PROJECT REPORT
TITLE:- SMART DOORBELL

❖ INTRODUCTION :-

- Internet of Things (IoT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment.
- Internet of Things is network of interconnected computing devices which are embedded in everyday objects, enabling them to send and receive data.
- In this project, I have built a SMART DOORBELL prototype that could replaces the old ding-dong doorbell. It is a smart doorbell because it sounds the name of person standing in front of the door to the person inside the house.
- Each of the family members has its own musical tune that would be use to give request for opening of the door.
- Family members can use a smartphone application providing separate programmable tunes so that those at inside house could hear that and identify which member is ringing the doorbell.
- This doorbell also consists an IR sensor which will be used by guest for giving a summon to the family member.
- A common musical tune was set for guests and strangers. If a family member doesn't carry the smartphone, then he/she can use the IR sensor to give the call.
- Moreover, this doorbell is also used at offices and clinics to give a call to the desired person without using the telephone.

❖ COMPONENTS :-

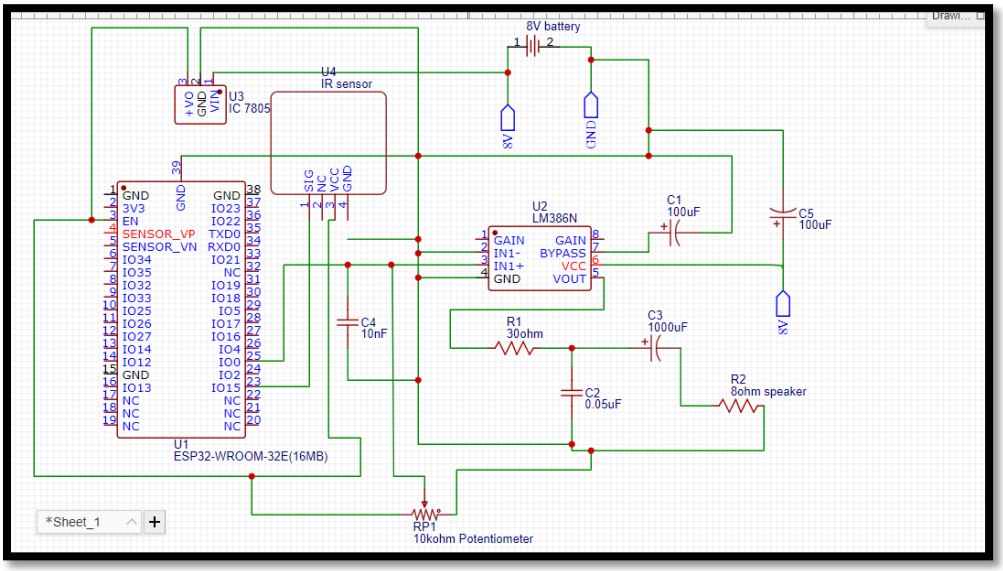
Sr. No	Component	Purpose
1	ESP32	IOT Gateway.
2	IR Sensor	To give a call by guests.
3	8 ohm speaker	To convert the electrical signal to the sound signal.
4	LM386 Audio Amplifier	It amplifies the audio coming out from the microcontroller and then feed to speaker.
5	8V Power Supply	It powers the doorbell
6	IC 7805	It provides the regulated 5V DC to the microcontroller
7	10kohm Potentiometer	To adjust the volume of the audio.
8	Smartphone	Used to give the user name as an input
9	Arduino IDE	To program the ESP-32
10	MIT App Developer	By using this tool, one can build their own mobile application.

❖ WORKING :-

- Firstly, Power up the system. After then, IOT gateway will start the server that will fulfill the request of the client.
- Connect your smartphone to the network created by ESP-32.
- Open the Android application that will used to give the request to the server.
- After giving the request in terms of the name of the person. The server then forwards it to the ESP-32. Microcontroller receives the message i.e. name of the person.
- Lastly, it will play the audio or tune programmed for the name of that person through an audio amplifier.

➢ If a family member doesn't carry the smartphone or in case guests arrive, then the/she or they can touch the IR sensor to sound the doorbell.

❖ SCHEMATIC:-



➢ Here, EN pin of ESP-32 used as VIN pin.

❖ NECESSARY TOOLS AND LIBRARIES:-

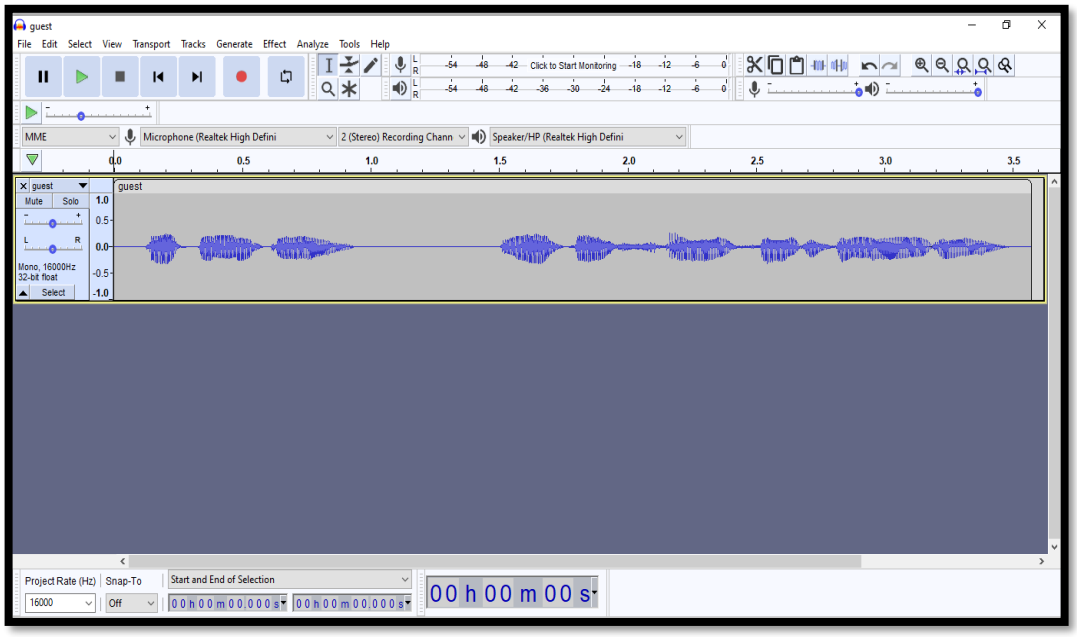
1. XT_AUDIO LIBRARY:-

➢ This library is used to play any audio from the ESP-32 microcontroller.

```
#include <WiFi.h>
#include <ArduinoJson.h>
#include <WebServer.h>
#include "XT_DAC_Audio.h"
#include "David.h"
#include "Lisa.h"
#include "guest.h"
#include "jack.h"

XT_DAC_Audio_Class DacAudio(25,0);  // D25 //


XT_Wav_Class David(david);
XT_Wav_Class Lisa(lisa);
XT_Wav_Class Jack(jack);
XT_Wav_Class Guest(guest);
```

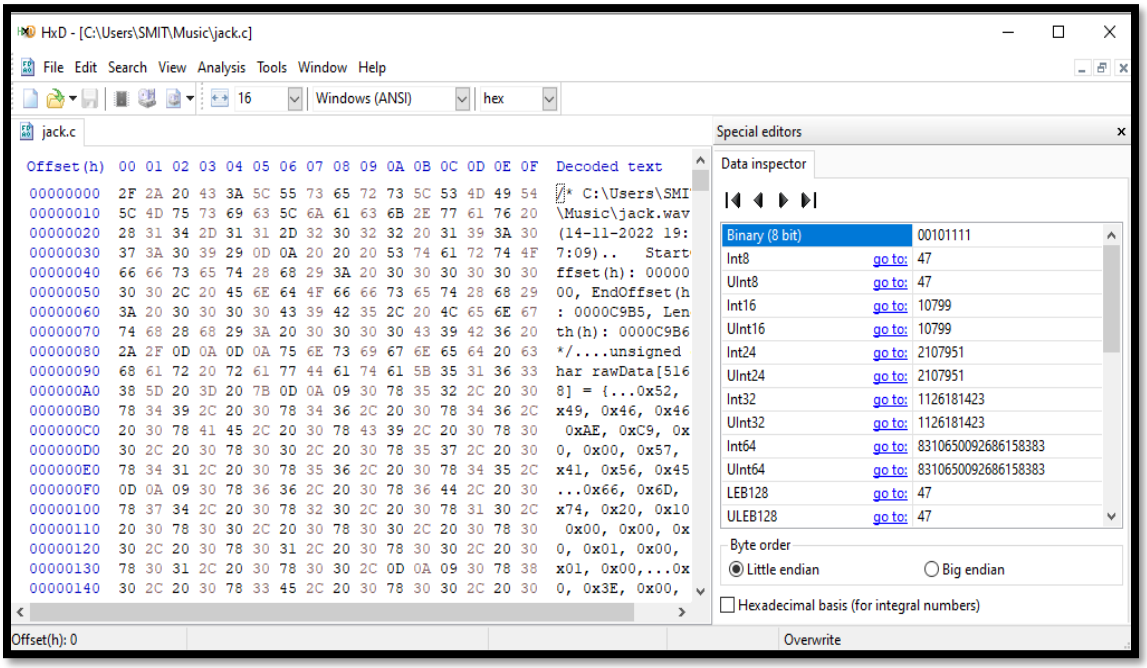
2. Audacity Tool:-

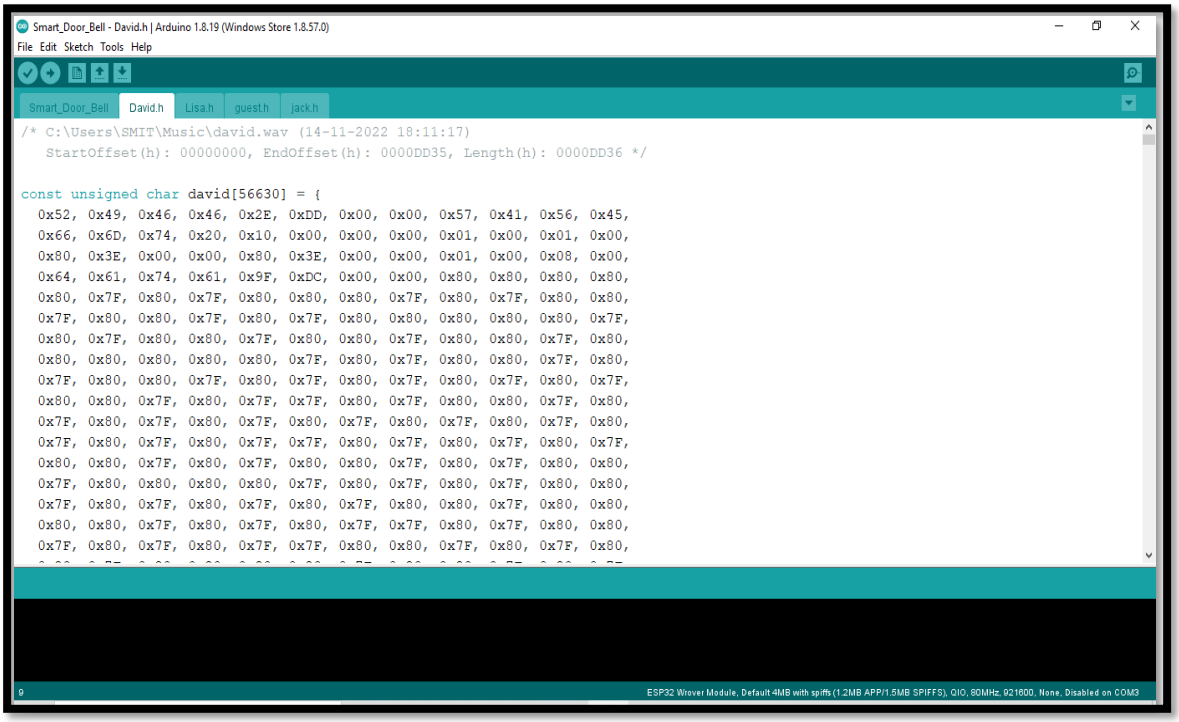
- This tool is used to convert the mp3 audio file to WAV file.
- After that resampling of audio file has done by using this tool.



3. Hex Editor: -

- This editor is used to generate the HEX file from the WAV file.
- Then, this HEX file will use as a header file after exporting as a C file.





4. WiFi and Webserver libraries:-

- **WiFi library is used to program the esp-32 in access point mode(AP) or in station mode.**
- **Here, we have programmed the esp-32 in access point mode to allow the client to connect esp-32.**
- **Whereas, webserver library is used to create a server that allows to fulfil the request of the client.**

❖ CODE:-

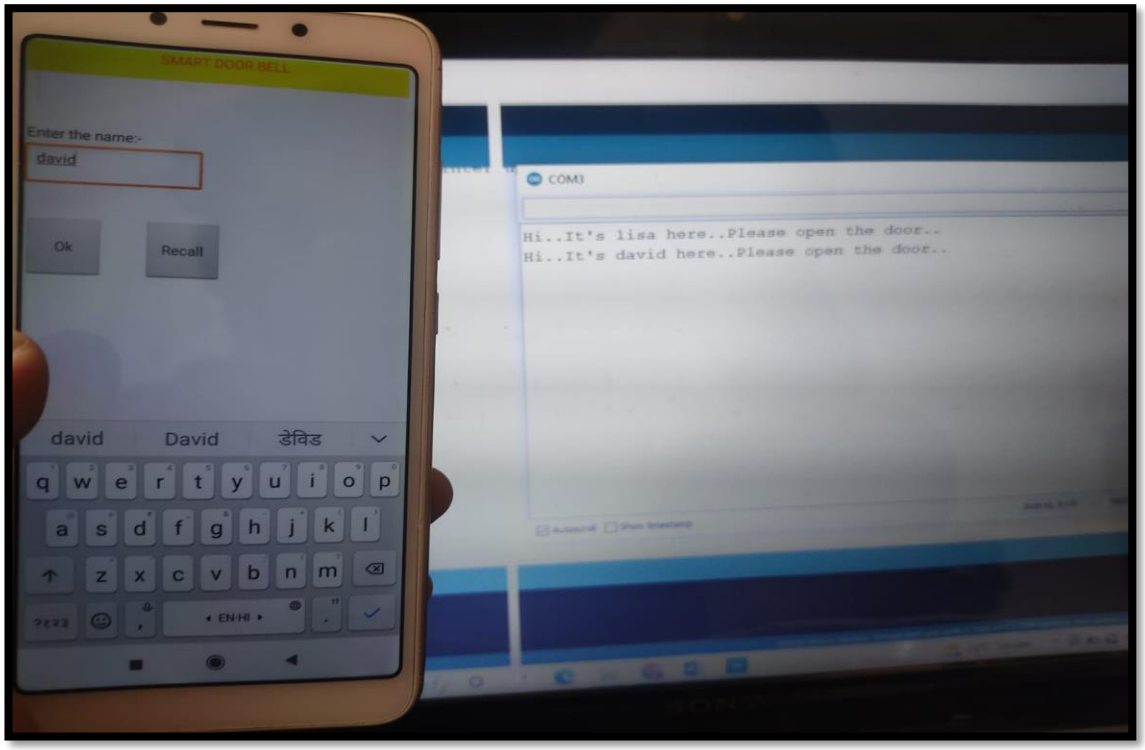
```
#include <WiFi.h>
#include <ArduinoJson.h>
#include <WebServer.h>
#include "XT_DAC_Audio.h"
#include "David.h"
#include "Lisa.h"
#include "guest.h"
#include "jack.h"

XT_DAC_Audio_Class DacAudio(25,0);  // D25 //
XT_Wav_Class David(david);
XT_Wav_Class Lisa(lisa);
XT_Wav_Class Jack(jack);
```

```
XT_Wav_Class Guest(guest);


WebServer server(80);
WiFiClient client;
const char* ssid = "HOME";
const char* password = "electronics";
const int interrupt_pin=4,test_pin=2;
volatile int curr_counter=0,prev_counter=0;
volatile long prev_millis=0;


IPAddress IP(192,168,4,4);
IPAddress GATEWAY(192,168,4,2);
IPAddress SUBNET(255,255,255,0);


void Data()
{
  String button_flag = server.arg("button");
  if(button_flag == "david")
  {
    Serial.println("Hi..It's david here..Please open the door..");
    DacAudio.Play(&David);
  }
  else if(button_flag == "lisa")
  {
    Serial.println("Hi..It's lisa here..Please open the door..");
    DacAudio.Play(&Lisa);
  }
  else if(button_flag == "lisa")
  {
    Serial.println("Hi..It's Jack  here..Please open the door..");
    DacAudio.Play(&Jack);
  }
  server.send(200,"text/html"," ");
}

void handle_client()
{
  server.send(200,"text/html","ROBOCON LDCE");
}

void Interrupt()
{
  if((millis() - prev_millis) > 1000)
  {
    curr_counter++;
```

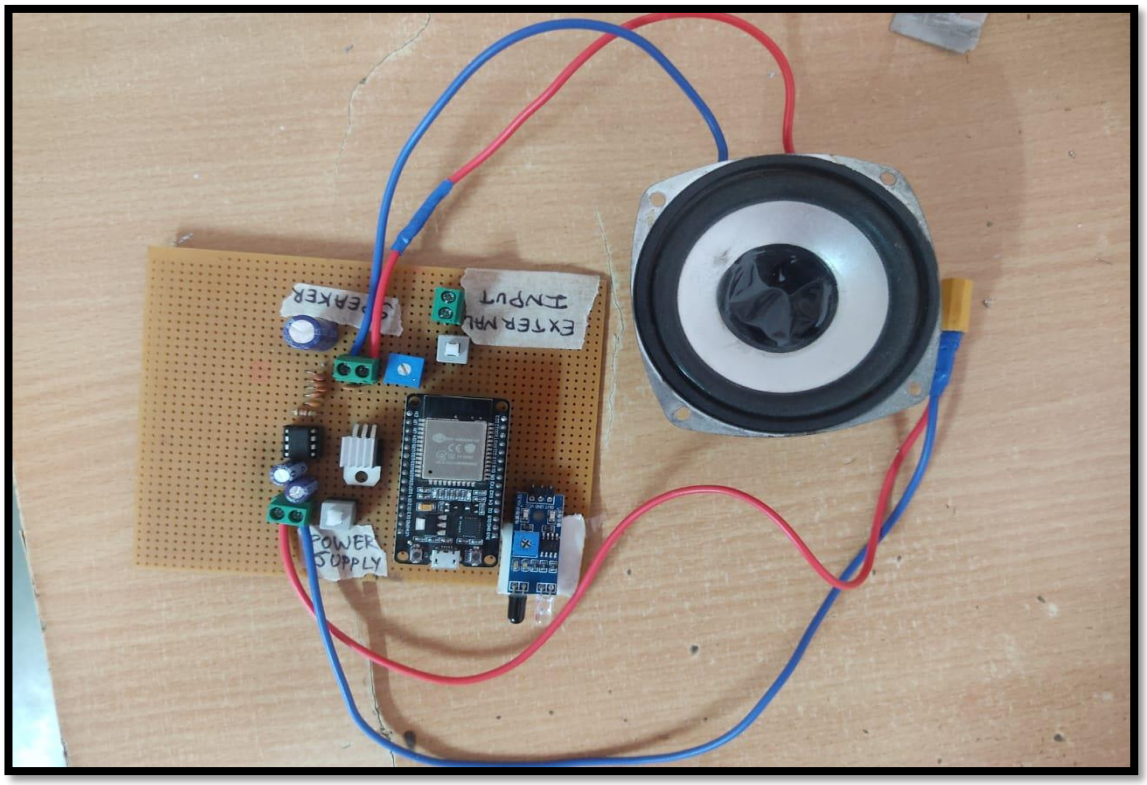
```
    prev_millis = millis();
  }

  if(curr_counter != prev_counter)
  {
    DacAudio.Play(&Guest);
    Serial.println("Audio playing");
    prev_counter = curr_counter;
  }


}

void setup()
{
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(interrupt_pin),Interrupt,FALLING);
  Guest.RepeatForever = false;
  Jack.RepeatForever = false;
  David.RepeatForever = false;
  Lisa.RepeatForever = false;
  WiFi.begin();
  WiFi.softAP(ssid,password);
  WiFi.softAPConfig(IP,GATEWAY,SUBNET);
  delay(1000);
  IPAddress Address = WiFi.softAPIP();
  Serial.println(Address);
  server.begin();
  server.on("/",handle_client);
  server.on("/data",Data);

}

void loop()
{
  server.handleClient();
  DacAudio.FillBuffer();
}
```

❖ OUTPUT :-

1. Software



2. Hardware :-



❖ REFERENCES: -

1. Audio DAC library for ESP32 :-

- https://www.xtronical.com/the-dacaudio-library-download-and-installation/

2. Hex Editor Tool : -

- https://mh-nexus.de/en/hxd/

3. Audacity Tool :-

- https://www.audacityteam.org/

4. MIT APP Developer (This tool will help to build mobile App) :-

- https://appinventor.mit.edu/