
GEBZE TEKNİK

ÜNİVERSİTESİ ELEKTRONİK

MÜHENDİSLİĞİ

ELM334

Microprocessors

Project-3

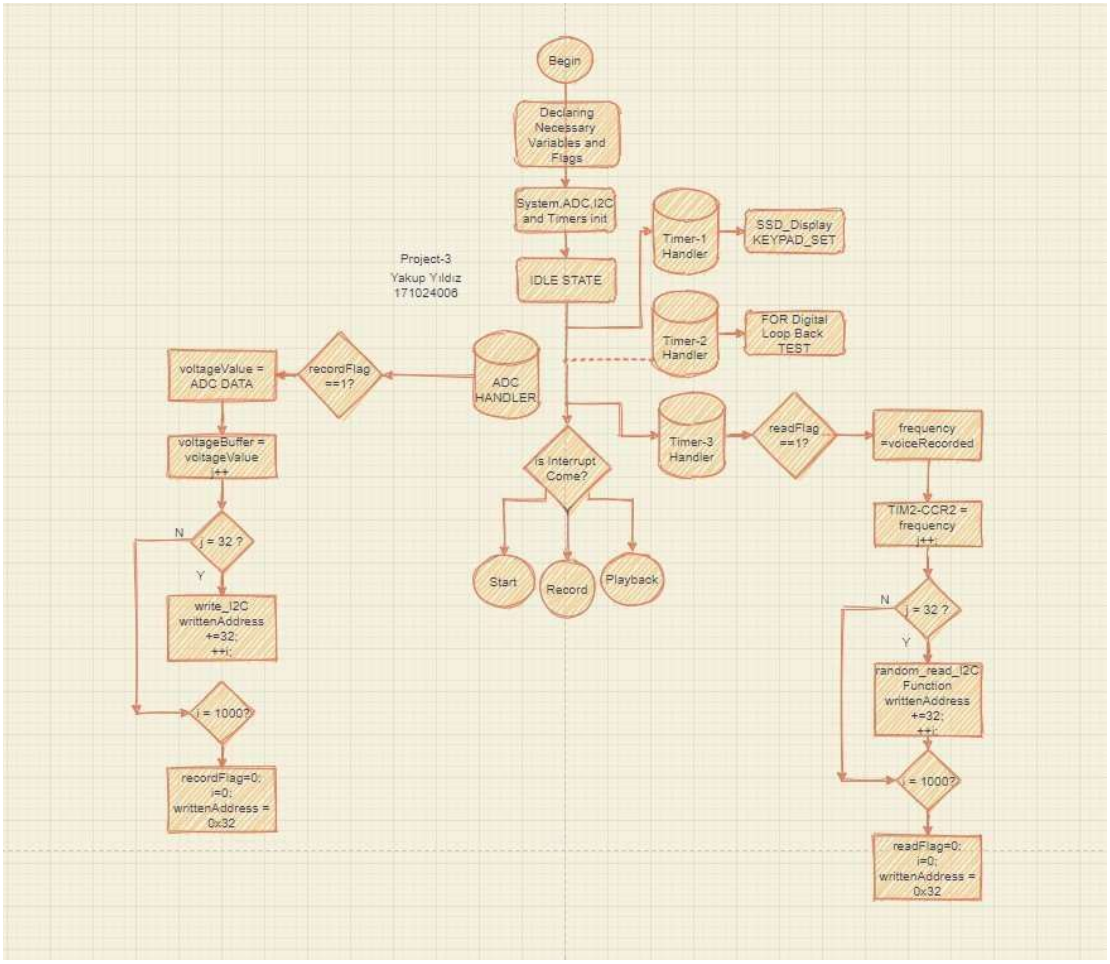
Yakup Yıldız

171024006

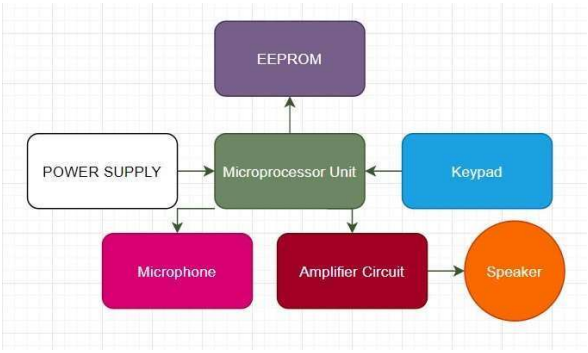
## 1. Introduction

This Project main objective is creating a digital voice recorder that can record voice, playback a voice, and delete data from EEPROM. The modules that is gonna be used in this Project ara Timer, PWM, ADC and External Interrupts. According to the numbers pressed from the keypad, stats must be realized. These stats are named record, playback and clear

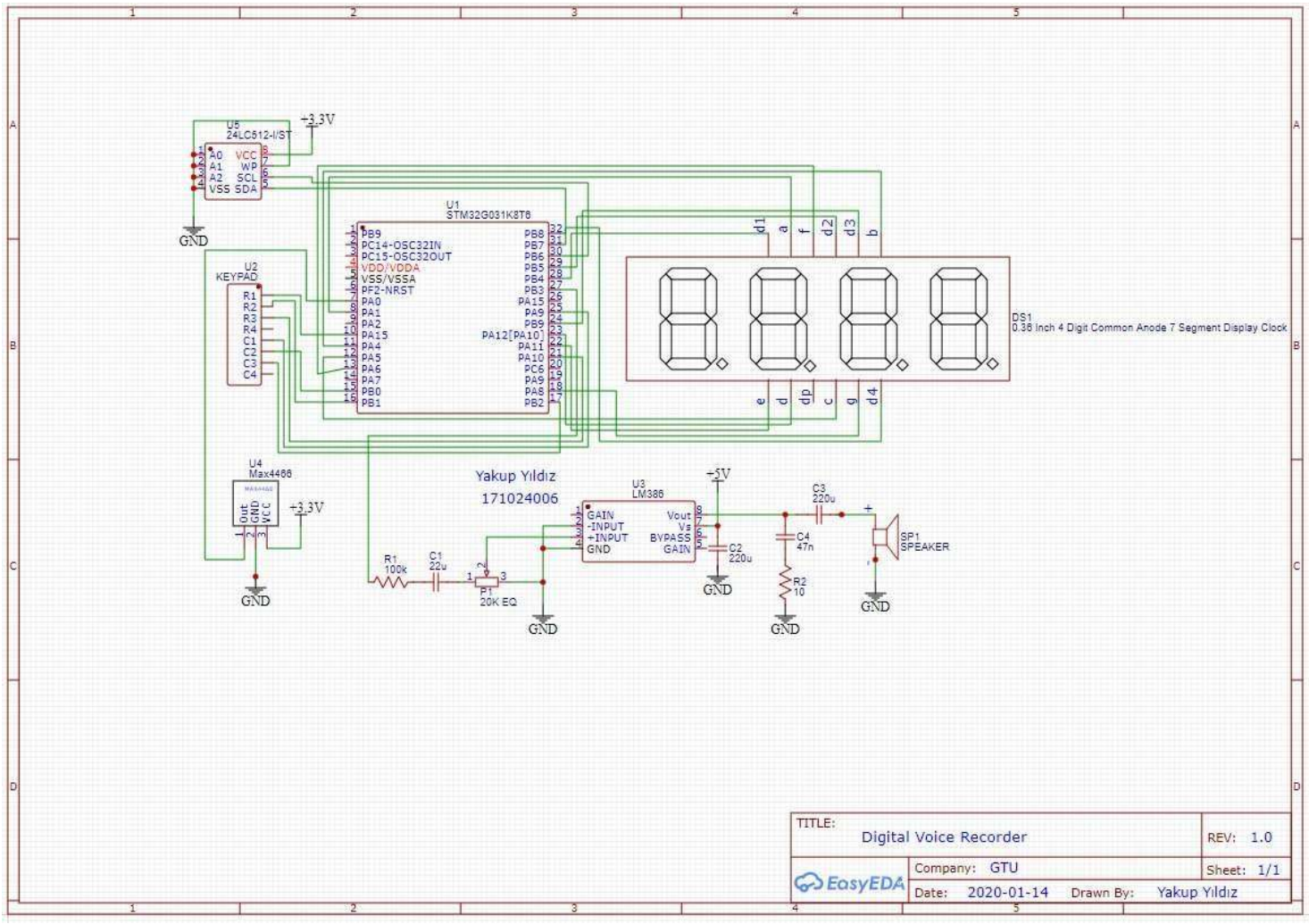
## 2. Diagrams



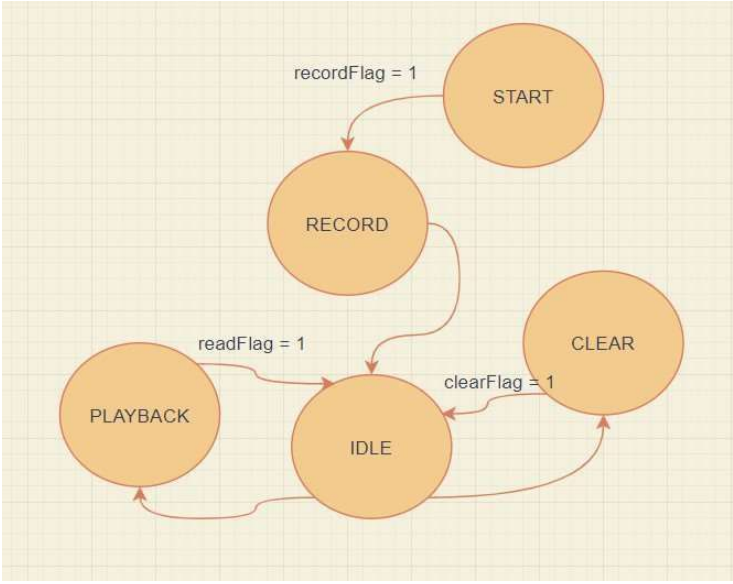

**Flowchart 1**




**Block Diagram 1**




**Schematic 1**




**State Transition Diagram 1**

**3. Parts List**

| | |
|---|---|
| **220uF,22uF,47nF capactiors** | **1tl** |
| **100kΩ,1kΩ,10Ω resistors** | **0.10tl** |
| **20k potentiometer** | **1.42tl** |
| **LM386 Opamp** | **1.65tl** |
| **Speaker** | **7tl** |
| **Max4466 Microphone** | **18tl** |
| **4x4 Keypad** | **8tl** |
| **4x7SD** | **5tl** |
| **24LC512 EEPROM** | **10tl** |
| **STM32G031K8T6** | **120tl** |

**4. Tasks**

**4. 1 Definin Pin Connections ✓**

The pins were determined considering the i2c pwm and adc.

**4. 2 Building a Digital Voice Recorder Circuit✓**

The materials to be used for this are determined. First the required amplifier circuit for the speaker was built then microphone was added to the circuit. EEPROM has been added to the circuit so that analog data from the microphone can be recorded and playback.

**4. 3 Understanding the EEPROM's Working Mechanism✓**

To do this, the datasheet of the eeprom has been examined. Looked at the pin function table and understood where the pins of the eeprom should be connected.

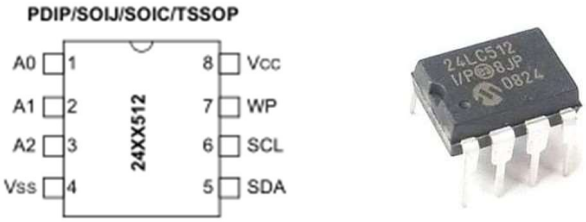



**Figure 1 24LC512 EEPROM**

TABLE 2-1: PIN FUNCTION TABLE

| Name | PDIP | SOIC | SOIJ | TSSOP | 14-lead TSSOP | DFN | CS | Function |
|---|---|---|---|---|---|---|---|---|
| A0 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | User Configured Chip Select |
| A1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | User Configured Chip Select |
| (NC) | — | — | — | — | 3, 4, 5 | — | — | Not Connected |
| A2 | 3 | 3 | 3 | 3 | 6 | 3 | 5 | User Configured Chip Select |
| Vss | 4 | 4 | 4 | 4 | 7 | 4 | 8 | Ground |
| SDA | 5 | 5 | 5 | 5 | 8 | 5 | 6 | Serial Data |
| SCL | 6 | 6 | 6 | 6 | 9 | 6 | 7 | Serial Clock |
| (NC) | — | — | — | — | 10, 11, 12 | — | — | Not Connected |
| WP | 7 | 7 | 7 | 7 | 13 | 7 | 4 | Write-Protect Input |
| Vcc | 8 | 8 | 8 | 8 | 14 | 8 | 1 | +1.7V to 5.5V (24AA512)<br>+2.5V to 5.5V (24LC512)<br>+1.7V to 5.5V (24FC512) |

**Figure 2 PIN FUNCTION TABLE**

A0, A1, A2, VSS and WP are grounded. SDA and SCL are connected to PB-7 and PB6 VCC connected to 3.3 volts for eeprom to work.

### 4.4 Defining EEPROM Address ✓

Since we connect a0, a1 and a2 in grounda, the binary value of the given address is 1010000. By converting it to hexadecimal, the address value is obtained.

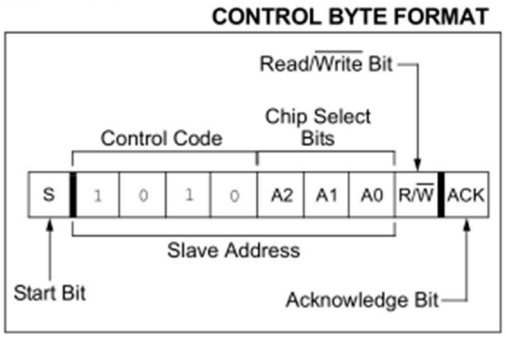



**Figure 3 Control Byte Format**

### 4.5 Defining ADC in Code for Analog Values Read from Microphone ✓

Finding the required register assignments manually using rm0444 reference when defining ADC.

### 4.6 Analog Value Reading ✓

Digitally assigning analog data from the microphone to a variable.

### 4.7 Delivering the Sound ✓

To give the analog values from the microphone in the form of a pwm wave to the speaker.

### 4.8 Determination of Sampling Frequency and Output Frequency for PWM ✓

In order to get a clean sound, the nyquist condition must be met.

### 4.9 Digital Loop Back Test ✓

Receiving the sound from the speaker, which is provided by delivering a sound from phone speaker to the microphone on the circuit

### 4.10 Creating Write and Read Functions for EEPROM ✓

It is necessary to write an algorithm according to the 32kbyte size.

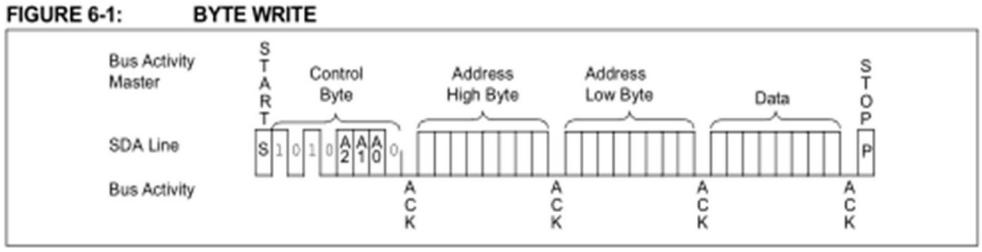



**Figure 4 Byte Write**

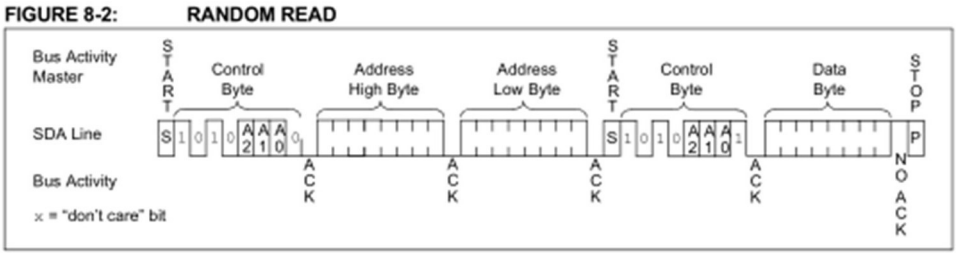



**Figure 5 Random Read**

### 4.11 Assigning the Desired Start, Record, Playback keys to keypad interrupts ✓

By adding flags in the code, operations are started according to them.

### 4.12 Printing States on 7SD ✓

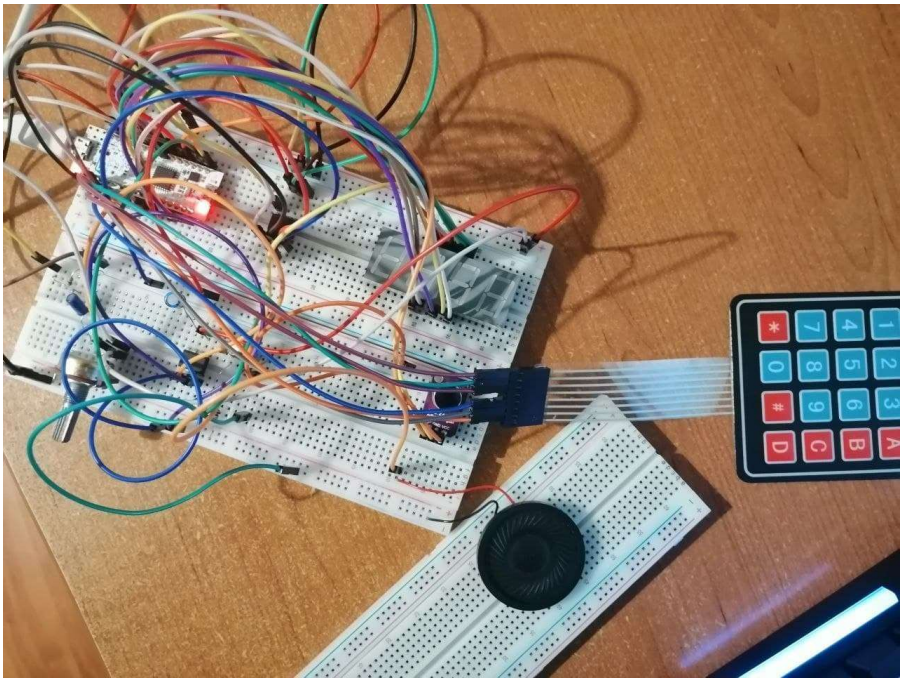
### 4.13 Checking Whether the EEPROM is Working Depending on the States ✓

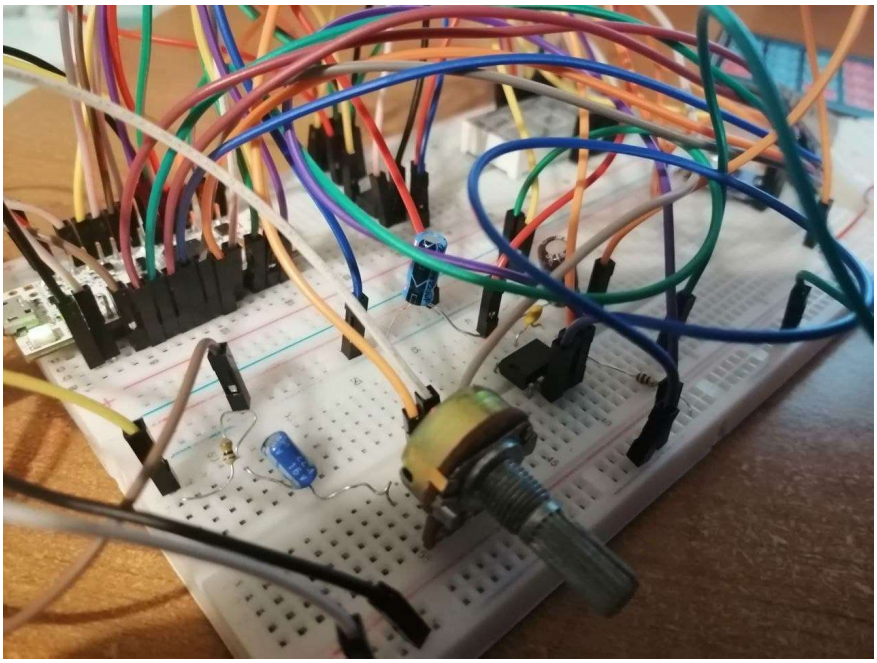
### 4. 14 Checking the Accuracy of Stats Printed on the SSD ✓

Except record others are working.

## 5. Project Setup

An amplifier circuit has been established that has been researched and studied on the internet. Only 3 rows and 3 columns are selected on the keypad because there are not enough pins in the microprocessor.



## 6. Methodology for any Numerical Work

To start with writing and reading from the EEPROM, the eeprom has an area of 64 kbytes. half of this 64 kbps is 32 kbts. “write_I2C” function was written to send this 32 kbyte analog-to-digital data. The function has 4 input variables to send this data. These entries are respectively eeprom's address, address to be written, data and size of data. In order to record the data, it is necessary to press the 1 key on the keypad. When 1 is pressed recordFlag becomes 1 and this function can be run on “ADC_COMP_IRQHandler”. The operation returned in ADC_COMP_IRQHandler is as follows: values converted from analog to digital are kept in variable named "voltageValue" and the mode of this "voltageValue" is taken according to 256 so that the data to be kept can be 8 bits.

Modified "voltage Value" is stored in an array called "voltage Buffer". In order to transfer data to each part of this array, the index j is constantly increased in the Handler, but when j = 32 it means 32 bytes are received. The “writtenAddress” is increased by 32 after the write operation is done. j equals zero and a variable i is incremented by 1. In total, this process needs to occur a thousand times so that we can write 32kbytes of data. If i = 1000, we know that 32kbytes of data is written into the “voltageBuffer”. When i = 1000, recordFlag = 0 and i is set to zero at the beginning of writtenAddres. In this way, the writing process is completed.

There is reading while writing is done. The same method is used for reading. After writing, the values written to EEPROM must be read. The same method is used for reading. When 2 is pressed from the keypad, the read function in timer3 handler operates. A loop mechanism has been set up to read 32kbytes of data into a variable called “voiceRecorded”. Thanks to this loop, values written in voiceRecorded are thrown into a variable named freq sent to “TIM2-CCR2”. Each time the “CCR2” changes a different sound should emerge. Since it is read 32kbytes in total, the recorded sound must be heard.

## 7. Missing Parts of the Project

The specified reading and writing algorithm does work. But the resulting sound is poor, this may be due to frequency mismatch. There is an error in the record text written in SSD. Playback takes more than 5 seconds and there is no countdown in SSD.

## 8. Challenges

It was difficult to adjust the sample frequency and the pwm frequency so that the sound from the microphone can be heard from the speaker. The pwm frequency was chosen much higher than the sample frequency, resulting in a less loud sound.

Another challenge is the process of writing and reading eeproma, almost the part that covers the project. A methodology has been developed accordingly to use half the EEPROM, but when the code was run, the desired was not fulfilled.

## 9. Results and Comments

The project was concluded successfully. But the sound from the speaker was not as good as expected The project was completed in an insufficient period of time due to the lack of information about the eeprom and the many mistakes that were taken during the operations. The project is not about setting up an algorithm in general, but it was about the communication between devices by correctly activating the necessary registers.

## 10. References


- 24LC512 datasheet
- RM0444 referance manuel