Speaker Recognition Based Lock

Electronics design lab – II Report

Group D3

Poojan Sheth 05D07002

Abhijeet Jain 05D07003

Sagar Shenvi 05D07021

Abstract: The aim of this project is to make a speaker based lock. The voice samples of the legitimate user are recorded and processed by using Mel-frequency cepstrum coefficients(MFCC), and the coefficient table stored by the processor. For any other user, the MFCC of his voice samples are compared with that of the legitimate user, and the lock unlocks only if a valid match is obtained. We use a MSP430 processor as the central processor for this circuit.

1.Problem Statement: A speaker based lock differs from ordinary locks in that it works by identifying the voice of the speaker rather than a password. Information about the legitimate speaker's voice samples is stored in the memory of the lock in some form. Whenever any user requests identification, his voice samples are recorded and processed to compare with that of the legitimate user. If the correlation between the two exceeds a certain threshold, then the user is granted access, else he is not allowed permission. We describe the software and the hardware setup of our lock now:

2.Design Approach:

A .Software: We use the mel frequency cepstrum coefficients(MFCC) as the transform to evaluate the voice samples. Taking the MFCC of a voice sample consists of the following

1. Take the Fourier transform of (a windowed excerpt of) a signal.

2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

These coefficients can represent the frequency content of the user's voice more accurately. We have terminated number of coefficients to 91. The MFCCs of the user are stored in the RAM, which requires lesser memory than storing the entire voice sample. For any prospective user, we compute the MFCC of his voice samples and then take its correlation with the legitimate user's MFCC. If this exceeds a threshold, the user is allowed excess else he is denied permission. We wrote a code using this MFCC approach on Matlab, and it yields success at a high rate by correctly identifying the legitimate user and rejecting others. It is almost 100% correct in allowing access to the legitimate user. However, its success rate for other users is not so good, though it rejects most, it might allow users (success rate might be around 80%).The other very important issue is the quantization error. As, MSP430 is a fixed point processor with number of bits per word being up to sixteen, we will have quantization error and it will affect the process. The feature vectors will get displaced non-uniformly in the feature space and so, both probability of false alarm and of no detection will increase.

The software driving SRAM and the LCD were written as per their timing diagrams. More about the RAM in the hardware section.

B.Hardware:

We mention the components we have used along with the reason for selecting them:

1)Processor: MSP430F168. It has 2kb RAM. One of the important reasons for selecting it was because it has hardware multipliers for quicker computation. It works with a 8Mhz clock for fast computations. It is also a low power processor with 4 sleep states, which is important in a device which would be kept on throughout the day. It has 10 bit high speed ADC for sampling the signal in multiple modes, and using the inbuilt timers we can set it to an exact sampling rate. The burning of the code is done through IAR workbench using a JTAG, and the changes in the various registers can be observed in the workbench.

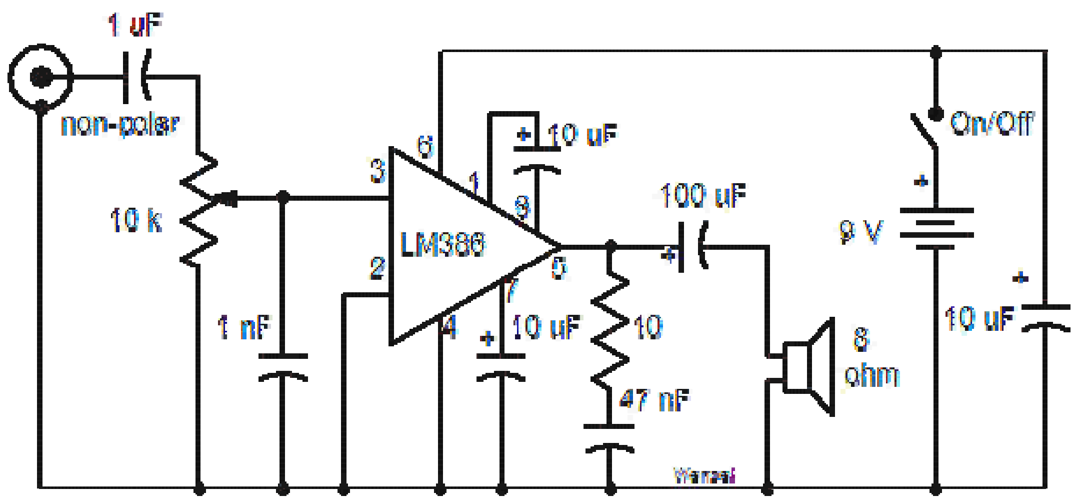
2) RAM: Cypress CY7C1049DV33-10VXI .As calculating MFCCs takes time which is greater than the sampling interval for our desired frequency. So, real-time processing cannot be done. This implies we need to store all the speech samples before processing them. Since saving the voice samples while taking the MFCC requires space, and the 2kb (internal RAM capacity) space is not sufficient, we use a 512 kb RAM. It also helps keep the look up tables. The reasons for choosing this particular RAM are easy interfacing with the MSP, and don't need to do block transfer. As block transfer introduces synchronization problems, RAM was chosen over an SD CARD. Also, we can get RAM that requires the same voltage as the MSP(3.3 V) eliminating the need for a separate source.

3)LCD: JHD162A LCD for user-lock interaction.

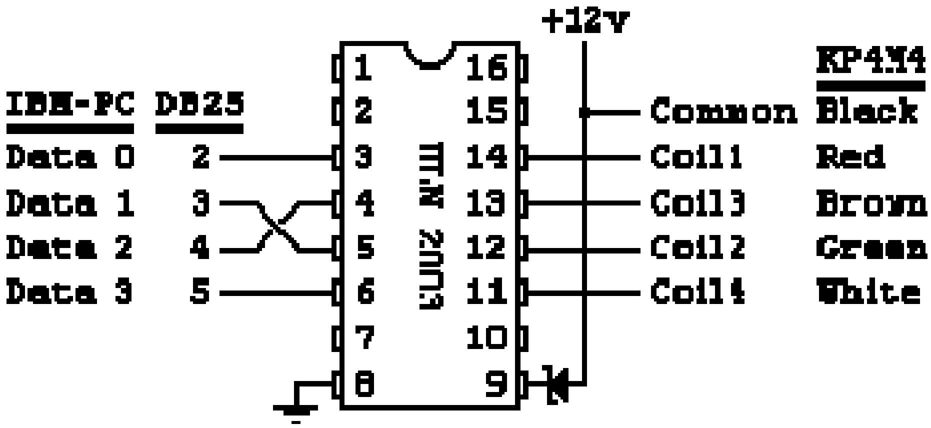
4)Mic circuit: Standard mic with an amplifier. We use LM386 as the amplifier.

5)Stepper Motor: We use the unipolar stepper motor motor. The driver chip used is ULN2003 .

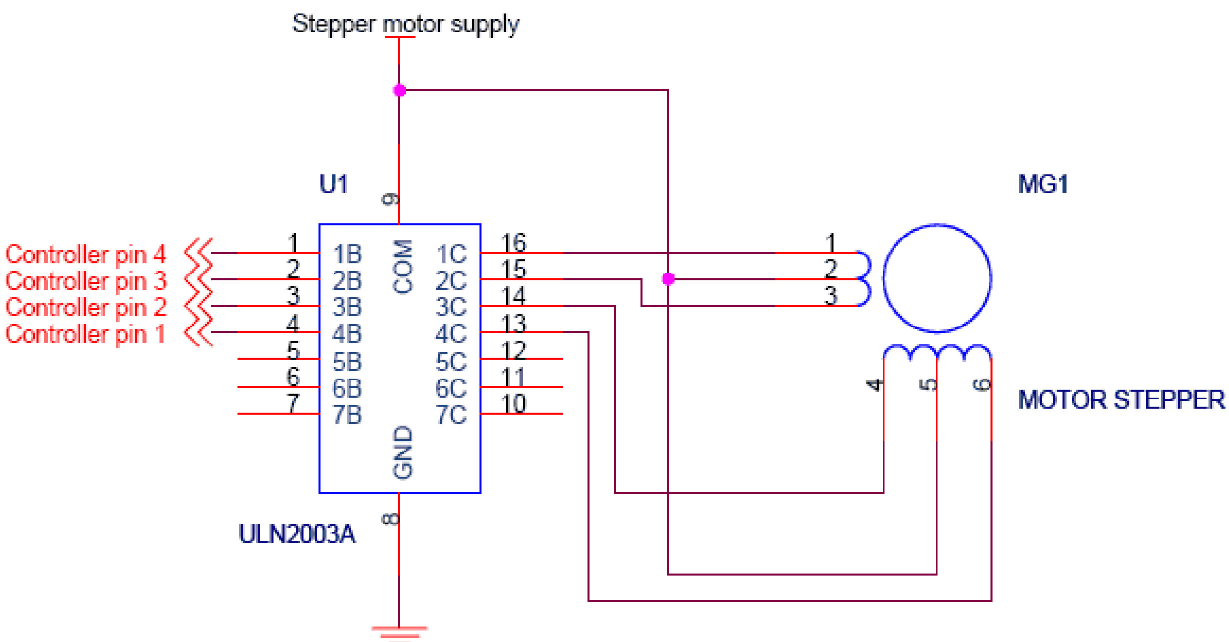
Circuit Diagram



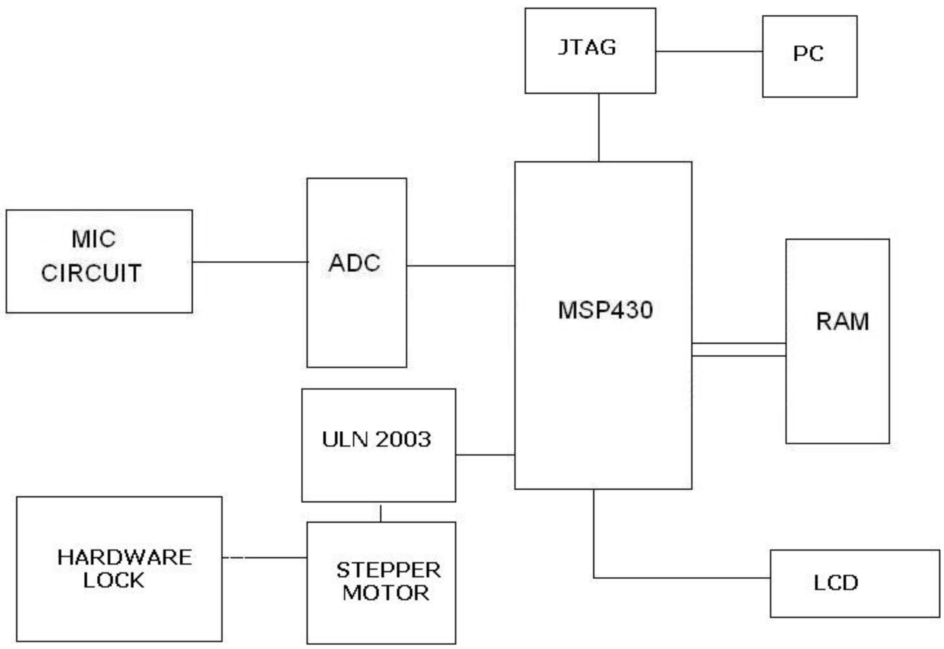
LM386 Audio Amplifier



Connection Diagram for Stepper motor driver



Flow Diagram and Description:



BLOCK DIAGRAM

We mention how the circuit hardware functions. The ADC samples the data at a sufficient rate. These samples are stored in the RAM by the MSP, and then we compute the MFCC of the stored samples. We compare the MFCC of the current user with that of the legitimate user. If the similarity exceeds the threshold, the MSP directs the driver chip of the stepper motor accordingly so that the lock unlocks and LCD is used for interfacing with the user. Else access is denied. The thresholding is done on the basis of Euclidean distance.

The ADC is used in the single channel repeat mode for doing the sampling. The sampling frequency is approximately 19khz.

3.Completed Parts of the Project:

The MSP430 daughter board along with its JTAG circuit is working correctly. The ADC of the MSP is working correctly. The interfacing of the RAM with the MSP has been tested by writing a code to write something to the RAM and then reading it back through another port. The RAM is working correctly as well. The MIC and amplifier circuit is working correctly. The LCD is working correctly, and we have written a code to interface it with the MSP. The MATLAB code for speech recognition using MFCC has been written and tested to work correctly.