BAHÇEŞEHİR UNIVERSITY

Faculty of Engineering and Natural Sciences

Department of Electrical and Electronics Engineering

(Ömer Ayar)

(2101069)

Training Report

SUMMER TRAINING EVALUATION FORM

Training Evaluation:

1) Is the company that the student has chosen to do his/her training at, appropriate?

Yes ☐ No ☐

2) Is the score in the Training Register Form greater or equal to 2.5/4.00 ?

Yes ☐ No ☐

3) Has the company executive officer approved the training?

Yes ☐ No ☐

Student is: Succesful ☐ Unsuccesful ☐ in the training.

Report Evaluation:

1) Is the format and the content of the report appropriate?

Appropriate ☐ Not Appropriate ☐

2) Is the content of the report original and satisfactory?

Yes ☐ No ☐

Correction Offerings:

Student is: Successful ☐ Unsuccessful ☐ in the training and report.

Training Evaluation Committee:

Member1 **Member2**

Weekly Schedules: 1st Week

Date	Tasks Accomplished
08/07/2024	**After participating in the orientation program, I met with the team and we got information about Audio Test Device and had conversations.**
09/07/2024	**The company's production factory in Tuzla had a sample of the device we wanted to make. We went to Tuzla to examine the device and we examined in detail the appearance of the device, the operations it performs and how it performs these operations. As a result of the examination, we decided what kind of components we would use.**
10/07/2024	**The alternatives of the components we will use were analyzed and compared in terms of price, performance and function, then the data was exported to an Excel file. Discussed the algorithms needed to test microphones and speakers correctly.**
11/07/2024	**Components were received. The Arduino IDE program was installed to load the drivers of the ESP32 and the written code. Possible connections of microphones and speakers were investigated.**
12/07/2024	**Programs were written and run to understand and test the logic of the processor. Researched libraries to meet the requirements for the purpose.**
	Executive Officer

Weekly Schedules: 2nd Week

Date	Tasks Accomplished
15/07/2024	**No work was carried out today due to the public holiday.**
16/07/2024	**A microphone was connected to the ESP32 and analog data reading operations were performed. In order to operate the two microphones sequentially, pin arrangements were made for different outputs and the microphones were programmed to be recorded back-to-back by coding accordingly.**
17/07/2024	**Due to the inability to correctly initialize the microphone, different algorithms were tried and various methods were worked on.**
18/07/2024	**Various studies were carried out to store the data obtained from the microphone in the correct format through modulation and conversion processes.**
19/07/2024	**Since the microphone could not be operated with the algorithms tried, an alternative microphone was researched and the appropriate microphone was ordered.**
	Executive Officer

Weekly Schedules: 3rd Week

Date	Tasks Accomplished
22/07/2024	**A new digital microphone was received, libraries and setup were completed, and the resulting graphs were analyzed.**
23/07/2024	**Studies on spectrum analysis of audio data were started using the ArduinoFFT library. The working principle of the FFT algorithm and how to obtain the frequency components of audio signals were learned.**
24/07/2024	**Studies on the filtering of microphone data and the calculation of values in dB. Various filtering techniques were investigated and applied.**
25/07/2024	**Work was carried out on recording the data obtained from the microphone in WAV format and coding was performed to store the data in the processor using the SPIFFS method.**
26/07/2024	**The SD card was installed and the necessary connections were made. Necessary arrangements were made to save the recorded 7 seconds of microphone data to the SD card.**
	Executive Officer

Weekly Schedules: 4th Week

Date	Tasks Accomplished
29/07/2024	**How to connect the LM386, speaker and ESP32 together was investigated and sample codes for their operation were examined.**
30/07/2024	**For the microphone tests, the sounds that should be emitted from the loudspeaker were researched and studies were carried out on how to obtain these sounds.**
31/07/2024	**Various sounds were created using the wave library with Python. The working principle of the library was examined.**
1/08/2024	**The sounds to be used were expressed mathematically and converted to WAV format with the Python wave library.**
2/08/2024	**The generated sound was uploaded to the ESP32's SD card for playback on the loudspeaker and experiments were conducted.**
	Executive Officer

Weekly Schedules: 5th Week

Date	Tasks Accomplished
5/08/2024	**The second microphone was connected and code was edited to enable the two microphones to record separately at different times. The tests of the edited codes were carried out.**
6/08/2024	**All components were connected and the microphone and speaker codes were merged. Sequencing and duration adjustments were made as needed and the system started to work in a certain order.**
7/08/2024	**The pin arrangements of the circuit were made and the circuit diagram was drawn.**
8/08/2024	**In accordance with the drawn circuit diagram, the connections of all components in the circuit were carefully soldered. During the soldering process, it was checked that the components were placed in the right places and a secure connection was ensured.**
9/08/2024	**The soldered circuit was checked for short circuits. Then, work began on creating an interface for the use of the system.**
	Executive Officer

Weekly Schedules: 6th Week

Date	Tasks Accomplished
12/08/2024	**We worked on setting the ESP32 as a WiFi access point, allowing multiple users to connect at the same time.A basic interface was developed using HTML and JavaScript to visualize the microphone data on the web server created on the ESP32.**
13/08/2024	**WaveformJS library was used to visualize the sound waveform in the web interface.**
14/08/2024	**Added a button on the HTML to start audio recording and a countdown timer to show the time remaining before the page refreshes.**
15/08/2024	**'Play', 'Download' and 'Delete' buttons have been added to the interface to observe and control microphones and test sounds."**
16/08/2024	**The performance of the ESP32 web server was optimized and file redirects on ESP32 were updated, enabling more efficient access to local server files. In addition, simultaneous spectrum analysis with audio was performed to visualize the data.**
	Executive Officer

Weekly Schedules: 7th Week

Date	Tasks Accomplished
19/08/2024	**Necessary documents and documentation were prepared for reporting the project process. The project presentation was completed and presented to the relevant people. Feedbacks were received after the presentation and an overall evaluation of the project was made.**
	Executive Officer

Contents

ABSTRACT

In this summer course, a comprehensive audio processing and visualization project was developed using the ESP32 microcontroller, microphone, speaker and various sensors. First, analog audio data was collected from the microphone and spectrum analysis was performed using Fast Fourier Transform (FFT) algorithm. Then, by configuring the ESP32 as a web server, users were able to visualize this data live on the web interface over the local network.

In the project, functions such as playing audio through the speaker and reading audio files using an SD card module were also successfully implemented. Furthermore, using the ESP32 as a WiFi Access Point (AP) allowed multiple users to simultaneously monitor sound waves and access recorded files. Consecutive audio recordings made with two microphones were stored in different files, enabling analysis of multiple sources of audio data.

Thanks to the web interface developed with HTML, JavaScript and the WaveformJS library, users were able to manage files and analyze sound waveforms in detail. Throughout this process, the project was optimized by solving the technical problems encountered and a web-based audio processing system was successfully completed.

1. INTRODUCTION

This internship report covers my summer training in audio processing and visualization and the projects I developed during this period. My interest in electronics and embedded systems led me to learn more about microcontrollers and audio technology. In this project, I focused on audio processing, data acquisition, analysis and visualization with the ESP32 microcontroller. During the internship, I learned about hardware and software integration as well as how to present audio data in a user-friendly interface with web technologies.

The ESP32 is a low-cost and versatile microcontroller, making it an ideal platform for wireless communication and data processing projects. It can be used in a variety of areas such as voice processing, digital data analysis and real-time applications, making it my main area of study during this internship. The project was chosen specifically because of my desire to gain practical experience in audio processing with microcontrollers and the opportunity to apply my theoretical knowledge in this field on a real project.

From the first day of the internship, I have been working on the accurate collection, filtering and visualization of microphone data, while at the same time aiming to deliver this data to users over a wireless network. The project involves a comprehensive process of collecting, recording, analyzing and presenting audio signals using microphones and speakers through a web-based platform. In this process, various audio processing algorithms, data storage methods, wireless communication protocols and web technologies were used.

The biggest factor in choosing this company was the opportunity to develop real-world applications in audio and signal processing. In addition, the ESP32 microcontroller and audio processing hardware used in this project can be applied in many different fields, which further increased my interest in the project. I gained experience in many areas such as microcontroller programming, wireless network setup and web-based data visualization. This internship also allowed me to explore the potential of microcontrollers for real-time data processing and interacting with users.

This report provides a detailed description of the work I carried out during the project, the technical challenges encountered and how I overcame them. First, I describe how the audio signals were collected and processed by the microcontroller. Then, I describe the methods used to transmit this data over the wireless network and visualize it on the web interface. I also discuss the technical details of data logging, storage and presentation to the users. Finally, I present the achievements at the end of this process and suggestions on how to improve this project in the future.

At the end of the internship, I gained extensive experience in both theoretical and practical aspects of microcontroller-based audio processing and web-based visualization, which I intend to use in more complex projects in the future.

2. ABOUT THE COMPANY

AGM Technology is a manufacturing company operating in the field of consumer electronics products, equipped with advanced technologies. Its headquarters is located at Çobançeşme Mahallesi, Kımız Sokak, No:16 / 34196 in the Bahçelievler district of Istanbul. AGM Technology specializes in the production of smartphones and has high quality standards and innovative production processes in the sector. The organizational structure of the company is determined by a corporate hierarchy. The CEO is at the top management level and the main departments reporting to the CEO are Finance, Production, Marketing, Sales, Human Resources and R&D.

The Finance department carries out the company's budgeting, accounting, cost analysis and financial reporting, while the Production department organizes the production processes, quality control and logistics activities at the facilities. The R&D department works on product development, implementation of new technologies and production optimization. The Marketing and Sales department determines the marketing strategies of the company's products and services and carries out sales activities, while the Human Resources department is responsible for recruitment, training, personnel management and employee satisfaction. In addition, the factory manager and the technical team overseeing operations at the production facilities are also an important part of the organization. Overall, the company's organizational structure aims to maintain harmony between both the head office and the production facilities.

Together with its factory and head office, AGM Technology employs between 201-500 employees in total. While most of the employees at the factory work in areas such as production, quality control and product assembly, the head office employs professionals specialized in product development, marketing, sales, customer service, finance, human resources and R&D. R&D engineers follow new product designs and technological developments, while the marketing team conducts market research and develops strategic sales plans. The human resources department organizes training programs and manages recruitment processes to increase employee productivity. The personnel in the factory are responsible for ensuring the efficiency of the production processes and supervising the compliance of the products with quality standards.

AGM Technology specializes in the production of smartphones, and its product portfolio includes models such as Oppo Reno 11 F 5G, Oppo A38 - CPH2579 and OPPO A60 - CPH 2631. To manufacture these products, the company uses modern facilities equipped with the most advanced technology. The production lines are equipped with fully and semi-automatic machines, ensuring high speed and high quality products. The production process is carried out on SMT (Surface Mount Technology) and FATP (Final Assembly and Test Procedure) lines. These lines ensure that the electronic components of the products are precisely assembled and tested. At the same time, the production facilities have a special section for R&D activities. In this department,

new technologies are discovered, future products are developed and innovation is supported. AGM Technology adopts production methods that meet international standards to improve product quality and provide a competitive advantage in the market. The company's globally experienced team closely follows technological developments through international collaborations and integrates these developments into its products. In this way, it increases customer satisfaction and the competitiveness of its products in the global market.

3. PRACTICAL TRAINING

3.1 Introduction to Key Components

ESP32 is a high performance and low energy consumption microcontroller platform developed by Espressif Systems. ESP32, which integrates WiFi and Bluetooth features, is used in many applications, especially IoT (Internet of Things) projects. Featuring a dual-core 32-bit Tensilica LX6 processor, ESP32 offers 2.4 GHz WiFi connectivity and BLE/Classic Bluetooth support. Thanks to these features, it is frequently preferred in projects requiring wireless communication, data transmission and remote control. ESP32 also supports a wide range of sensors and devices

with input/output interfaces such as multiple GPIO pins, ADC (Analog Digital Converter), PWM, SPI, I2C. The ESP32 can be used in both hobbyist projects and professional applications, providing flexibility to developers with its open source software support and strong community.

Figure 3.1.1 ESP32 pinout

1.The pin structure of the ESP32 supports various communication protocols such as digital and analog input/output, UART, SPI, I2C.(Figure 3.1.1)

2.GPIO pins are used for general purpose input and output operations and can be assigned to different modes.(Figure 3.1.1)

The INMP441 is a digital MEMS microphone with low power consumption and high sensitivity, making it a common component in sound detection applications. It includes a built-in analog-to-digital converter (ADC) and can convert audio signals into digital format and transfer them directly to microcontrollers. Working with the I2S (Inter-IC Sound) interface, INMP441 offers high quality audio data transmission and can be used in projects such as noise cancellation,

voice recording and voice command systems.

Pin functions are as follows: VDD is the microphone's 3.3V power input; GND is the ground connection. SD (Serial Data) is the data output where digital audio data is transmitted. SCK (Serial Clock) synchronizes the data transmission by carrying the I2S clock signal. WS (Word Select) signals the start of the data frame by determining the left/right channel selection. With these pins, the INMP441 transmits audio signals digitally.

Figure 3.1.2 INMP441

1.The INMP441 is a high-quality digital microphone that offers low power consumption and high sound quality.(Figure 3.1.2)
2.With its I2S interface, it transmits audio data directly in digital format, allowing easy integration into microcontroller projects.(Figure 3.1.2)

The LM386 is a low-voltage audio amplifier and is often used in portable audio projects. Operating from 4 to 12V, this integrated circuit is especially ideal for small speakers, amplifying audio signals for higher sound output. The LM386 can be used in combination with external

components, the gain is adjustable and offers clean audio amplification with low distortion.

Figure 3.1.3 LM386

1.The LM386 is an integrated amplifier widely used to provide audio amplification with low power consumption.(Figure 3.1.3)

2.With its small size and simple circuit design, it is preferred in various applications, especially in portable audio systems.(Figure 3.1.3)

A loudspeaker is a device that converts electrical signals into sound waves. The principle of operation is based on the electric current vibrating the speaker diaphragm to create sound.Speakers usually have two connection cables, plus (+) and minus (-).

Figure 3.1.4 Loudspeaker

1.A loudspeaker is a transducer that transmits sound waves into the air, allowing sound to be heard.(Figure 3.1.4)

2.Different sizes and types are used to produce sound in various frequency ranges, making them suitable for a wide range of uses, from music systems to voice communication applications.(Figure 3.1.4)

Micro SD card module is a component used for data storage and transfer. It is often used in embedded system projects, especially in combination with microcontrollers. This module allows users to store and easily access various types of data such as audio, image or text files. Pin functions of the micro SD card module include VCC (power input, usually 3.3V or 5V), GND (ground connection), MISO (Master In Slave Out, the pin that transmits data from the microcontroller to the module), MOSI (Master Out Slave In, the pin that transmits data from the module to the microcontroller), SCK (Serial Clock, the pin that provides the clock signal for data transmission) and CS (Chip Select, the pin that enables the SD card).

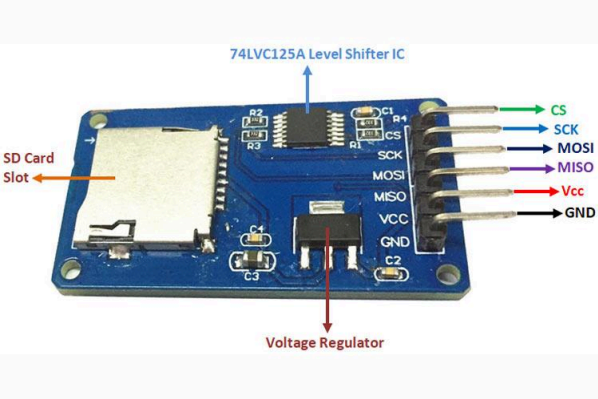

Figure 3.1.4 MicroSD Module

1.The MicroSD module facilitates data saving and reading operations using microSD cards to provide storage space.(Figure 3.1.4)

2.Thanks to its small size and low cost, it is a popular component for data management in various microcontroller projects.(Figure 3.1.4)

3.2 Setting Up the Required Environment

Before starting this installation procedure, you need to have Arduino IDE installed on your computer.

1.In your Arduino IDE, go to **File**> **Preferences**

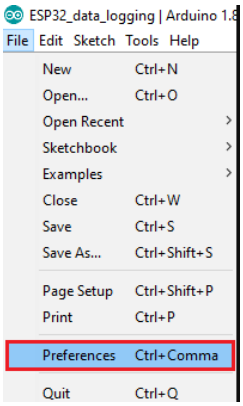

Figure 3.2.1 ArdunioIDE File Menu

1.The File Menu of the Arduino IDE provides you with the necessary options to manage your projects, create new files and open existing projects.(Figure 3.2.1)

2.This menu provides access to basic functions such as saving and exporting your code and viewing sample projects.(Figure 3.2.1)

2.In the "Additional Board Manager URLs" section, we need to enter the link for esp32 and click the ok button

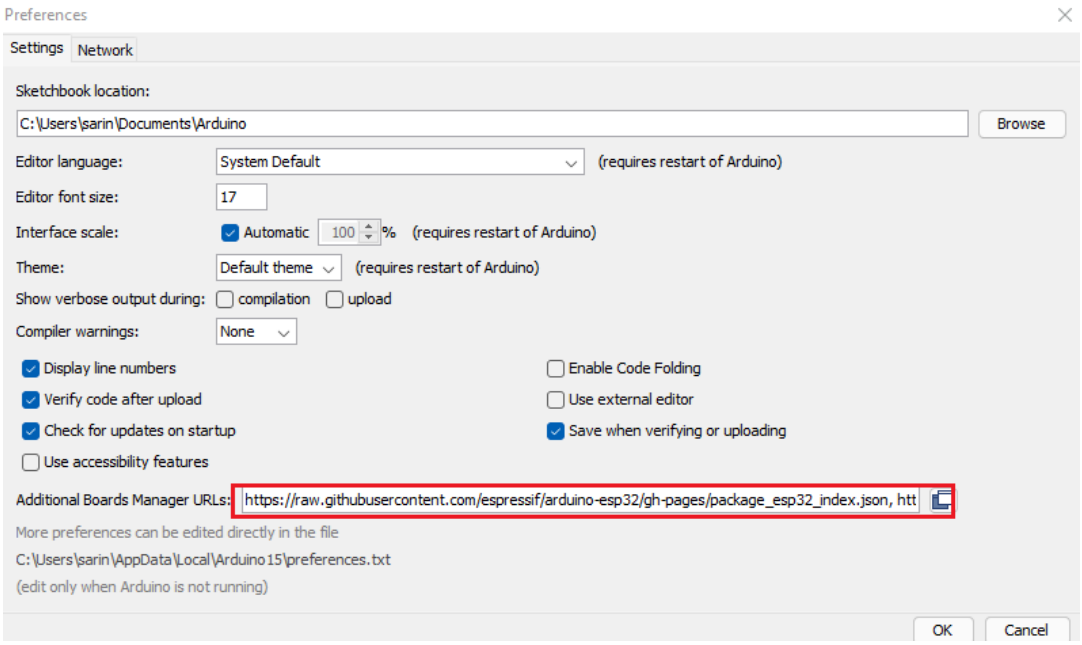

Figure 3.2.2 ArdunioIDE Preferences

1.The Arduino IDE Preferences menu allows customizing the software experience by personalizing user settings.(Figure 3.2.2)

2.In this section, various options such as theme, language and library settings can be edited.(Figure 3.2.2)

3.Open the Boards Manager. Go to **Tools** > **Board** > **Boards Manager…**

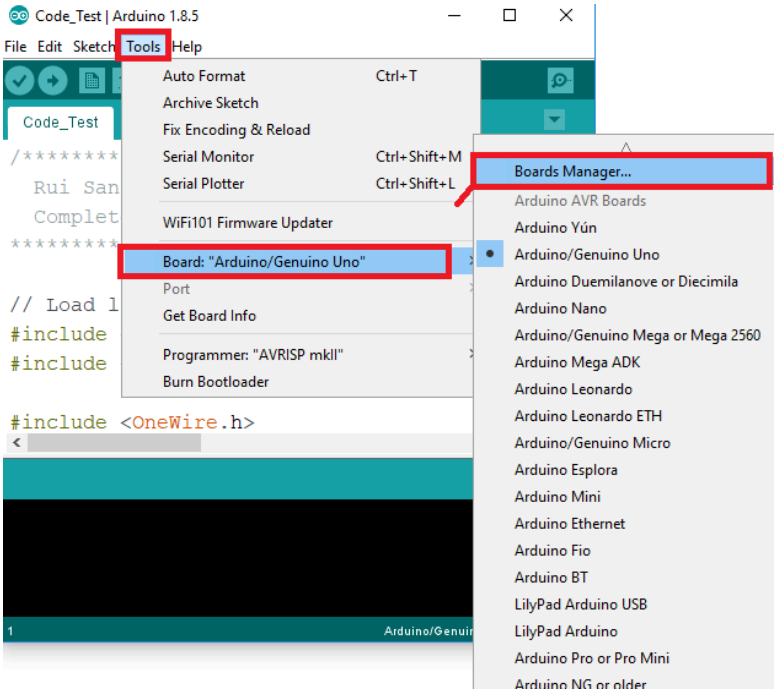

Figure 3.2.3 ArdunioIDE Tools

1.The Arduino IDE Tools menu is used to manage the settings of connected devices and access different board programming options.(Figure 3.2.3)

2.This menu includes important functions such as library manager, serial monitor and board selection, thus facilitating the development process.(Figure 3.2.3)

4.Search for **ESP32** and press install button for the "**ESP32 by Espressif Systems**":

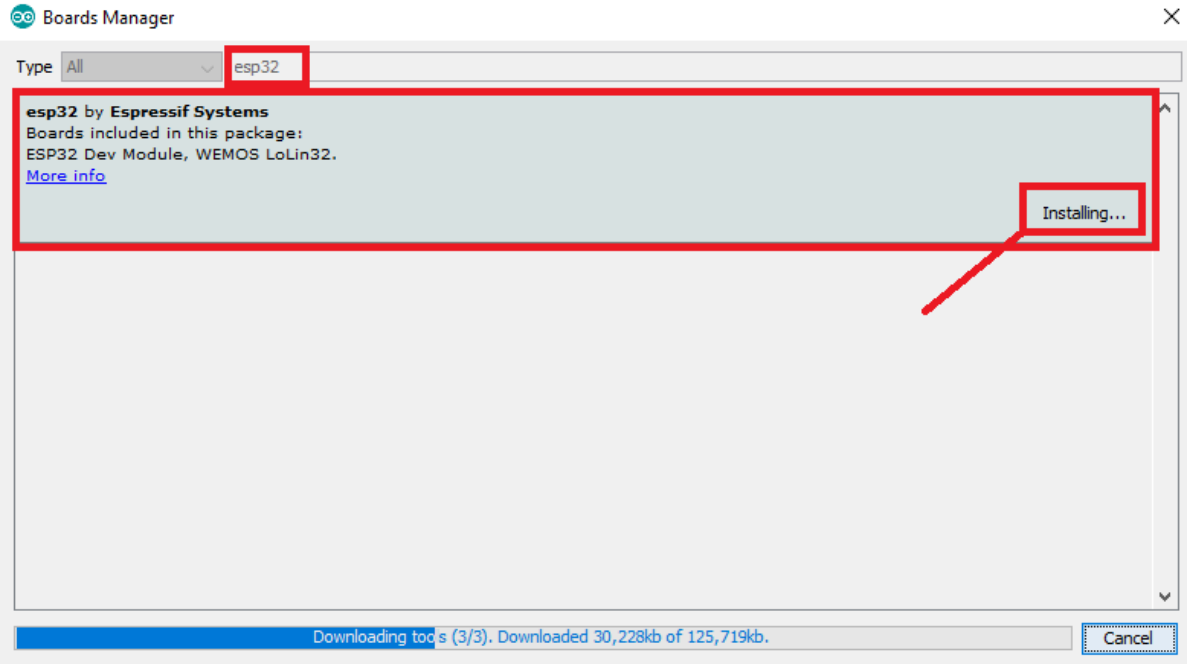

Figure 3.2.4 ArdunioIDE Board Manager

1.Arduino IDE Board Manager is used to install and update different types of boards, so you can support various microcontrollers in your projects.(Figure 3.2.4)

2.With its user-friendly interface, it is very easy to add new boards or manage versions of existing boards.(Figure 3.2.4)

And after all the downloads are complete, the "CP210x USB to UART Bridge VCP Drivers" must be installed for the computer to recognize esp32, after which esp32 will work on our computer.

3.3 Connections of Key Components

I collected audio data using two INMP441 digital microphones with the ESP32 microcontroller. For the first microphone, I connected the SDI pin to the GPIO 34 pin of the ESP32, the SCK pin to the GPIO 18 pin, the WS (frame) pin to the GPIO 25 pin and the CLK pin to the GPIO 26 pin. For the second microphone, I connected SDI to GPIO 32 pin, SCK to GPIO 23 pin, WS to GPIO 22 pin and CLK to GPIO 21 pin. After collecting the audio data, I connected the LM386 amplifier to GPIO 27 pin of the ESP32 and routed the audio output to the speaker. For the power

supply of the amplifier, I connected the VCC pin to 5V and the GND pin to GND. I also used a micro SD card module to store data in the project. I connected the CS pin of the SD card module to GPIO 5, the SCK pin to GPIO 18, the MOSI pin to GPIO 23 and the MISO pin to GPIO 19. Thus, I created the necessary infrastructure to save the audio data to the SD card and analyze it later.

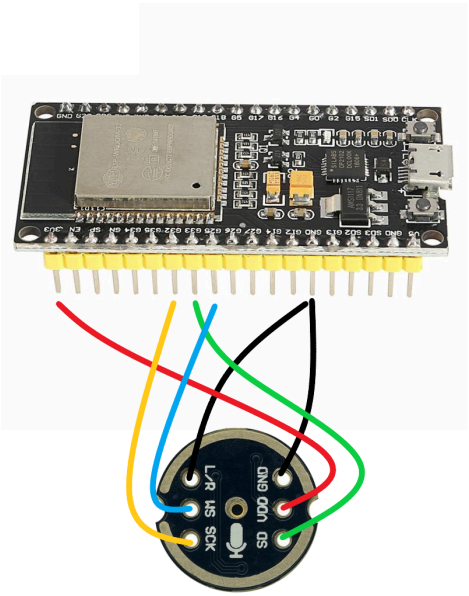

Figure 3.3.1 Example Connection ESP32 with INMP441

1.The connection between the ESP32 and INMP441 enables audio recording by transmitting the microphone's digital audio data directly to the microcontroller.(Figure 3.3.1)

2.This connection must be made with the correct supply and ground connections with appropriate pin assignments.(Figure 3.3.1)

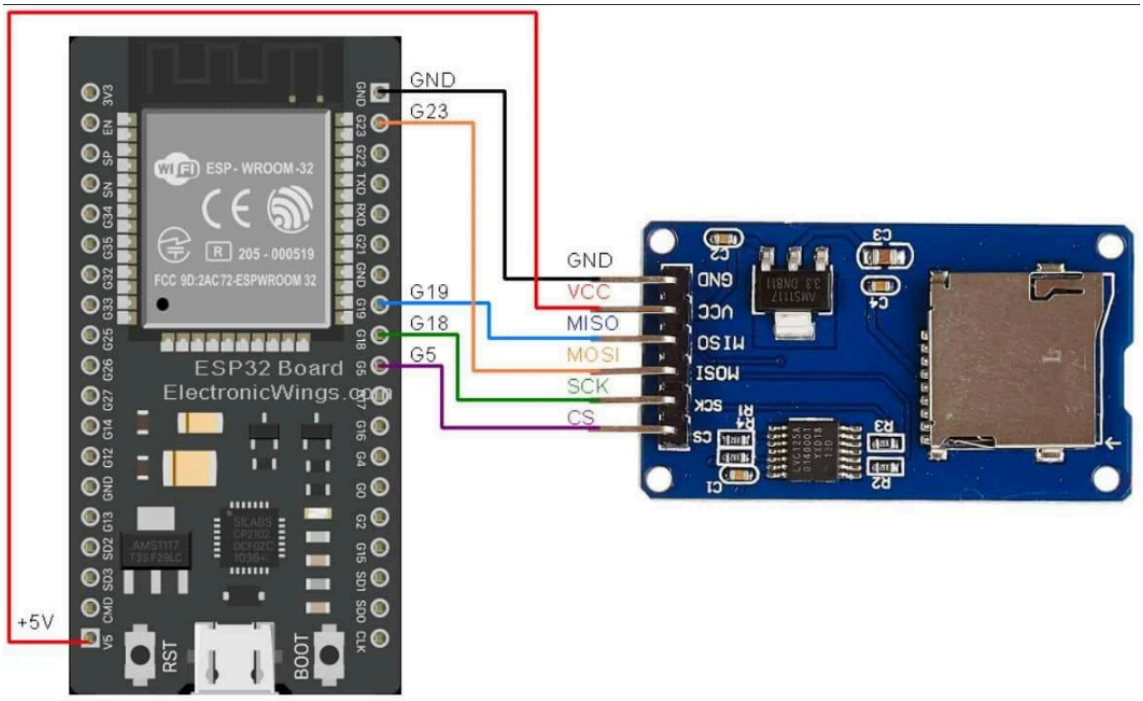

Figure 3.3.2 Example Connection ESP32 with MICROSD module

1.The microSD module connection with the ESP32 provides the SD card communication required for saving and reading data.(Figure 3.3.2)

2.With the right pin assignments and connections, it offers a wide range of storage options for data management projects.(Figure 3.3.2)

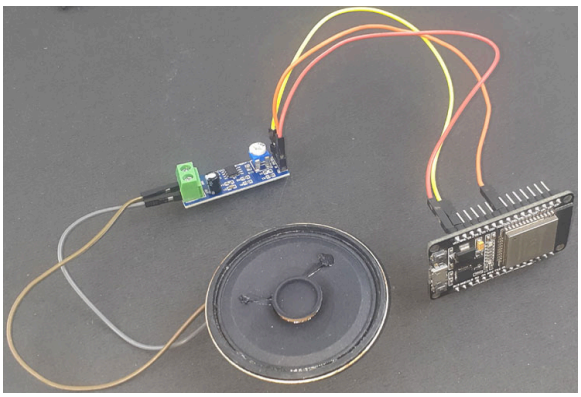

Figure 3.3.3 Example Connection ESP32 with LM386 and Speaker

1.With the ESP32, the LM386 and speaker connection allows digital audio signals to be amplified to produce a dynamic audio output.(Figure 3.3.3)

2.This structure is ideal for improving sound quality and providing higher volume levels in audio projects.(Figure 3.3.3)

3.4 Code Explanation

This project aims to record audio using two INMP441 microphones connected to an ESP32 microcontroller, save the recorded audio data in WAV format on a micro SD card, and provide users with the ability to manage and visualize the recordings through a Wi-Fi access point. The process is handled by a web server running on the ESP32, and each component of the system is integrated with different algorithms.

Once the project starts, the ESP32 is configured as a Wi-Fi access point (SSID: "ESP32_AP"). This is achieved using the WiFi.softAP() function, allowing users to connect wirelessly to the ESP32. A web server is created using the AsyncWebServer library, where multiple routes are added for various tasks like starting a recording, downloading, or deleting audio files. For instance, the /start_recording route triggers the recording process, while the /download_espmic1 and /download_espmic2 routes allow users to download the recorded files. To delete the recorded files from the server, the /delete_espmic1 and /delete_espmic2 routes are used.

For the ESP32 to interface with the SD card, the SD card module is initialized over SPI. This is done using the SD.begin() function. If the SD card module is successfully initialized, a "SD card initialized" message is printed to the serial monitor. If initialization fails, an error message informs the user, and the program halts further operations. During file operations, separate WAV files are created for each microphone (/espmic1.wav and /espmic2.wav). The audio data is recorded to these files, and when files need to be deleted, the deleteFile() function is used.

The digital audio data from the microphones is transmitted to the ESP32 via the I2S protocol. Each microphone is connected to I2S ports to send data, and this data is read using the I2S protocol. The first microphone collects data through the I2S_NUM_0 port, while the second microphone operates on the I2S_NUM_1 port. Drivers for each port are installed using the i2s_driver_install() function, and then the pins assigned to the microphones are configured using i2s_setpin(). The recordMicrophones() function starts the recording process for each microphone sequentially. The first microphone records for 7 seconds, and once the recording is complete, there is a 3-second wait before the second microphone starts recording for another 7 seconds. The recorded data is written to the SD card in WAV format.

The writeWAVHeader() function is critical for ensuring that the recorded audio data is correctly stored as a WAV file by adding the necessary header information. This header includes important details such as the file size, sample rate, number of channels, and bit depth. When the recording is complete, this information is included in the file to ensure it is a valid WAV file.

After the recording is finished, a beep sound is generated from the speaker at a frequency of 2 kHz lasting for 7 seconds. This sound is generated using the tone() function, and the beep is stopped with the noTone() function when the recording ends.

Users can interact with the system via the web server running on the ESP32. For example, users can initiate a recording session by accessing the /start_recording route. When the recordingStarted flag is set to true on the server, the recording begins, and data from the microphones is captured. Once the recording is finished, users can download the audio files via the /download_espmic1 and /download_espmic2 routes. Additionally, if users wish to delete the files, they can do so by accessing the /delete_espmic1 and /delete_espmic2 routes.

The main loop continuously monitors user inputs and manages the system's functionality. The recordingStarted flag is checked to determine when to begin recording. While recording, a beep sound is played, and once the data is captured from the microphones, the flag is updated to false, preparing the system for the next recording session.

This system combines multiple capabilities of the ESP32 to create an integrated solution for audio recording, storage, and access. The synchronized processing of digital audio data from microphones via I2S, its proper storage on an SD card in the correct format, and making the data accessible to users over Wi-Fi are the core functions of the project. The recorded files, stored in WAV format, are suitable for professional use, while the web server offers a user-friendly interface.

3.5 Final Device and System Architecture

In the project, soldering was performed between the ESP32 and other components using jumper cables. This method allowed the connections to form a more organized and compact structure. Thanks to the jumper cables, an arrangement has been made to take up less space between the components with minimum space usage. This not only increased the portability of the project, but also reduced clutter in the circuit design. This approach allows for a more efficient layout a

allows for easy replacement and maintenance of components when needed.

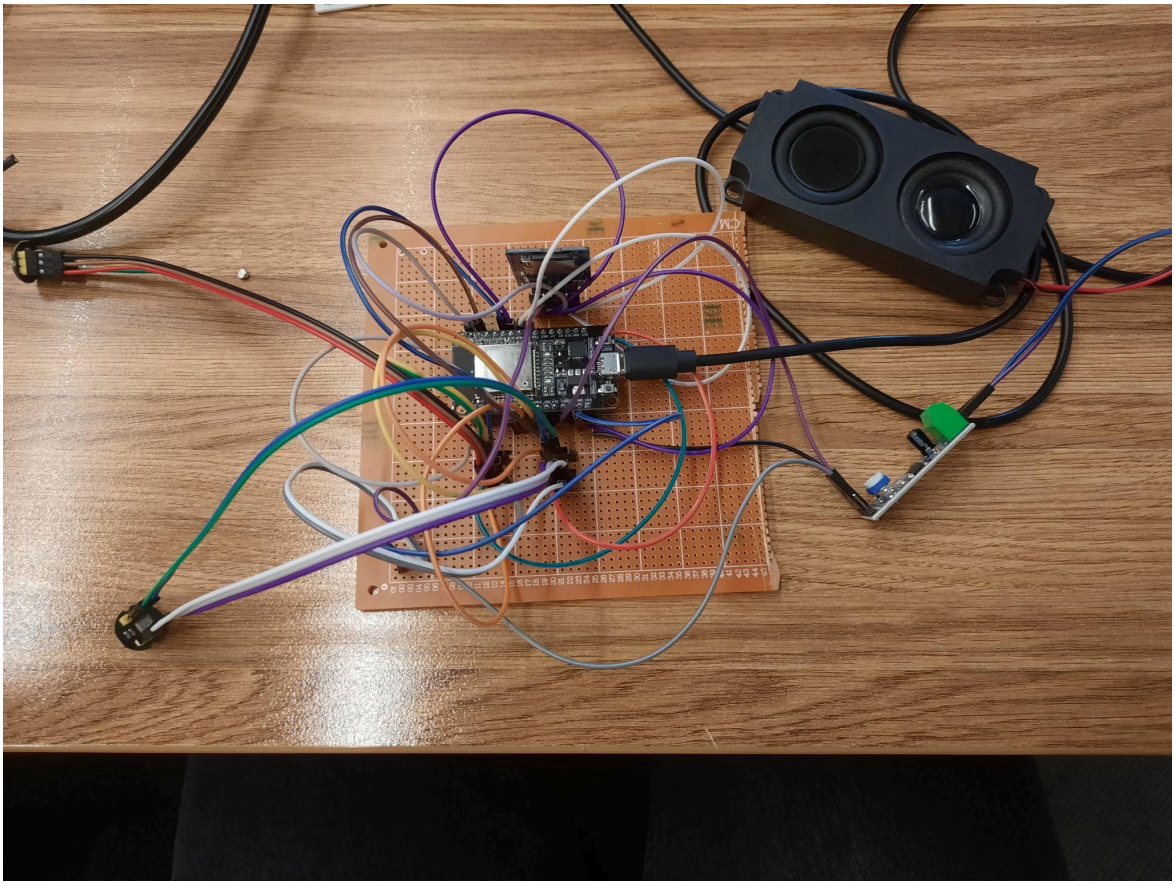

Figure 3.5.1 Final Circuit

1.The final circuit combines the ESP32, INMP441 microphone, LM386 amplifier and microSD module to provide a comprehensive audio processing solution.(Figure 3.5.1)

2.This circuit design enriches the user experience by integrating audio recording and playback functions.(Figure 3.5.1)

The ESP32 provides the ability to quickly create a web server thanks to its integrated Wi-Fi feature, allowing users to access their projects remotely. The web server is ideal for functions such as visualizing sensor data, supporting multiple users, and enabling remote control. The interface created looks like this.

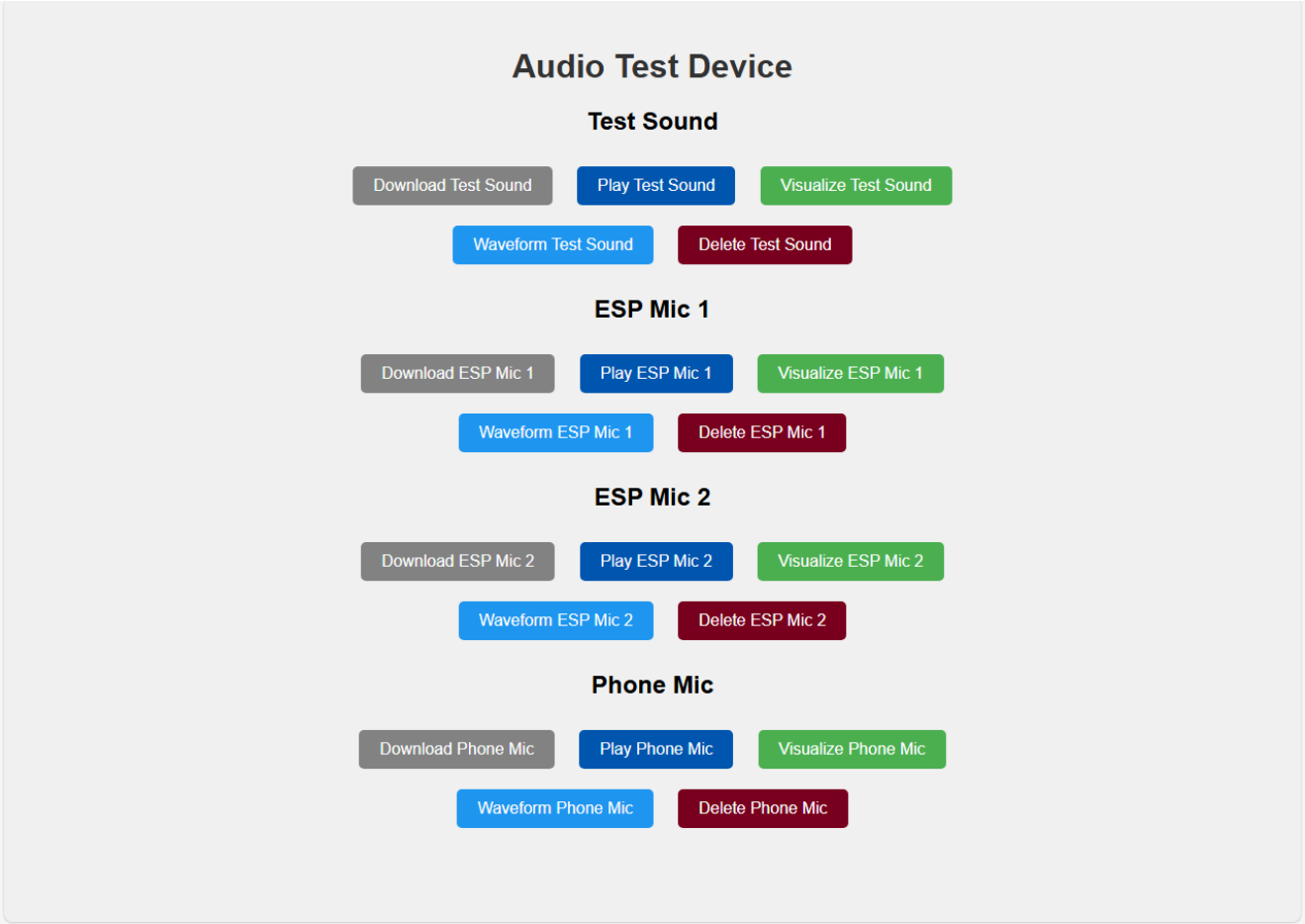

Figure 3.5.2 ESP32 Web Server Home Page

1.The ESP32 Web Server Home Page provides an interactive interface, making it easy for users to access audio data.(Figure 3.5.2)

2.This page provides access to key functions such as audio recording, visualization and file management, thus enhancing the user experience.(Figure 3.5.2)

The 'visualize' option on the server shows the size of the sound instantaneously, while the 'waveform' option shows a graph of the sound function.

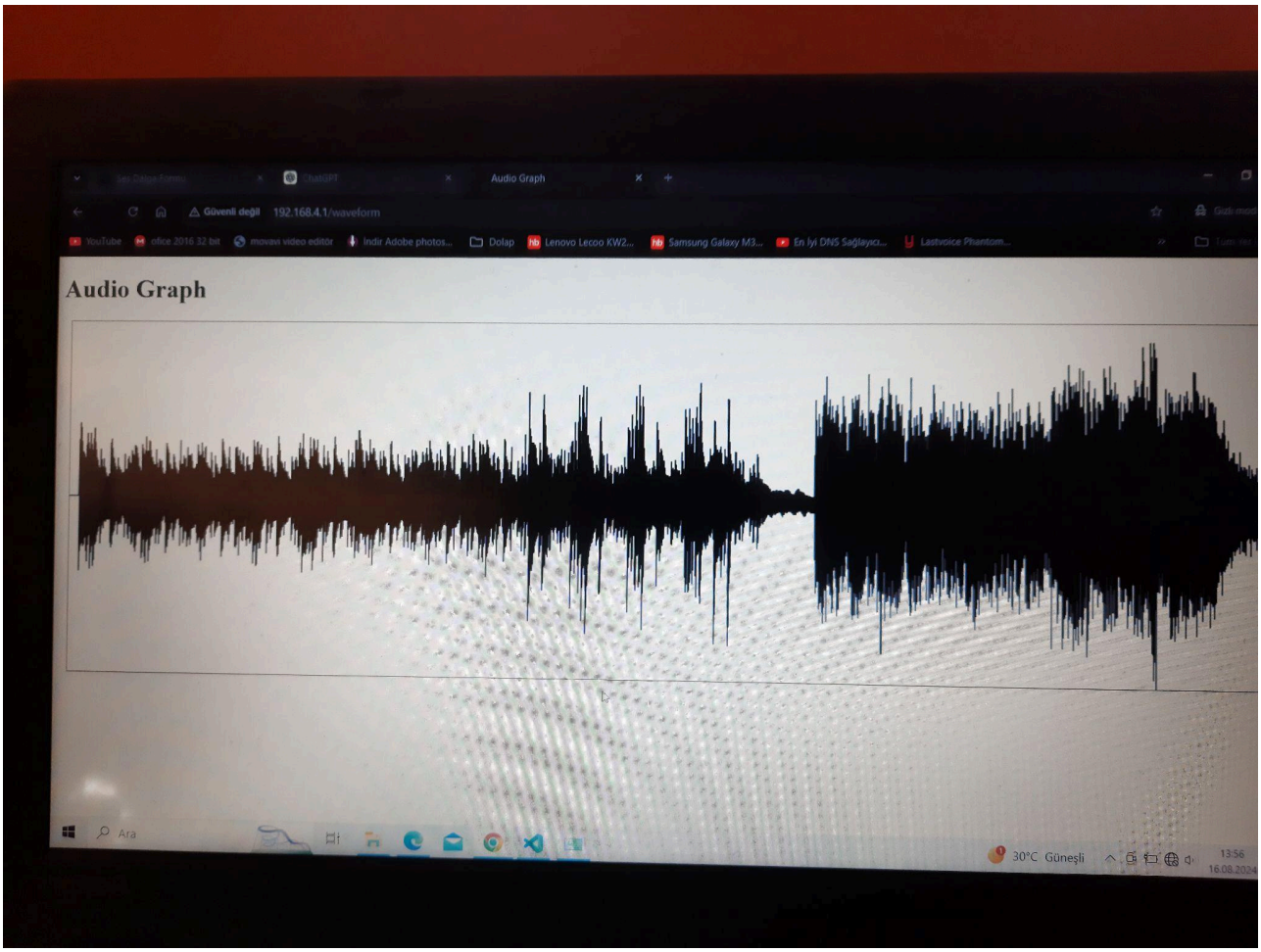

Figure 3.5.3 Waveform of recorded sound

1.The waveform of the recorded sound visually represents the change of the audio signal over time.(Figure 3.5.3)

2.This graph is an important tool in audio processing, helping to analyze the amplitude and frequency of the sound.(Figure 3.5.3)

4. CONCLUSIONS

My internship at AGM Technology provided me with extensive experience in consumer electronics manufacturing and R&D processes. The company's modern SMT and FATP lines enabled high quality and speed production using advanced manufacturing technologies. During the internship, I contributed to projects to improve the performance of microphones and speakers in smartphones, which improved my ability to find solutions to technical problems and helped me understand the importance of balancing quality and speed in the production process.

The company's work environment was very dynamic and collaborative. R&D, production and quality control teams were in constant communication with each other, which ensured that product development processes proceeded efficiently. In this way, I had the opportunity to work with engineers from different disciplines within the team and gained a global perspective. Collaborating with professionals from various cultural and technical backgrounds gave me significant experience in project management and teamwork.

In terms of technical management, I observed how AGM Technology effectively managed its production processes, implementing detailed planning and regular follow-up systems to ensure timely and quality completion of projects. The knowledge I gained during this process helped me gain important technical and managerial skills that I can apply in my professional life in the future.

REFERENCES

1.www.randomnerdtutorials.com:

(From this site I followed the steps on how to run ESP32. To test that the processor was working correctly I used some sample code from the site.)

2.www.forum.arduino.cc:

(In order to solve the problems I was experiencing, I looked at the comments of other users who had similar problems and got some information about it.)

3.www.agmteknoloji.com:

(I have used information about the organization of the company and the work carried out in my report with reference here.)

4.www.dronebotworkshop.com:

(With reference to this site, I learned how to operate the microphone in the I2S protocol and examined the related codes.)

5.www.esp32.com:

(In order to solve the problems I was experiencing, I looked at the comments of other users who had similar problems and got some information about it.)

6.www.instructables.com:

(With reference to this site,I got information about ESP32 web server installation. In line with the information I received, I organized my code in this direction.)