

**NAGARJUNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

# Department of Information Science and Engineering

Internet of Things Lab

LAB MANUAL

**V SEMESTER**

**Course Code: 21ISL55**

2023-24

COURSE TITLE: Internet of Things

COURSE CODE: 21ISL55

SEMESTER: V

BRANCH: INFORMATION SCIENCE AND ENGINEERING

FACULTY: Mrs. Nethra BR

LAB OBJECTIVES:

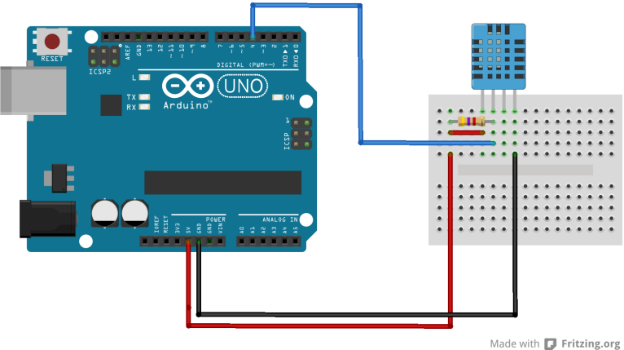
1. Introduce evolution of internet technology and need for IoT
2. Understand IoT architecture and various protocols and software.
3. Train the students to build IoT systems using sensors, single board computers and open source IoT platforms.

COURSE PREREQUISITES:

1. Familiarization with concept of IOT, Arduino and perform necessary software Installation.
2. Study of different operating systems for Arduino Understanding the process of OS installation on Arduino.

# LIST OF LAB PROGRAMS

|  |  |
| --- | --- |
| SL.NO | PROGRAM |
| **1.** | To know the Temperature and Humidity using DHT11 sensor. |
| **2.** | Write a code and build a model of traffic signaling using Arduino. |
| **3.** | Write a code and build a model to Controlling LED with Push Button. |
| **4.** | Write a code and build a model for ARDUINO BASED MOTION DETECTION USING PIR SENSOR |
| **5.** | Write a code and build a model to Interfacing Soil Moisture Sensor with Arduino. |
| **6.** | Write a code and build a model to check the Air Quality Monitoring using MQ-135 |
| **7.** | Write a code and build a model for Rain Detection System using Arduino and Rain Sensor. |
| **8.** | Write a code and build a model for Audio Frequency Detector. |
| **9.** | Write a code and build a model to implement ARDUINO PIANO |
| **10.** | Write a code and build a model for ARDUINO BASED SPEECH TO TEXT. |

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);

}

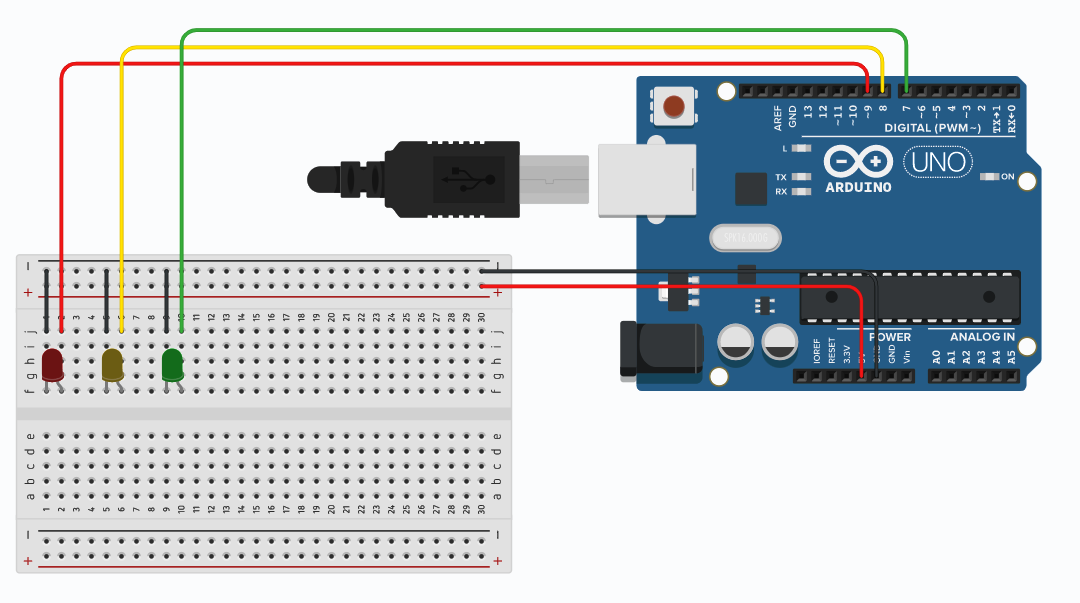
<https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621>

**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>

1. **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>

1. **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) {

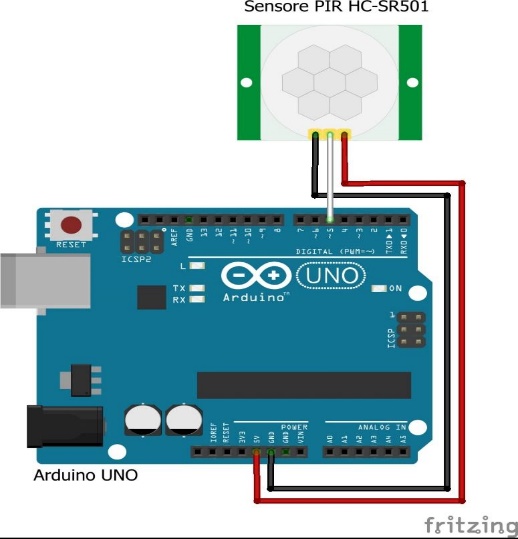
    Serial.println("Motion detected!");

    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>

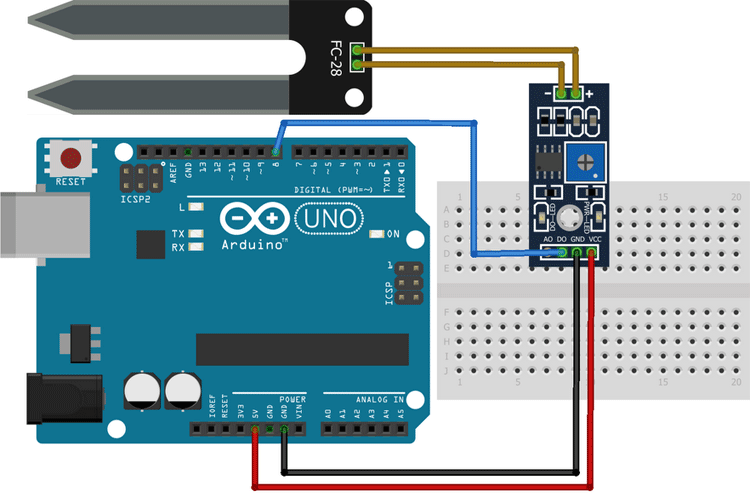
1. **Soil Moisture Sensor**

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

void setup() {

pinMode (A0, INPUT);

Serial.begin(9600);

}

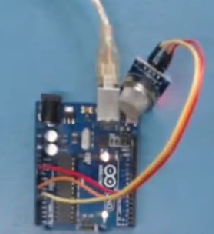
void loop() {

int my\_value=analogRead(A0);

delay(1000);

Serial.println(my\_value);

}



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

1. **Touch detection using Arduino**

#define ctsPin 2

// Pin for capactitive touch sensor

int ledPin = 13;

// pin for the LED

void setup()

{

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

pinMode(ctsPin, INPUT);

}

void loop()

{

int ctsValue = digitalRead(ctsPin);

if (ctsValue == HIGH)

{

digitalWrite(ledPin, HIGH);

Serial.println("TOUCHED");

}

else{

digitalWrite(ledPin,LOW);

Serial.println("not touched");

}

delay(500);

}

1. Ultrasonic Sonic sensor

int trig=10;

int echo=9;

long timeInMicro;

long distanceInCm;

void setup()

{

  Serial.begin(9600);

  pinMode(10,OUTPUT);

  pinMode(9,INPUT);

}

void loop()

{

 digitalWrite(trig,LOW);

 delayMicroseconds(2);

 digitalWrite(trig,HIGH);

 delayMicroseconds(10);

 digitalWrite(trig,LOW);

 timeInMicro= pulseIn(echo,HIGH);

 distanceInCm = ((timeInMicro/29)/2);

 Serial.println(distanceInCm);

 delay(1000);

}

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

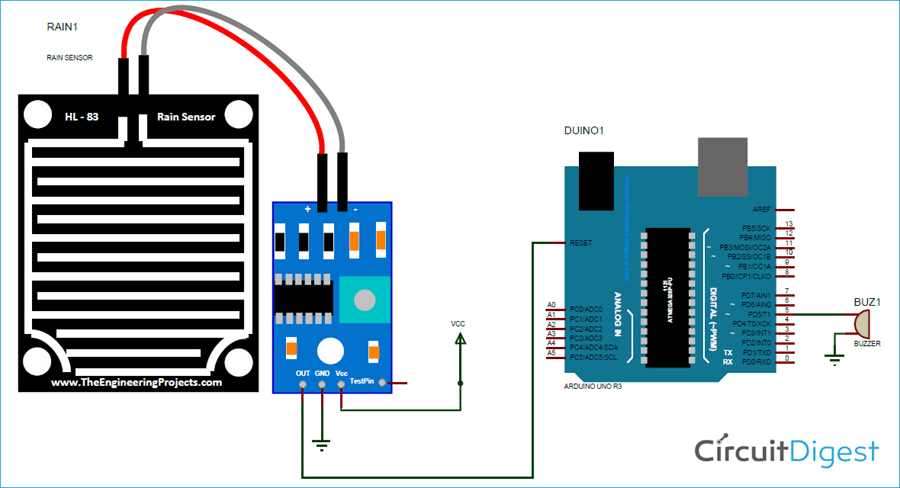
#define rainfall A0

#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>

1. **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()

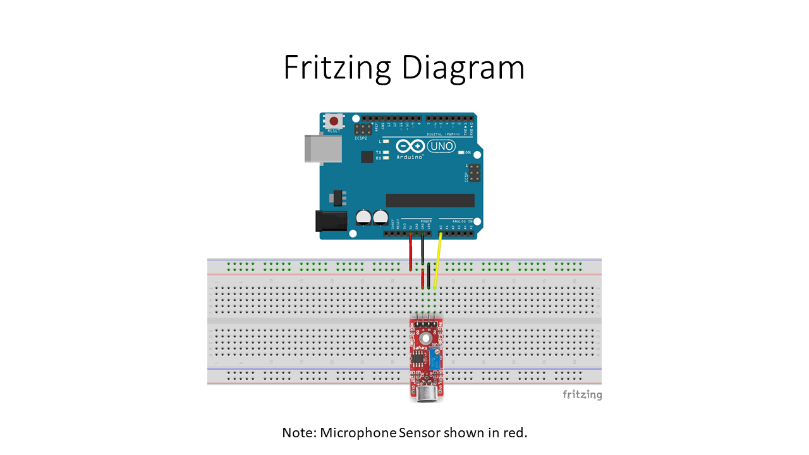
{

/\*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>

1. **ARDUINO PIANO**

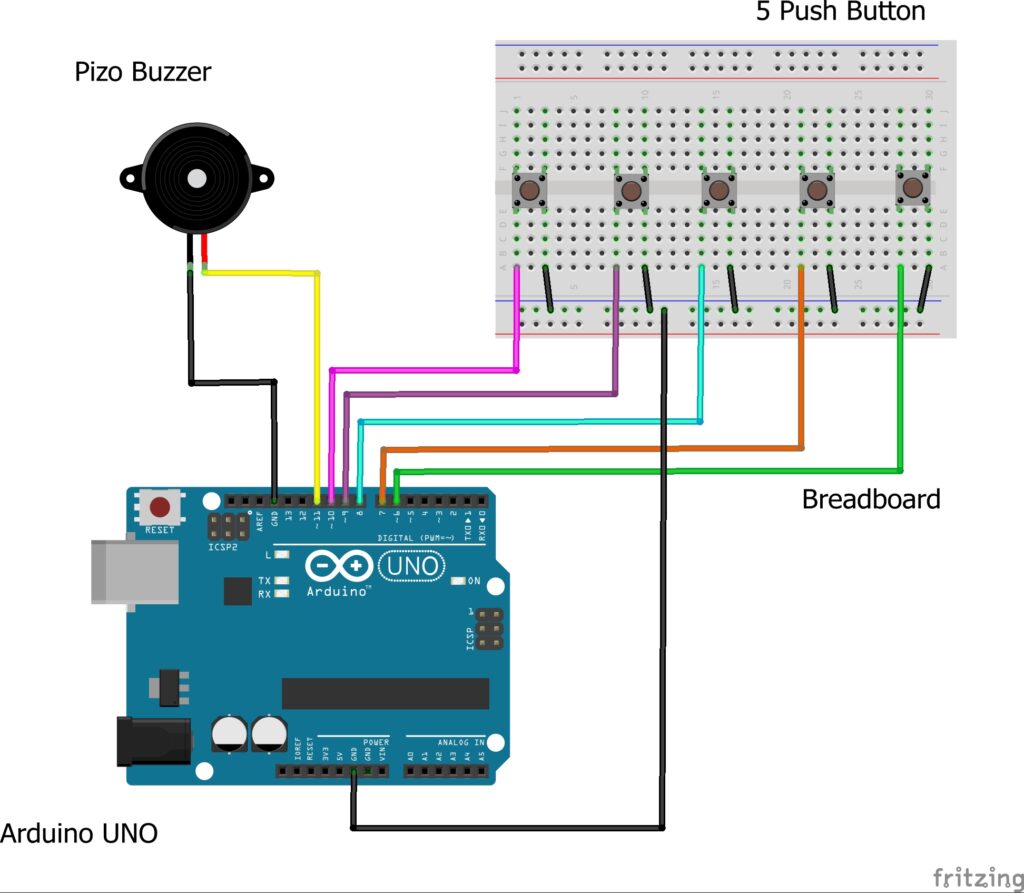
#define NOTE\_C 262

#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);

}

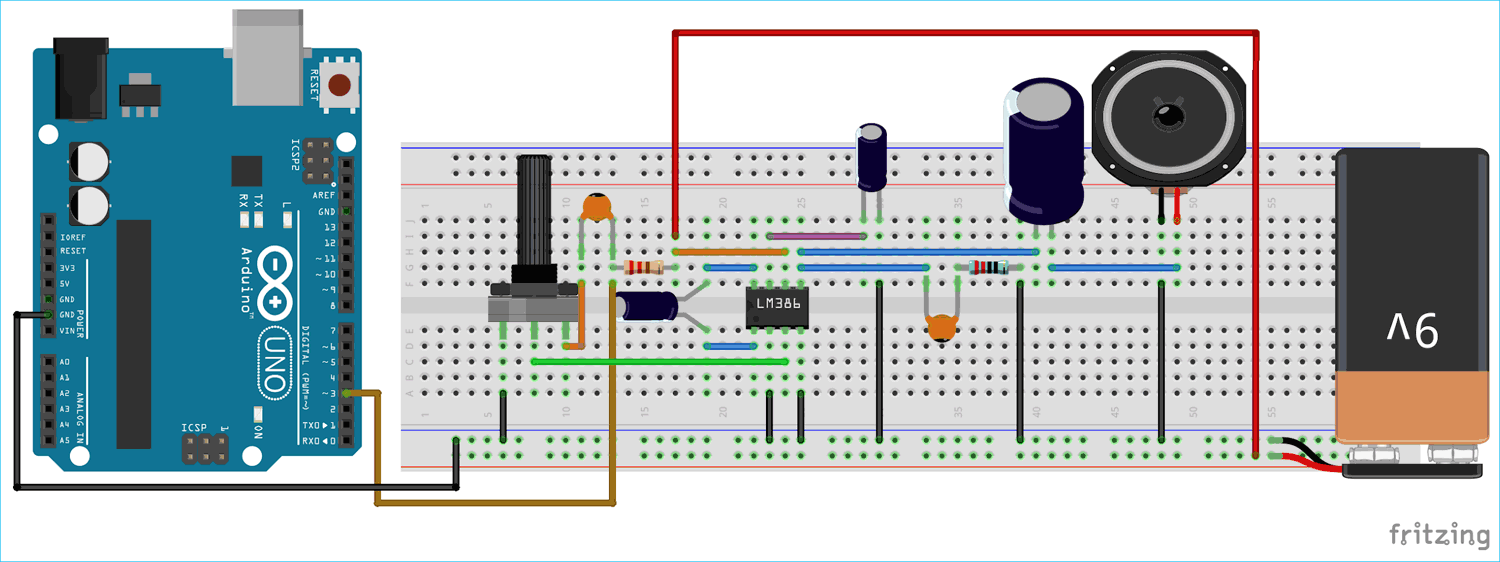
<https://techatronic.com/arduino-based-piano-using-push-buttons-arduino-mini-projects/>

**10. 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>

