University of Technology in Wrocław
Electronics department
Intermediate project

Speech recognition on Embedded system

Autor:
Damian Krata

Course instructor:
Witold Paluszyński, Ph.D.

Abstract

The aim of the project can be divided into two parts: desktop application and embedded application. The goal of the first part was to develop software in MATLAB which will be able to recognize speech (single letters) and use it to control menu. The second part was focused on microcontroller software which include support for SD card, LCD display and embedded microphone. Both parts have been integrated to fulfil all established goals.

January 23, 2018

1 Introduction

The aim of the project was to get familiar with speech recognition methods and prepare software in MATLAB which will be able to control simple menu by voice. It should recognize single letters, numbers and words (such as *back*, *forward*, *up*, *down*) which may be helpful during menu usage. Furthermore embedded software such as support for SD card, LCD screen and embedded microphone has been developed.
Both parts have been integrated. In result application presented below has been created. Application will be useful in further development of master thesis project.

2 System architecture

System has been divided into two parts: embedded and desktop. As we can see on Figure 2 embedded application consist of microcontroller with external devices such as SD card, microphone and LCD display. Desktop application sends information about recognized character to embedded part (via UART). Microcontroller is responsible for displaying received character on LCD screen and saving it on SD card.
Desktop application was written in MATLAB environment. External microphone has been used in development because of higher voice quality and built-in filtration.

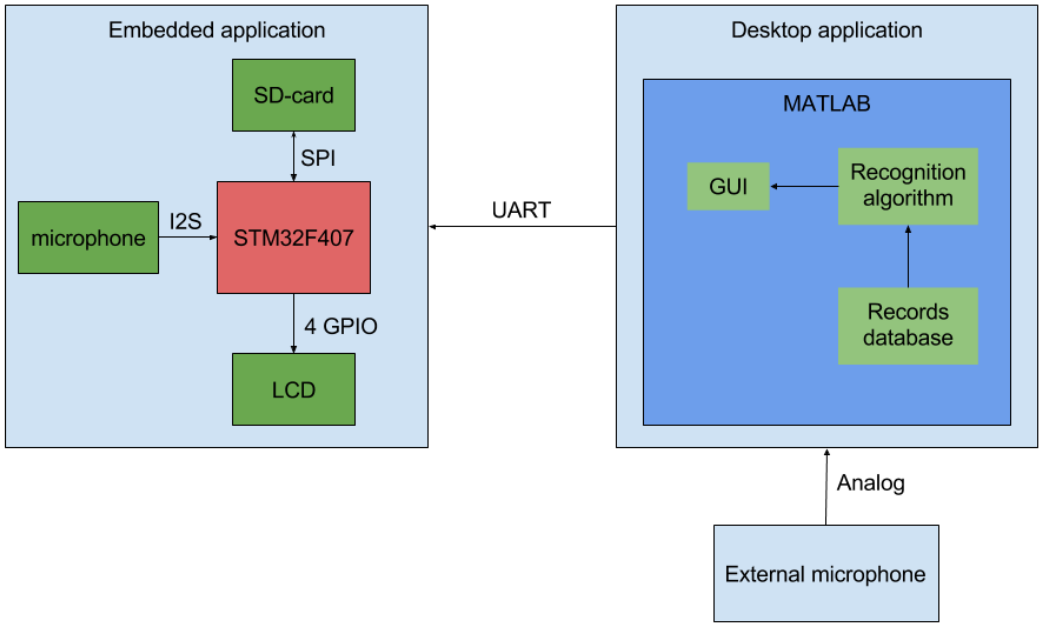


Figure 1: System architecture

2.1 Embedded application

As it was mentioned, embedded application has been build based on STM32F407VGT microcontroller. It was placed on STM32 Discovery development board. We can also find there digital microphone MP45DT02 ST-MEMS. LCD display and SD card have been connected as external devices via cables.

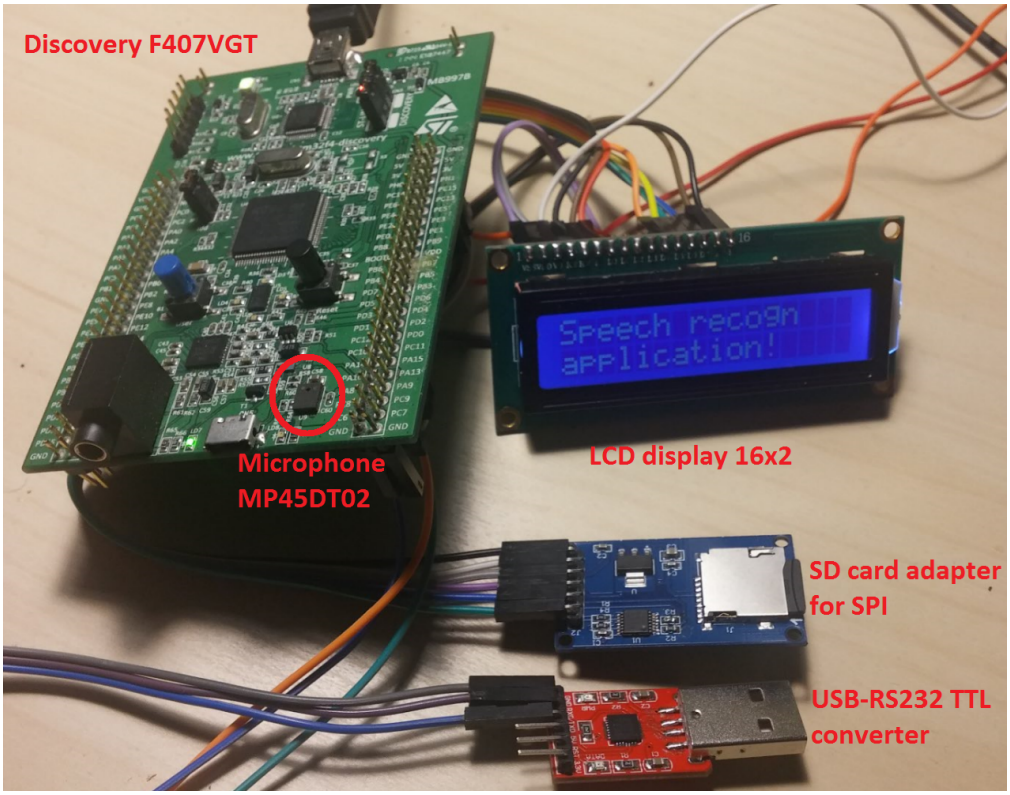


Figure 2: System outlook

Development tools that have been used:

- SW4STM32 - System Workbench - as IDE,
- STM32CubeMX - as low level code generator,
- STM Studio - as debug application,
- HAL library - as low level library,
- FatFS library - as library which allow to save text files in FAT32 format.

2.1.1 LCD driver

LCD display is equipped with hardware driver HD44780. Microcontroller communicates with such driver via 4 line data interface and control lines such as E - enable, R/W - read/write which determines direction of communication, RS - determines if value send to HD driver should be interpreted as instruction or data to display [1]. All of them are configured as GPIO output.
LCD is used to display data received from desktop application. In further development whole system will be placed on microcontroller and LCD would be the only source of real time information about recognized symbols.

2.1.2 SD card

SD card (class 4) is placed in SD card to SPI adapter with outputs characteristic for this protocol: MISO(Master Input Slave Output), MOSI(Master Output Slave Input), CLK(clock) and SS(Slave-Select).
To ensure possibility of log analysis on PC $FatFS$ library has been used in development [2]. It allows to save a file in FAT32 format and read it later on PC.
In future development SD card will be used to store reference samples for speech recognition.

2.1.3 Embedded microphone

Microphone MP45DT02 is placed on STM32 Discovery board. It is an omnidirectional, digital MEMS microphone with PDM signal as an output [3].
PDM is a pulse-density modulation in which analog signal is represented as a stream of bits. Density of the pulses corresponds to the analog signals amplitude [4].

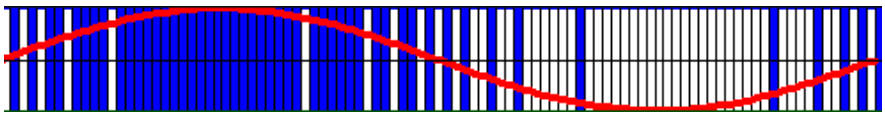


Figure 3: PDM signal [5]

I2S has been used to get data from microphone in PDM format. One of I2S pin (CK-Clock) sends clock signal to microphone with frequency equal to 2.5MHz. As an response, microphone is changing state on its pin called DOUT which has to be read while falling edge of clock occurs. This output is acquired in blocks of 16 samples. Such signal can not be used as an input of speech recognition and it must be converted to PCM (pulse-code modulation) format in which 16-bit value corresponds to signal amplitude at some time t. That is why the data coming from the microphone is sent to the decimation process, which consists of two parts: a decimation filter converting 1-bit PDM data to PCM data and two individually configurable IIR filters (low pass and high pass). The reconstructed audio is in 16-bit pulse-code modulation (PCM) format [6], [4].

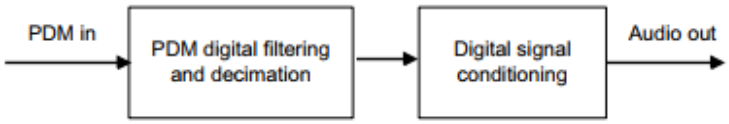


Figure 4: PDM to PCM conversio [7]

PCM signal may be used in speech recognition algorithm, however it is still very noisy, that is why in this project external microphone has been used. Output from microphone has been presented on figure 5. Signal presents numbers from one to four spoken by a person. In future application signal filter will be implemented on microcontroller and whole speech recognition will be performed on it [8].

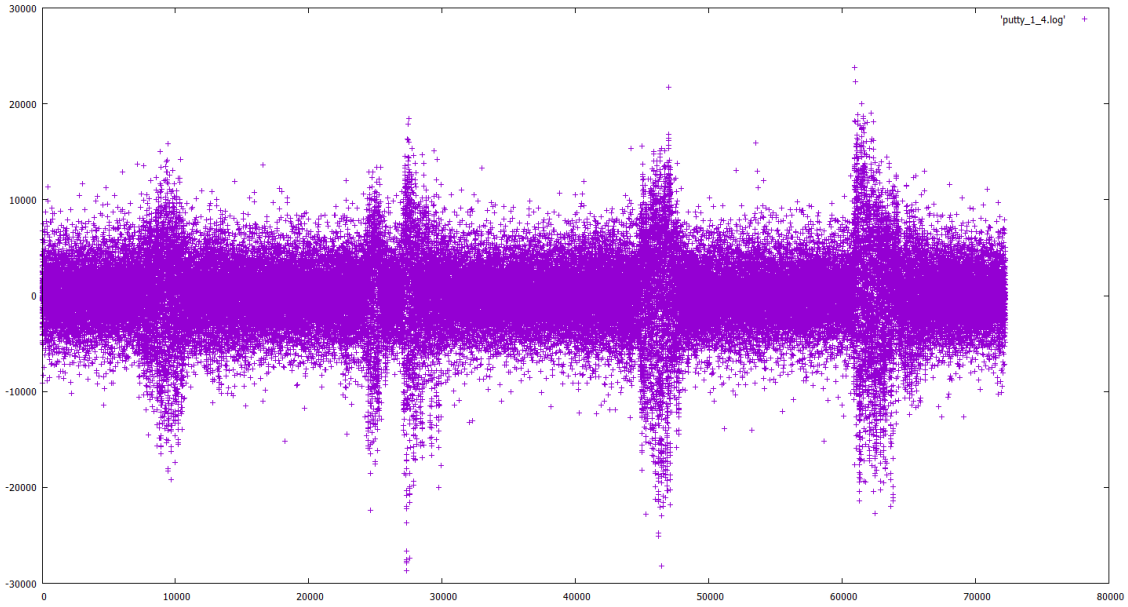


Figure 5: Data from embedded microphone

2.2 Desktop application

Desktop application has been fully developed in Matlab [9]. It consist of three main parts which are:

- records database which contains reference recordings of letters from a to g, numbers from 1 to 5 and also words which may be useful in menu operations (30 samples of each character, 480 samples in total gathered from three different people),
- speech recognition algorithm based on MFCC coefficients and pattern comparison,
- graphical user interface which allows the user to configure parameters of recorded sample, speech recognition algorithm parameters ect.

2.2.1 Records database

Records database, which currently consist of 480 samples, is used to build set of voice feature vectors which will be later compared with new sample spoken by user. All samples were recorded with 44100 Hz frequency and 16 bit value scale. Each of them last 1.5 seconds and weight 130KB. It means that used SD card (4Gb capacity) could handle up to 30 000 of such samples.

2.2.2 Speech recognition algorithm

Speech recognition algorithm has been build based on Mel-Frequency Cepstral Coefficients(MFCC) which are the most commonly used feature extraction method in automatic speech recognition [7]. Algorithm block diagram has been presented on figure 6.

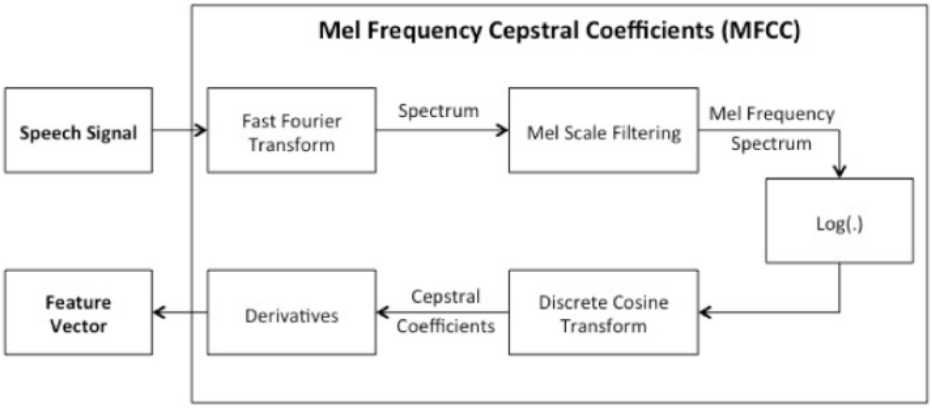


Figure 6: Block diagram of the MFCC algorithm [7]

Input speech signal has been recorded by external analog microphone using MATLAB function *audiorecorder*. Receiving MFCC works in 5 steps: [7], [10]

1. Fast Fourier Transform
   At first frequency domain representation of the input signal needs to be computed. It is done by calculating the Discrete Fourier Transform($X_k = \sum_{n=0}^{N-1} X_n e^{-\frac{2\pi i}{N} nk}$, where N is the number of sampling points within a speech frame).

2. Mel-Frequency Spectrum
   In the second step computation of mel-frequency spectrum has been performed. The spectrum is filtered with N_d different band-pass filters and the power of each frequency band is computed. Such filtering mimics the human ear because the human auditory system uses the power over a frequency band as signal for further processing. The filter bank with the band-pass filters cannot mimic the ear because the ear can use any frequency as center frequency. For N_d equally distanced band-pass filters on the mel-scale has been used. The mel-scale is a special scale (non-linear) that is adapted to the non-linear pitch perception of the human auditory system.

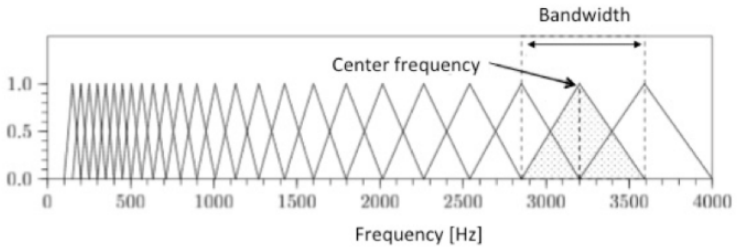


Figure 7: Filterbank with 25 triangular bandpass filters to compute the mel frequency spectrum [7]

3. Logarithm
   Third part computes the logarithm of the signal (it mimic the humap perception of loudness) and it is simply computed as $c^{(3)}_{\tau,j} = log(c^{(2)}_{\tau,j})$ where $j = 0, 1, ..., N_d$.

4. Cepstral Coefficients
   Fourth step tries to eliminate the speaker dependent characteristics by computing the cepstral coefficients. The cepstrum is computed as $c^{(4)}_{\tau,j} = \sum_{j=1}^{N_d} c^{(3)}_{\tau,j} cos\left[\frac{k(2j-1)\pi}{2N_d}\right]$ where N_{mc} is a chosen cepstral coefficients typically between thirteen and twenty.

5. Derivatives
   In the fifth step derivative is computed to reporesent the dynamic nature of speech.

As an output we get matrix of Mel-Frequency Cepstral Coefficients which may be stored in memory and compared with new recorded sample. In the end Euclidean distances between columns of two matrices is computed. The smaller the distance is, the better. Sample from database with the smallest distance to newly recorded signal define class of it.

2.2.3 GUI

Graphical user interface has been developed in MATLAB tool called *GUIDE* and has been presented on Figure 8.

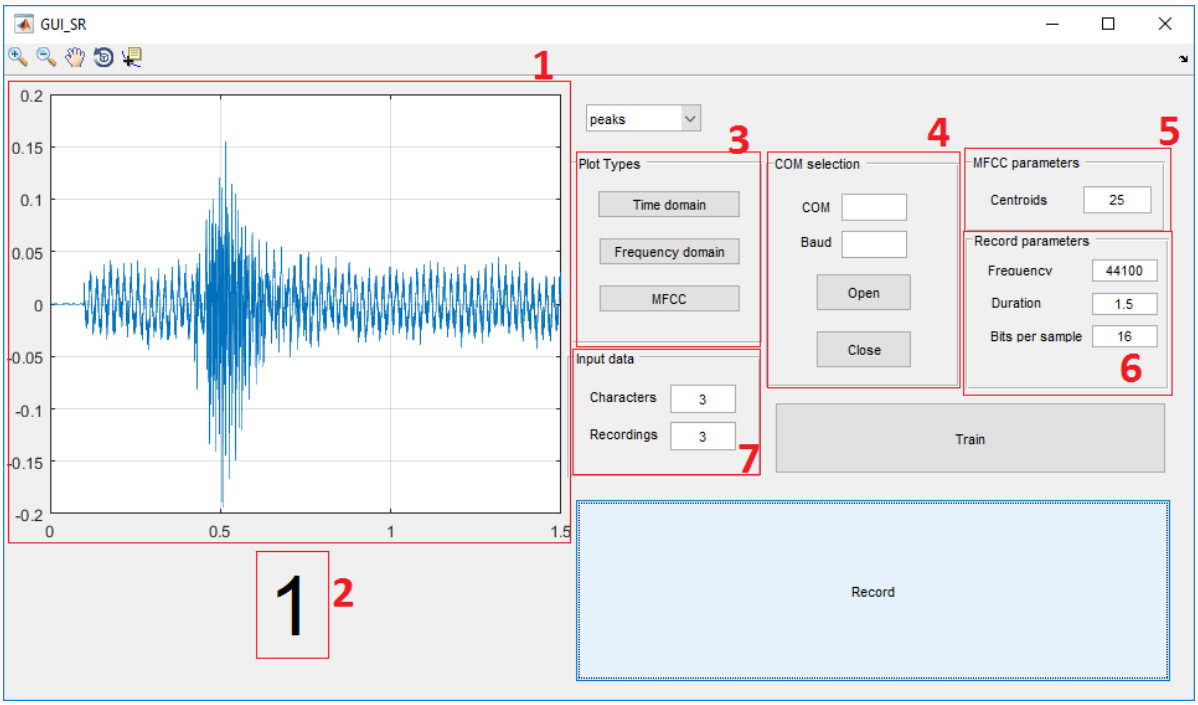


Figure 8: Graphical User Interface

Core items of GUI has been numbered in red. They may be defined as:

1. plot window which is presenting last recorded sample (by default in time domain however it may be changed in block 3).
2. Character recognized by algorithm.
3. Plot type selection. It is possible to choose between time domain, frequency domain and Mel-Frequency Cepstral Coefficients.
4. Block defining parameters of communication with embedded system. Port COM to which RS232-USB converter is connected and baud rate have to be chosen.
5. Number of Cepstral coefficients for speech recognition algorithm.
6. Recording parameters such as frequency, record duration and number of bits per sample.
7. Input data define how many chars from base we want to take into consideration during recognition (be it a, b, c means that parameter should be equal 3 ect., maximally 16) and also how many samples of each char we take in $Train$ phase during which set of MFCC are computed.

We also have two buttons. $Train$, which starts calculating MFCC of given amount of samples from database and *Record* which start recording new sample and perform whole algorithm right after that.

3 Application data flow

To start application at first we need to *train* our program. After clicking $Train$ window with warning appears, it stays for 1.5 second, and after that time disappears and then training starts. During that 1.5 seconds program is recording environment and set reference point at 130% of maximal value in that record. Thanks to that it is able to recognize when nothing was said and print proper message. After that calibration program is ready to work.
From now on, it is possible to watch and record new samples. If communication with embedded system is needed, it is necessary to select proper port COM and baud rate (115200 with included hardware) and click to *Open* button. From now, every recognized character will be displayed on LCD screen and also saved on SD card in *Logs.txt* file.

4 Test

Program output has been presented on figure 9, 10. Words from one to five has been said and in result GUI desktop application displayed last recognized character. LCD screen displays whole set of recognized values, which was also saved on SD card in file $LOGS.TXT$.

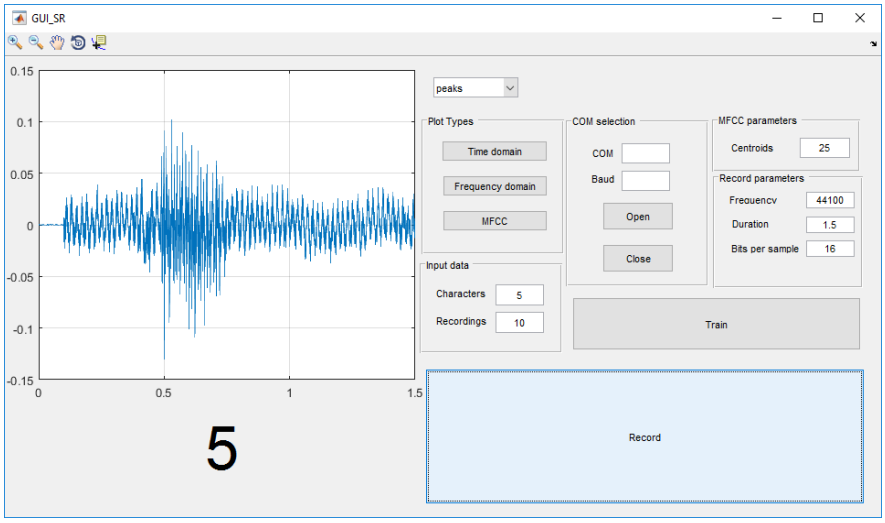
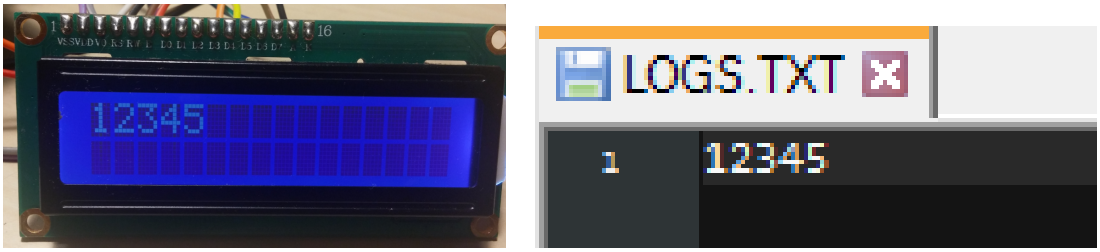


Figure 9: Application test



(a) LCD test

(b) SD write test

Figure 10: Test result for LCD and file on SD card

5 Conclusions

- All goals of the project has been accomplished,
- algorithm has problems with characters which sound similarly such as 'b' and 'd' or '4' and 'forward',
- algorithm works based on previously prepared database and can not recognize new words,
- in order to recognize new words, new feature such as character concatenation must be developed,
- signal from embedded microphone is very noisy and needs to be filtered before further development of speech recognition on embedded system,
- project was step for future master thesis project and it will be a good base for future development.

References

[1] WyŚwietlacz alfanumeryczny hd44780. `http://mikrokontrolery24.pl/peryferia/wyswietlacze/hd44780.html`.

[2] STMicroelectronics. *Developing Applications on STM32Cube with FatFs*, 2 edition, 5 2014. UM1721.

[3] STMicroelectronics. *MEMS audio sensor omnidirectional digital microphone*, 1 edition, 6 2016. MP45DT02-M.

[4] STMicroelectronics. *PDM audio software decoding on STM32 microcontrollers*, 1 edition, 9 2011. AN3998.

[5] Pulse-density modulation. `https://en.wikipedia.org/wiki/Pulse-density_modulation`.

[6] Jan Szemiet. Analizator widma z fft na stm32 z cortex-m4. `http://mikrokontroler.pl/2013/12/06/analizator-widma-z-fft-na-stm32-z-cortex-m4/`.

[7] Michael Lutter. Mel-frequency cepstral coefficients. `http://recognize-speech.com/feature-extraction/mfcc`.

[8] C. Bernal-Ruiz, F. E. Garcia-Tapias, B. Martin del Brio, A. Bono-Nuez, and N. J. Medrano-Marques. Microcontroller implementation of a voice command recognition system for human-machine interface in embedded systems. In *2005 IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 5 pp.–591, Sept 2005.

[9] Create apps with graphical user interfaces in matlab. `https://www.mathworks.com/discovery/matlab-gui.html`.

[10] Prof. Savitha Upadhya Koustav Chakraborty, Asmita Talele. Voice recognition using mfcc algorithm. 2014.