2012 IACSIT Hong Kong Conferences
IPCSIT vol. 30 (2012) © (2012) IACSIT Press, Singapore

Implementation of Voice Recognition in Low Power Microcontroller

Nitin Kandpal[+], Yashodhan Mandke and Amit Patwardhan

SOST Department, I2IT, Pune, India

Abstract. This paper presents the voice recognition algorithm and implementation of the same in AVR ATmega128 microcontroller. Due to acoustic nature of speech it's difficult to recognize by microcontroller and requires lot of processing, computation and filtering. In 8 bit microcontrollers, availability of less SRAM makes the task complex. To solve the problem there is need of some characteristic or feature of speech which makes the word unique. By using bank of filter method features can be extracted which generates finger print. The applications of this project are in voice controlled handicap chair, voice security system and simple embedded systems.

Keywords: discrete fourier transformsfilters, microcontroller and analog-digital conversion.

1. Introduction

Before Alexander Graham Bell became famous of inventing telephone in the late 1870's he spent years of time to build a system for deaf to visualize speech but failed to do so [1]. His work was demonstrated by number of people who have been trying to develop speech recognition system. The first attempt to implement automatic system began in the 1950. The first significant ASR system builds in Bell labs in 1952 by Davis Biddulph while isolated word recognition system was investigated in the 1970's [2]. There are several ways of characterizing speech. One highly quantitative approach is in term of information theory that speech can be represented in terms of its message content or information [3]. The chances of human voice frequency going below 600 Hz or above 4000 Hz is very less. Majority of human voice frequency lies between 1000 Hz to 3300 Hz [5]. The voice waves are created by vibration and are propagated in air by vibration of particles of media. Due to acoustic nature of voice it is complex task to recognize voice in microcontroller. To recognize the speech first step is to understand the characteristic of word, features of that word. E.g. if anyone says 'hello'; means the word hello has some features or content because of which one can listen the same hello word. Voice recognition is to provide intelligence to embedded system so it can interpret voice and execute commands accordingly. Application of project is that simple embedded system can be controlled by some voice command.

2. Speech Recognition

Initially the various voices from different speakers were stored in Matlab. By implementing FFT algorithm and correlation between voices we can able to detect the some voice commands.

[+] Corresponding author. Tel.: + 020-25478256; fax: + 2022934592.
E-mail address: *nitinsonu5@gmail.com.*

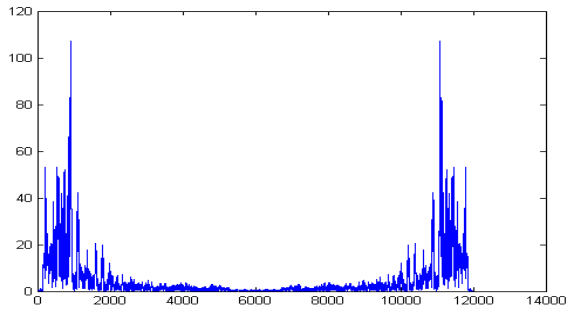

Fig. 1: FFT of voice command move forward

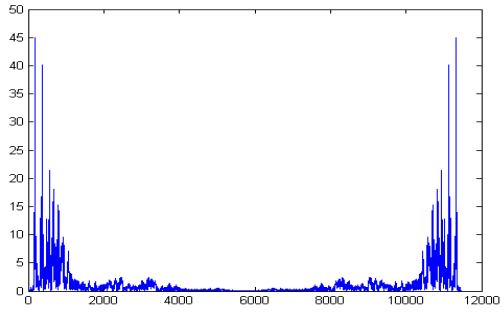

Fig. 2: FFT of voice command move backward

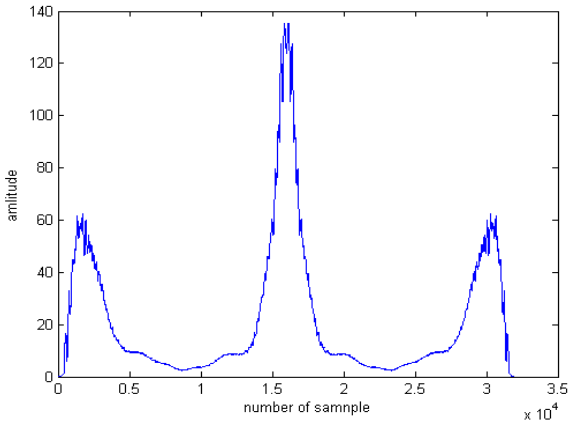

Fig. 3: Correlation for voice command with move forward

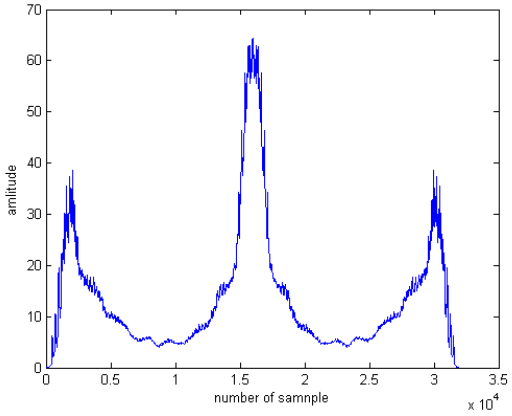

Fig. 4: Correlation for voice command with move reverse

In Figure .5 the maximum correlation value between move forward with move forward is around 130 and in Figure .6 the maximum correlation value between move forward with turn right is around 65 which is very less than 130 so by this same command can be detected. This algorithm was working very effectively to recognize voice command. But to implement FFT algorithm in microcontroller is very complex task because of problem of floating point and imaginary number. To reduce the implementation complexity bank of filter method used in this project which enables to detect only four to five orthogonal word

2.1. Bank of filter analysis

Using bank of filter method 8 bandpass Chebyshev filters were implemented with band of frequency of each filter is 200 Hz. Chebyshev filter is used to separate one band of frequencies from another. The primary attribute of Chebyshev filter is their speed, typically more than an order of magnitude faster than the windowed sinc. By using 8 filters, a range of 200Hz – 1800Hz frequency component can be collected and from Euclidean distance formula voice can be detected.

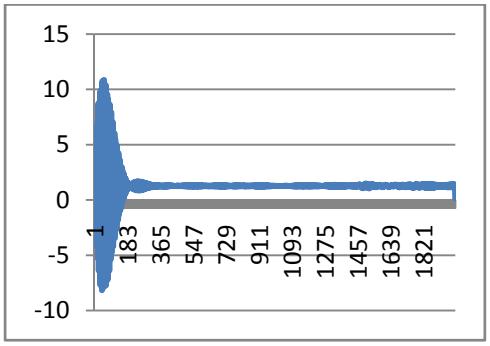

Fig. 5: Chebyshev filter output cutoff frequency 200 to 400

Characteristics of command “Chal” applied in Chebyshev filter whose pass band frequency is 200 Hz - 400 Hz output of filter shown in Figure.7. As shown in Figure.7 filter gives all characteristics of command

between 0 to 200 samples because pass band frequency is 200 Hz. To recognize speech its required to observe at frequency content of speech. To implement the 4th order Chebyshev band pass filter, two 2nd order filters are used.

Equation of filter

$$y_1(n) = b_{11}x_1(n)+b_{12}x_1(n-1)+b_{13}x_1(n-2)-a_{11}y_1(n-1)-a_{12}y1(n-2) \quad (1)$$

$$y_2(n)=b_{21}y_1(n)+b_{22}y_1(n-1)+b_{23}y_1(n-2)-a_{21}y_2(n-1)-a_{22}y_2(n-2) \quad (2)$$

Where a and b are coefficients of filter and g is gain of filter. Coefficients a and b are determined by Matlab command

$$[b,a] = \text{cheby2}(2,40,[Freq1, Freq2]) \quad (3)$$

$$[sos2, g2] = \text{tf2sos}(B2, A2,'up','inf') \quad (4)$$

Where, Freq1 and Freq2 are normalize cutoff frequency.

To recognize voice, 8 different Chebyshev filters are used where each Chebyshev filter produces 2000 samples individually. In all, 16000 samples are collected and segmented in groups of 125 samples. These 125 samples are added together and each segment is summed up to produce a set of 128 point output. These 128 points is considered as a single fingerprint of the given voice. The fingerprint represents characteristic of sound in frequency domain as time involves. The interest is here that how the power involves in particular band of frequency. This fingerprint is just a vector of number each number represents the energy or average power that heard in particular frequency band, during particular interval time.

In order to identify the similarity between voices, Euclidean distance measurement is used here. It is very similar to the Correlation algorithm and in cases where spectrum has no negative spikes and has a good signal-to-noise ratio, it will produce equivalent results. The mainadvantage of the Euclidean Distance method over the Correlation method is that it is faster and less computation required.

$$\text{Euclidean distance } d= (\sum (x_i-y_i)^2)^{1/2}$$

Where x is Voice finger print, x=x1,x2,x3…x128and y is Stored finger print, =y1,y2,y3….y128.

In Bank of filter method, 16000 samples passes through filter because of that a lot of computation is required and the output of the algorithm is slow, so we used Euclidean method instead of Correlation.

When the Euclidean distance values between two voices is minimum, it is considered as a same voice and hence the voice is detected.

3. Hardware Implementation

3.1. Microcontroller AVR ATmega128

The ATmega128 is low power CMOS 8 bit microcontroller based upon RISC architecture. The Atmega128 have 128k bytes of in system programmable flash with read while capabilities, 4 k bytes EEPROM, 4 k static RAM, 53 general purpose I/O lines, 32 general purpose working register, real time counter, four flexible timer counter with compare modes and PWM, 2 USART, an 8 channel 10 bit ADC with optional differential input stage with programmable gain [8].

We are using 4 kHz sampling frequency for detecting some particular words commands because when we take 8 kHz sampling frequency for 1 second duration, the memory requirement go high up to 8 kb static RAM which would neither be technically nor economically feasible to these high end processor. The voice recorded from microphone at 4 kHz sampling frequency for 0.5 second which requires 2 kb static RAM which is available in ATmega128.

USART speed

The Atmega128 has two USART's USART 0 and USART 1. To know what is going on in microcontroller it is important to connect microcontroller with computer that is the main purpose to implement USART here.

Sampling frequency of voice command = 4000 samples /second

To send 4000 sample in a second baud rate required = 4000x8 = 32000 bits /second

standard baud rate = 115200

At 1 M Hz frequency maximum 4800 baud rate can achieve, to achieve 115200 baud rate external crystal is used.

External crystal frequency = 11.0592 M Hz

For interfacing the External crystal fuse bit change in ATmega128

L fuse w: 0xc1:m U fuse w: 0xd9:m C fuse w: 0xff:m

ADC speed

The ATmega128 have 10 bit successive approximation ADC. ADC port of ATmega128 is port PF. Sampling frequency of each command is 4k Hz so requirement is that more than 4000 sample should be converted by ADC in a second.In ATmega128 ADC first conversion require 26 clock cycles than after for each conversion it takes 13 clock cycles.

number of conversion at least required in a second = 4000

number of clock cycle required for 4000 ADC conversion= 26+3999x13=52013

External crystal clock of ATmega128 =11.0592x 10^6

So prescalar that can be used= crystal clock frequency/number of cycle required

= 11059200/52013= 212

ADC speed of ATmega128at prescalar 128 =11059200/{128.(13)+13} =6594

Programm

To store any string or use flash Ram memory, progmem command is used. Progmem store the data in flash (program) memory instead of SRAM. The progmem keyword is a variable modifier; it should be used only if the datatype defined in pgmspace. It tell the complier “put the information in flash memory”, instead of into SRAM where it would normally go. Finger prints of voice commands are stored into flash RAM. By using flash RAM we are saving SRAM because SRAM memory is used for take the data from voice command in real time.

8 Bit internal timer

By using 8 bit timer whenever the TOVO flag interrupt call one value of ADC will passes for processing. Sampling frequency is 4000 so vector over flow interrupt call 4000 in second.

Target time count = (1/Target frequency)/ {(1/timer clockfrequency)-1}

= [(1/4000)/ {(1/11059200)-1}]= 2765 count

It shows that TCNT0 count from 0 to 2765 than TCNT0 will be reset to 0 and one value of ADC will pass.

4. Result

Table.1 shows the result of bank of filter method for single speaker. The minimum Euclidean distance between voice commands is the same command.

Table. 1: Euclidean distances between commands for speaker 1

	Chal	Ruk	Daya	Left
Chal	**022350**	35813	50178	32915
Ruk	37171	**25519**	33170	39476
Daya	38671	37381	**32513**	35718
Left	31823	38892	32195	**27501**

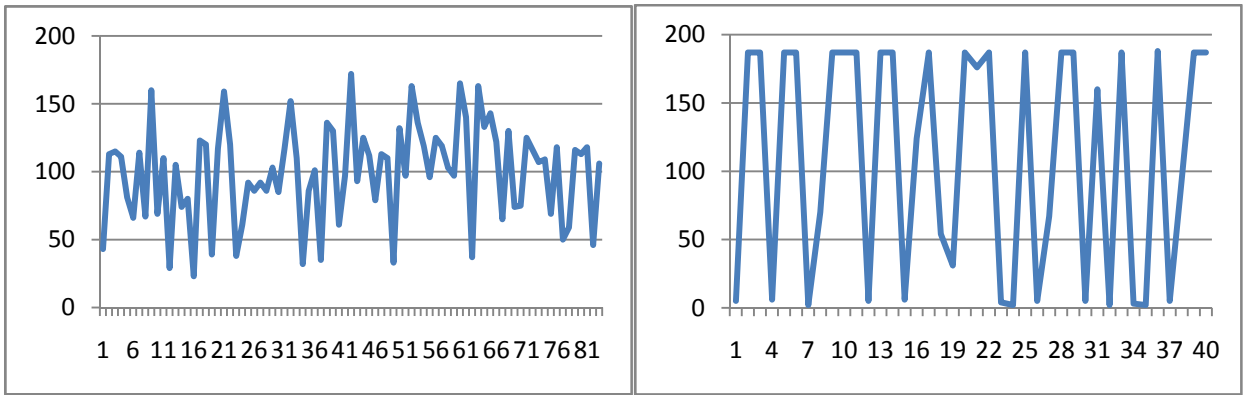

Fig. 6: Output of microcontroller without input voice and with voice

5. Conclusion

The implementation of word recognition in low power microcontroller is very complicated work because lot of processing and filtering required for extracting the feature of speech. To convert the voice signal in electrical signal by microphone creates lot of problem due to acoustic nature of voice. There are some external parameters also which effect voice recognition system such as environmental noise. In noiseless environment the system accuracy is around 70 to 75% depends upon microphone position also and in noisy environment system accuracy is around 40 to 45%.

6. Acknowledgement

I thank to Prof. Rabinder Henry, Prof. Amit Patwardhan, my seniors and my class friends for helping me in whichever way possible

7. References

[1] A High Performance Custom Hardware Backend Search Engine for a Speech Recognition System ,Edward Lin , December 13, 2007 Department of Electrical and Computer Engineering Carnegie Mellon University.

[2] Speech Recognition on DSP: Algorithm Optimization and Performance Analysis , YUAN Meng: The Chinese University of Hong Kong July 2004.

[3] C. E. Shannon “A Mathematical theory of communication” Bell system Tech J. volume 27 pp 623- 656 , October 1968.

[4] *L. R. Rabiner and B. H. Juanng Fundamental of speech recognition, Prentice hall, Englewood cloiffs, N.J, 1993.*

[5] L. R. Rabiner and R. W. Schafer, Digital processing of speech signal, Prentice hall , Englewood cliffs NJ , 1978.

[6] Jesus savage, Carlos Rivera, Vanessa Aguilar :*Jsolated word speech regognition using vector quantization technique and artificial neural network*

[7] Joseph Picone , *Signal modeling technique in speech recognition*, Texas instruments system and information science laboratory Tsukuba

[8] Data sheet of AVR 128

[9] AVR Freaks tutorials, www.avrfreaks\forums\tutorials.