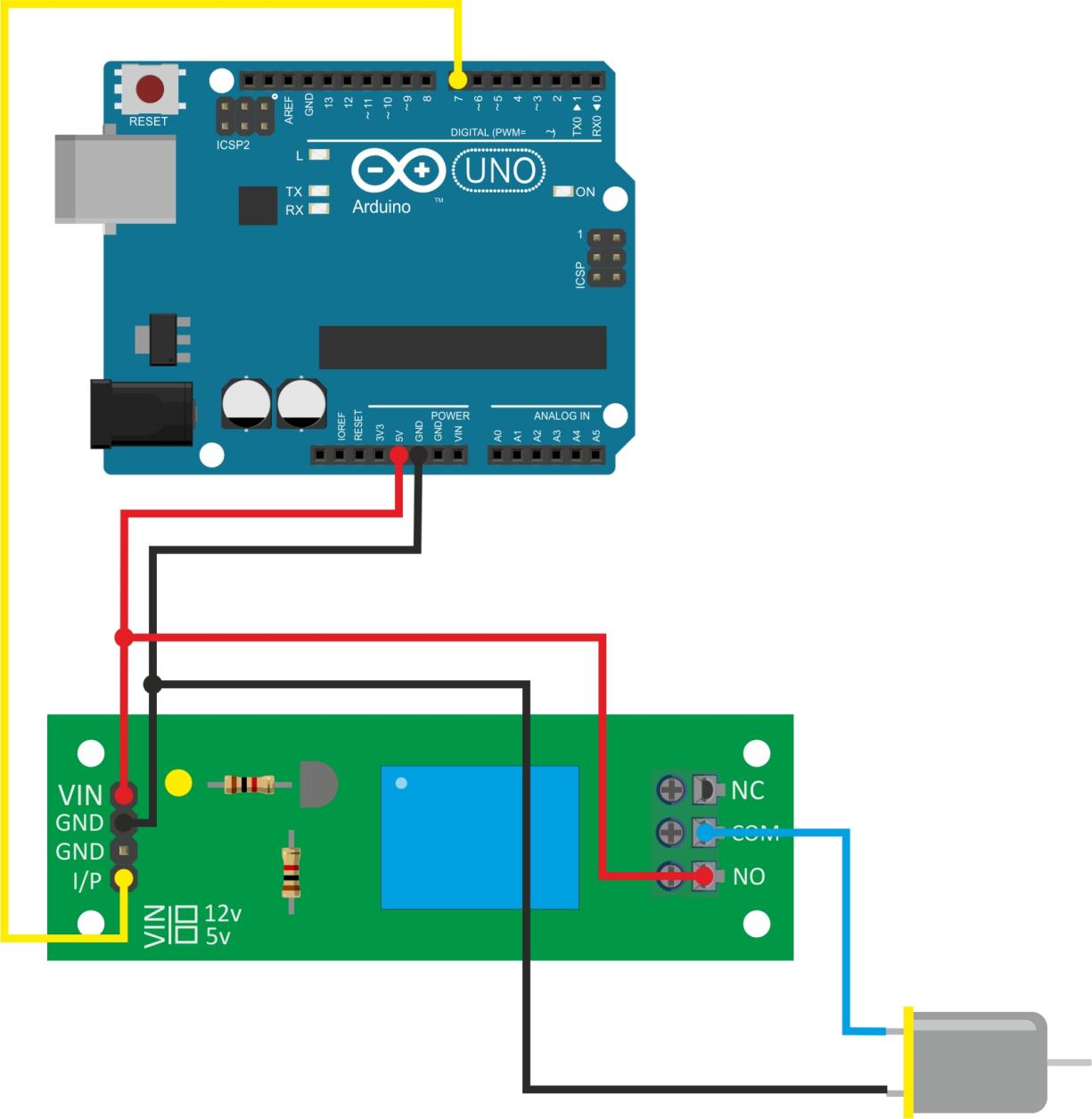
Serial.println("");

delay(2000); // wait two seconds

}

Expt 5: To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.

Circuit:



Code:

int relay = 7;

// the setup function runs once when you press reset or power the boardvoid setup() {

 // initialize digital pin LED\_BUILTIN as an output.

 pinMode(relay, OUTPUT);}

// the loop function runs over and over again forevervoid loop() {

 digitalWrite(relay, HIGH);   // turn the LED on (HIGH is the voltage level)

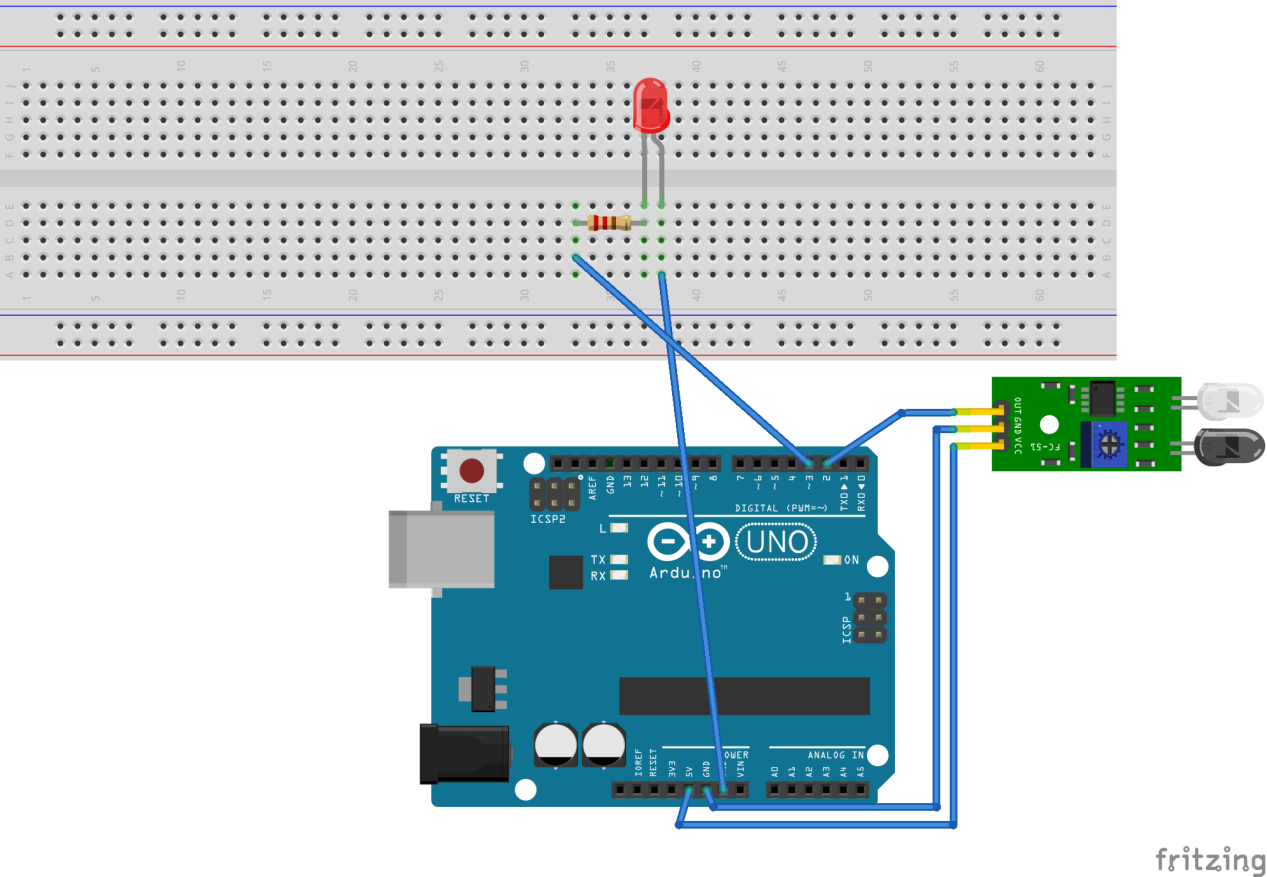
 delay(2000);                       // wait for a second

 digitalWrite(relay, LOW);    // turn the LED off by making the voltage LOW

 delay(2000);                       // wait for a second}

Expt 6: To interface IR sensor with Arduino/Raspberry Pi and write an application to detect obstacle and notify user using LEDs.

Circuit:



Code:

/\*\*\* www.arduinopoint.com \*\*\*/

/\*\*\* Arduino with IR Sensor \*\*\*/

int SensorPin = 2;

int OutputPin = 13;

void setup() {

pinMode(OutputPin, OUTPUT);

pinMode(SensorPin, INPUT);

Serial.begin(9600);

}

void loop() {

int SensorValue = digitalRead(SensorPin);

Serial.print("SensorPin Value: ");

Serial.println(SensorValue);

delay(1000);

if (SensorValue==LOW){ // LOW MEANS Object Detected

digitalWrite(OutputPin, HIGH);

}

else

{

digitalWrite(OutputPin, LOW);

}

}