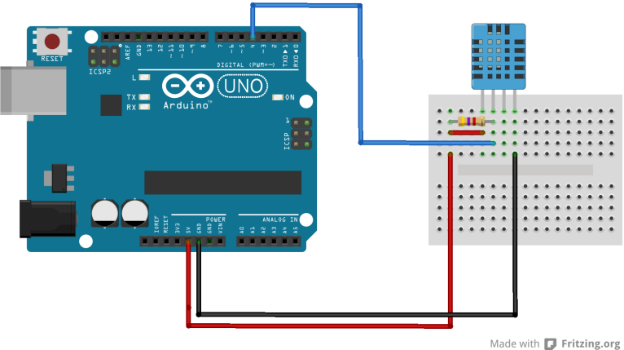
**1.  DHT11 with Arduino.**  
#include <dht11.h>

#define DHT11PIN 4

dht11 DHT11;

void setup()

{

Serial.begin(9600);

}

void loop()

{

Serial.println();

int chk = DHT11.read(DHT11PIN);

Serial.print("Humidity (%): ");

Serial.println((float)DHT11.humidity, 2);

Serial.print("Temperature (C): ");

Serial.println((float)DHT11.temperature, 2);

delay(2000);

}

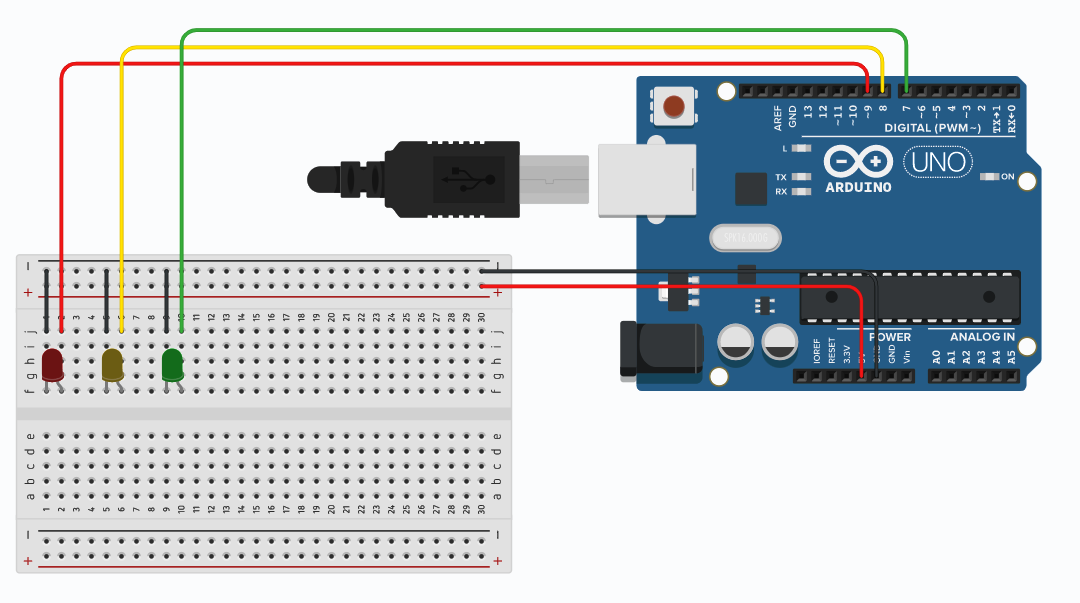
<https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621>

**2. Traffic Light using Arduino**

int red = 9;

int yellow = 8;

int green = 7;



void setup(){

pinMode(red, OUTPUT);

pinMode(yellow, OUTPUT);

pinMode(green, OUTPUT);

}

void loop(){

digitalWrite(red, HIGH);

delay(15000);

digitalWrite(red, LOW);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(green, HIGH);

delay(20000);

digitalWrite(green, LOW);

//

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

digitalWrite(yellow, HIGH);

delay(1000);

digitalWrite(yellow, LOW);

delay(500);

}

<https://projecthub.arduino.cc/agarwalkrishna3009/traffic-light-using-arduino-a-beginner-project-35f8c6>

3. **Ultrasonic Sensor Distance Radar**

#include <Servo.h>

/\*

code for arduino bord ultrasonic radar system.

for more arduino projects ---> teamdiy.in

\*/

Servo leftRightServo; // set a variable to map the servo

int leftRightPos = 0; // set a variable to store the servo position

const int numReadings = 10; // set a variable for the number of readings to take

int index = 0; // the index of the current reading

int total = 0; // the total of all readings

int average = 0; // the average

int echoPin = 7; // the SRF05's echo pin

int initPin = 8; // the SRF05's init pin

unsigned long pulseTime = 0; // variable for reading the pulse

unsigned long distance = 0; // variable for storing distance

/\* setup the pins, servo and serial port \*/

void setup() {

leftRightServo.attach(9);

// make the init pin an output:

pinMode(initPin, OUTPUT);

// make the echo pin an input:

pinMode(echoPin, INPUT);

// initialize the serial port:

Serial.begin(9600);

}

/\* begin rotating the servo and getting sensor values \*/

void loop() {

for(leftRightPos = 0; leftRightPos < 180; leftRightPos++) { // going left to right.

leftRightServo.write(leftRightPos);

for(index = 0; index<=numReadings;index++) // take x number of readings from the sensor and average them

digitalWrite(initPin, LOW);

delayMicroseconds(50);

digitalWrite(initPin, HIGH);// send signal

delayMicroseconds(50); // wait 50 microseconds for it to return

digitalWrite(initPin, LOW); // close signal

pulseTime = pulseIn(echoPin, HIGH); // calculate time for signal to return

distance = pulseTime/58; // convert to centimetres

total = total + distance; // update total

delay(10);

}

average = total/numReadings; // create average reading

if (index >= numReadings){// reset the counts when at the last item of the array

index = 0;

total = 0;

}

Serial.print("X"); // print leading X to mark the following value as degrees

Serial.print(leftRightPos); // current servo position

Serial.print("V"); // preceeding character to separate values

Serial.println(average); // average of sensor readings

}

/\*

start going right to left after we got to 180 degrees

same code as above

\*/

for(leftRightPos = 180; leftRightPos > 0; leftRightPos--) { // going right to left

leftRightServo.write(leftRightPos);

for (index = 0; index<=numReadings;index++) {

digitalWrite(initPin, LOW);

delayMicroseconds(50);

digitalWrite(initPin, HIGH);

delayMicroseconds(50);

digitalWrite(initPin, LOW);

pulseTime = pulseIn(echoPin, HIGH);

distance = pulseTime/58;

total = total + distance;

delay(10);

}

average = total/numReadings;

if (index >= numReadings) {

index = 0;

total = 0;

}

Serial.print("X");

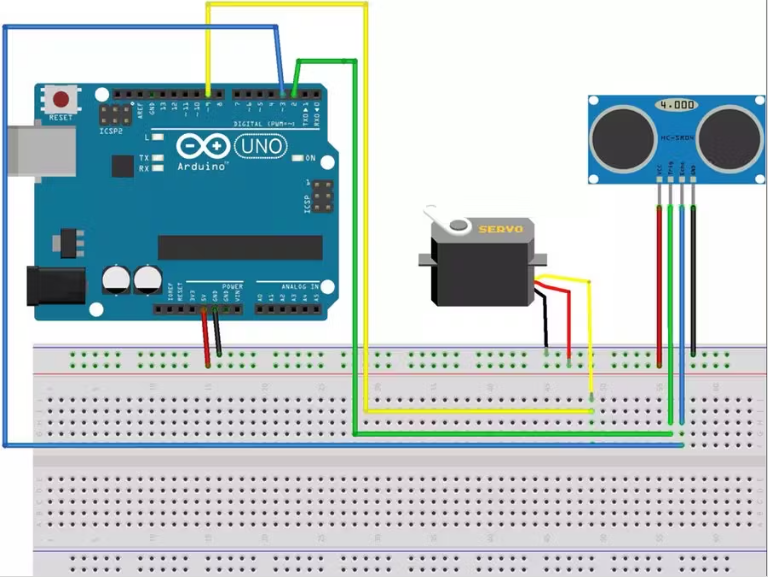
Serial.print(leftRightPos);

Serial.print("V");

Serial.println(average);

}

}



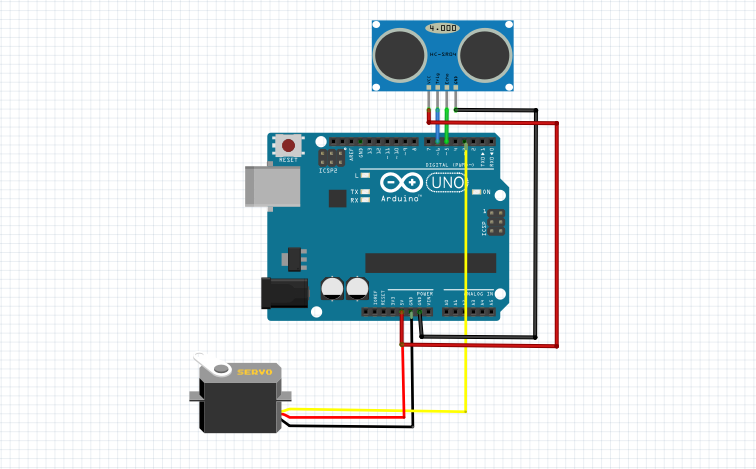
<https://www.hackster.io/Manikantsavadatti/diy-ultrasonic-radar-system-1f4d1c>

4. **Arduino Trash Can**

#include<Servo.h>

Servo servo;

int const trigPin = 6;

int const echoPin = 5;

void setup()

{

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

servo.attach(3);

}

void loop()

{ int duration, distance;

digitalWrite(trigPin, HIGH);

delay(1);

digitalWrite(trigPin, LOW);

// Measure the pulse input in echo pin

duration = pulseIn(echoPin, HIGH);

// Distance is half the duration devided by 29.1 (from datasheet)

distance = (duration/2) / 29.1;

// if distance less than 0.5 meter and more than 0 (0 or less means over range)

if (distance <= 50 && distance >= 0) {

servo.write(50);

delay(3000);

} else {

servo.write(160);

}

// Waiting 60 ms won't hurt any one

delay(60);

}

<https://projecthub.arduino.cc/ashraf_minhaj/arduino-trash-bot-auto-openclose-trash-bin-c81eff>

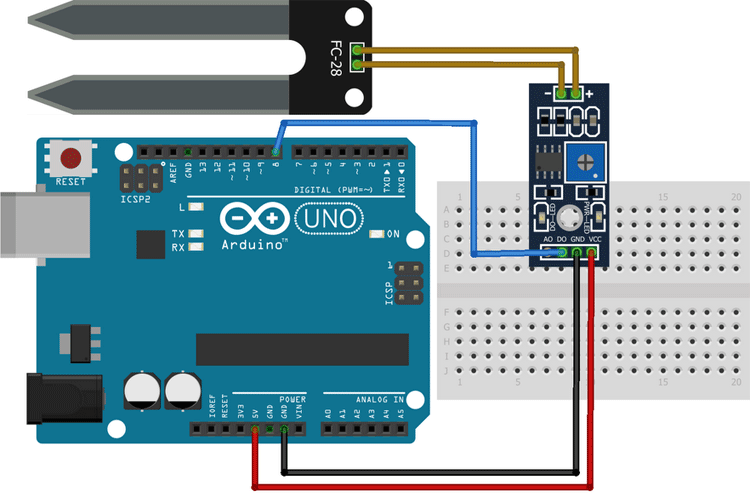
5. **Soil Moisture Sensor**

// Moisture Sensor Arduino Code

//By Circuitdigest

#define ledPin 6

#define sensorPin A0



void setup() {

  Serial.begin(9600);

  pinMode(ledPin, OUTPUT);

  digitalWrite(ledPin, LOW);

}

void loop() {

  Serial.print("Analog output: ");

  Serial.println(readSensor());

  delay(500);

}

//  This function returns the analog data to calling function

int readSensor() {

  int sensorValue = analogRead(sensorPin);  // Read the analog value from sensor

  int outputValue = map(sensorValue, 0, 1023, 255, 0); // map the 10-bit data to 8-bit data

  analogWrite(ledPin, outputValue); // generate PWM signal

  return outputValue;             // Return analog moisture value

}

<https://circuitdigest.com/microcontroller-projects/interfacing-soil-moisture-sensor-with-arduino-uno>

**6. CONTROLLED LEDS USING ARDUINO AND Push Button**

void setup() {

  pinMode(12,OUTPUT);

  pinMode(2,INPUT);

}

void loop() {

  if(digitalRead(2) == HIGH){

  digitalWrite(12,HIGH);

  }

  else{

digitalWrite(12,LOW);

  }

}

https://www.youtube.com/watch?v=yBgMJssXqHY&t=1121s

**7. ARDUINO AND MQ-135 Air Quality Monitoring**

#include <LiquidCrystal\_I2C.h>

#include "MQ135.h"

#define ANALOGPIN A2 // Define Analog PIN on Arduino Board

#define RZERO 206.85 // Define RZERO Calibration Value

MQ135 gasSensor = MQ135(ANALOGPIN);

LiquidCrystal\_I2C lcd(0x27, 16,2);

void setup()

{

lcd.init();

lcd.begin(16,2);//Defining 16 columns and 2 rows of lcd display

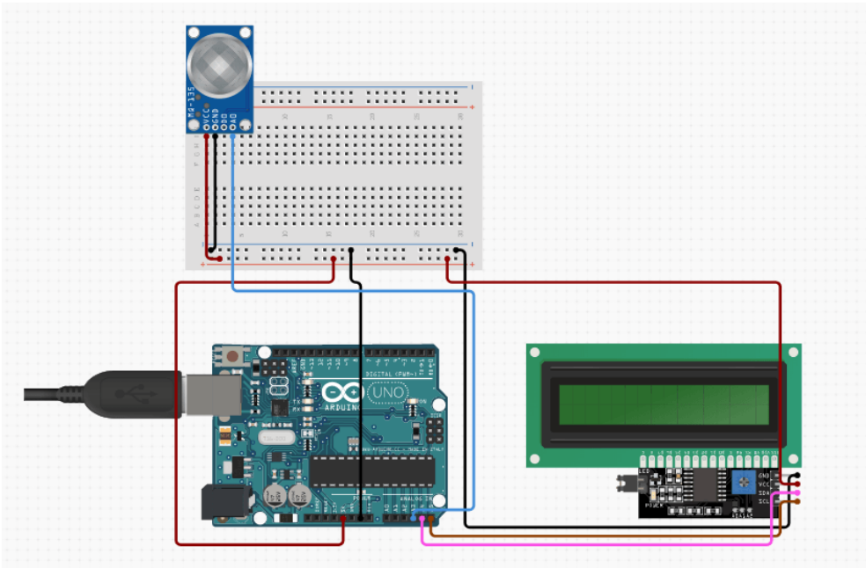
lcd.backlight();

Serial.begin(9600);

float rzero = gasSensor.getRZero();

delay(3000);

Serial.print("MQ135 RZERO Calibration Value : ");

 Serial.println(rzero);

}

void loop() {

float ppm = gasSensor.getPPM();

delay(1000);

Serial.print("CO2 ppm value : ");

Serial.println(ppm);

lcd.setCursor(0,0);

lcd.print(" Tech Maker ");

lcd.setCursor(0,1);

lcd.print("CO2 : ");

lcd.print(ppm);

lcd.print("PPM");

}

<https://electronicsprojectshub.com/interfacing-mq-135-gas-sensor-with-arduino/>

**8. ARDUINO BASED MOTION DETECTION USING PIR SENSOR**

const int pirPin = 5;  // Digital pin connected to the PIR sensor

void setup() {

  pinMode(pirPin, INPUT);

  pinMode(7, OUTPUT);  // Set PIR sensor pin as input

  Serial.begin(9600);      // Initialize serial communication for debugging (optional)

  delay(5000);

}

void loop() {

  int pirState = digitalRead(pirPin); // Read PIR sensor state (HIGH or LOW)

  if (pirState == HIGH) {

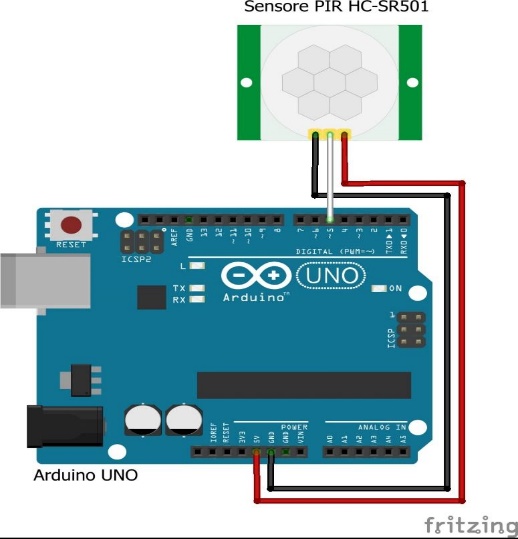
    Serial.println("Motion detected!");

    delay(5000); // Delay for 1 second (adjust as needed)

    // Motion detected

    digitalWrite(7, HIGH);

delay(500);



  } else {

    // No motion

    Serial.println("No motion detected.");

    delay(5000);

    digitalWrite(7,LOW);

  delay(500);

  }

}

<https://circuitdigest.com/microcontroller-projects/interface-pir-sensor-with-arduino>

**9. Rain Detection System using Arduino and Rain Sensor**

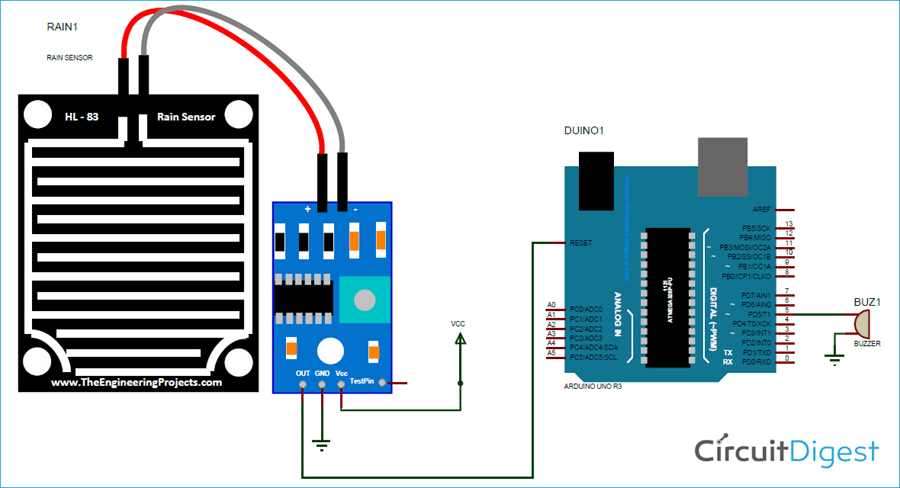
#define rainfall A0

#define buzzer 5

int value;

int set=10;

void setup() {

  Serial.begin(9600);

  pinMode(buzzer,OUTPUT);

  pinMode(rainfall,INPUT);

  }

void loop() {

 value = analogRead(rainfall);

 Serial.println("LOL");

 Serial.println(value);

 value = map(value,0,1023,225,0);

 Serial.println(value);

 if(value>=set){

  Serial.println("rain detected");

  digitalWrite(buzzer,HIGH);

 }

 else{

  digitalWrite(buzzer,LOW);

 }

 delay(200);

}

<https://circuitdigest.com/microcontroller-projects/rain-detector-using-arduino>

**10. Audio Frequency Detector**

#include "arduinoFFT.h"

#define SAMPLES 128 //SAMPLES-pt FFT. Must be a base 2 number. Max 128 for Arduino Uno.

#define SAMPLING\_FREQUENCY 2048 //Ts = Based on Nyquist, must be 2 times the highest expected frequency.

arduinoFFT FFT = arduinoFFT();

unsigned int samplingPeriod;

unsigned long microSeconds;

double vReal[SAMPLES]; //create vector of size SAMPLES to hold real values

double vImag[SAMPLES]; //create vector of size SAMPLES to hold imaginary values

void setup()

{

Serial.begin(115200); //Baud rate for the Serial Monitor

samplingPeriod = round(1000000\*(1.0/SAMPLING\_FREQUENCY)); //Period in microseconds

}

void loop()

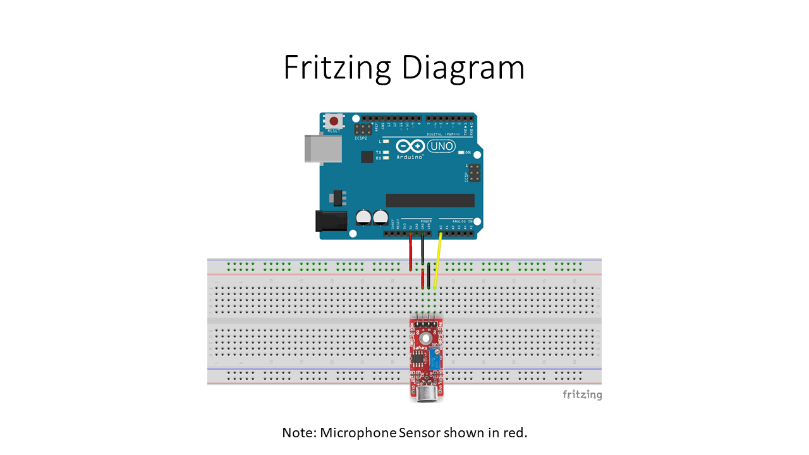
{

/\*Sample SAMPLES times\*/

for(int i=0; i<SAMPLES; i++)

{

microSeconds = micros(); *//Returns the number of microseconds since the Arduino board began running the current script.*

vReal[i] = analogRead(0); //Reads the value from analog pin 0 (A0), quantize it and save it as a real term.

vImag[i] = 0; //Makes imaginary term 0 always

/\*remaining wait time between samples if necessary\*/

while(micros() < (microSeconds + samplingPeriod))

{

//do nothing

}

}

/\*Perform FFT on samples\*/

FFT.Windowing(vReal, SAMPLES, FFT\_WIN\_TYP\_HAMMING, FFT\_FORWARD);

FFT.Compute(vReal, vImag, SAMPLES, FFT\_FORWARD);

FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);

/\*Find peak frequency and print peak\*/

double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING\_FREQUENCY);

Serial.println(peak); //Print out the most dominant frequency.

/\*Script stops here. Hardware reset required.\*/

while (1); //do one time

}

<https://projecthub.arduino.cc/lbf20012001/audio-frequency-detector-d300e3>

**11. ARDUINO PIANO**

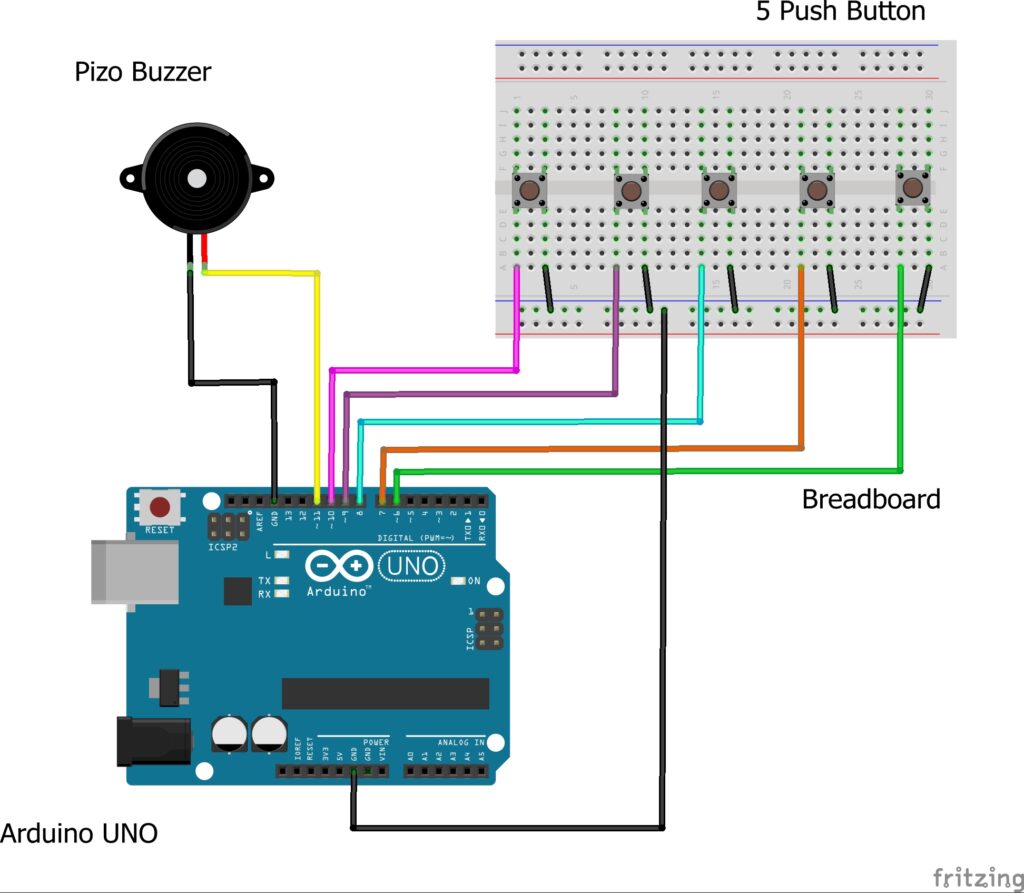
#define NOTE\_C 262

#define NOTE\_D 294

#define NOTE\_E 330

#define NOTE\_A 440

#define NOTE\_B 493

 #define ACTIVATED LOW

const int PIEZO = 11;

const int BUTTON\_C = 10;

const int BUTTON\_D = 9;

const int BUTTON\_E = 6;

const int BUTTON\_A = 5;

const int BUTTON\_B = 4;

void setup()

{

pinMode(BUTTON\_C, INPUT);

digitalWrite(BUTTON\_C,HIGH);

pinMode(BUTTON\_D, INPUT);

digitalWrite(BUTTON\_D,HIGH);

pinMode(BUTTON\_E, INPUT);

digitalWrite(BUTTON\_E,HIGH);

pinMode(BUTTON\_A, INPUT);

digitalWrite(BUTTON\_A,HIGH);

pinMode(BUTTON\_B, INPUT);

digitalWrite(BUTTON\_B,HIGH);

}

void loop()

{

while(digitalRead(BUTTON\_C) == ACTIVATED)

{

tone(PIEZO,NOTE\_C);

}

while(digitalRead(BUTTON\_D) == ACTIVATED)

{

tone(PIEZO,NOTE\_D);

}

while(digitalRead(BUTTON\_E) == ACTIVATED)

{

tone(PIEZO,NOTE\_E);

}

while(digitalRead(BUTTON\_A) == ACTIVATED)

{

tone(PIEZO,NOTE\_A);

}

while(digitalRead(BUTTON\_B) == ACTIVATED)

{

tone(PIEZO,NOTE\_B);

}

noTone(PIEZO);

}

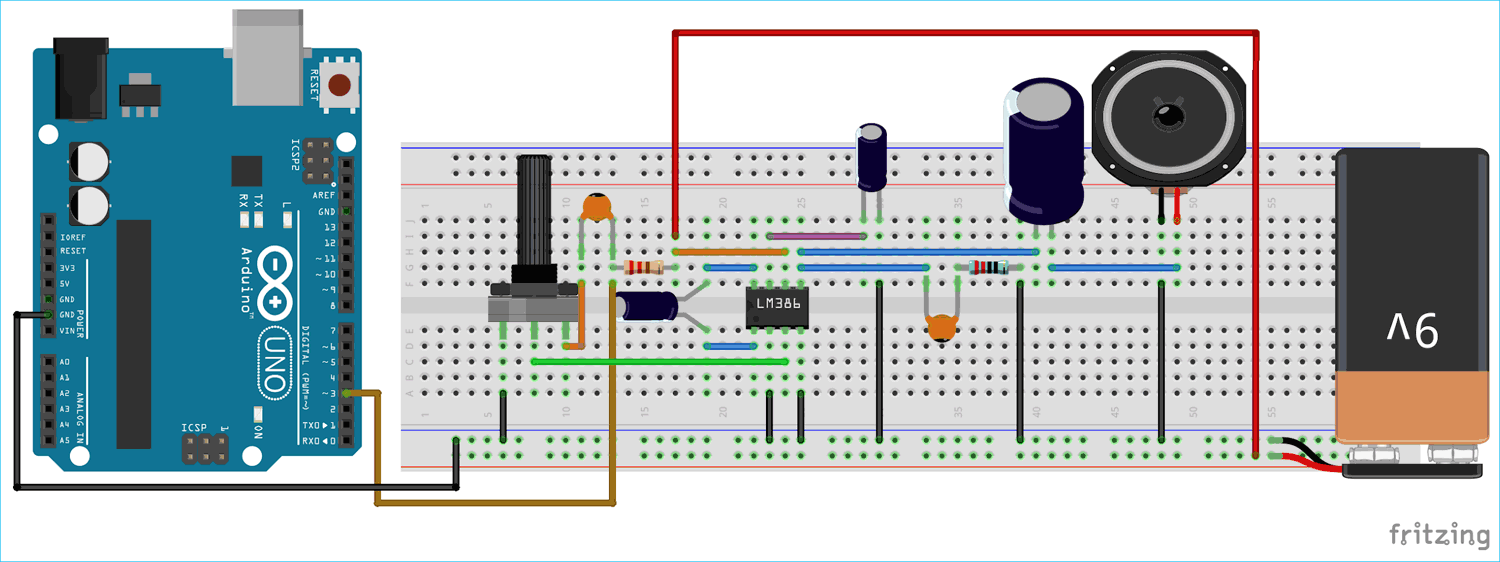
<https://techatronic.com/arduino-based-piano-using-push-buttons-arduino-mini-projects/>

**12. ARDUINO BASED SPEECH TO TEXT**

#include "Talkie.h"

#include "Vocab\_US\_Large.h"

#include "Vocab\_Special.h"

Talkie voice;

void setup() {

}

void loop() {

    voice.say(spPAUSE2);

    voice.say(sp2\_DANGER);

    voice.say(sp2\_DANGER);

    voice.say(sp3\_STORM);

    voice.say(sp3\_IN);

    voice.say(sp3\_THE);

    voice.say(sp3\_NORTH);

}

<https://circuitdigest.com/microcontroller-projects/arduino-based-text-to-speech-converter>

**13. Automatic Water Dispenser using Arduino**

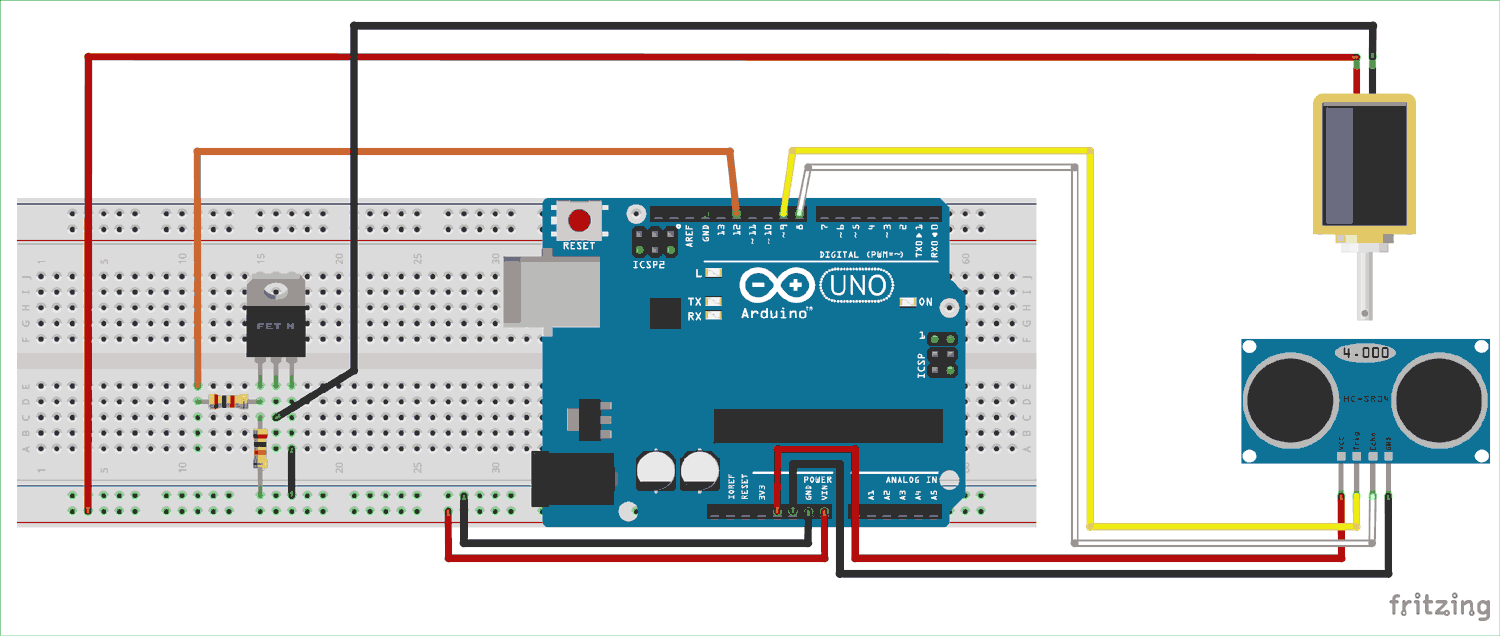
#define trigger 9  
#define echo 8  
#define LED 13  
#define MOSFET 12

float time=0,distance=0;

void setup()

{

Serial.begin(9600);

pinMode(trigger,OUTPUT);  
pinMode(echo,INPUT);

pinMode(LED,OUTPUT);

pinMode(MOSFET,OUTPUT);

delay(2000);  
}  
void loop()

{  
measure\_distance();

if(distance<10)  
{  
digitalWrite(LED,HIGH);digitalWrite(MOSFET,HIGH);

}

else

{

digitalWrite(LED,LOW);digitalWrite(MOSFET,LOW);

}

delay(500);  
}

void measure\_distance()

{  
digitalWrite(trigger,LOW);

delayMicroseconds(2);

digitalWrite(trigger,HIGH);

delayMicroseconds(10);

digitalWrite(trigger,LOW);

delayMicroseconds(2);  
time=pulseIn(echo,HIGH);  
distance=time\*340/20000;

}

<https://circuitdigest.com/microcontroller-projects/arduino-automatic-water-dispenser>