Vol:1 Issue:1 ISSN 2278 - 215X

Microcontroller Implementation of a Voice Command Recognition System for Human Machine Interface in Embedded System

Mrs.Sunpreet Kaur Nanda[1], Prof. Nilesh N. Kasat[2]

[1] PG Student, Department of Electronics and Telecommunication,Sipna's College of Engineering & Technology, Amravati, Maharashtra , India, sunpreetkaurbedi@gmail.com
[2] Assistant Professor, Department of Electronics and Telecommunication ,SIPNA's College of Engg. And Technology, Amravati, Maharashtra , India, knilesh123@rediffmail.com
Affiliated to S.G. B. Amravati University, Amravati. Maharashtra State, India.

***Abstract* — The speech recognition system is a completely assembled and easy to use programmable speech recognition circuit. Programmable, in the sense that the words (or vocal utterances) you want the circuit to recognize can be trained. This board allows you to experiment with many facets of speech recognition technology. It has 8 bit data out which can be interfaced with any microcontroller (ATMEL/PIC) for further development. Some of interfacing applications that can be made are authentication, controlling mouse of your computer and hence many other devices connected to it, controlling home appliances, robotics movements, speech assisted technologies, speech to text translation and many more.**

Keywords : MATLAB, TRAIN

I. Introduction

Speech recognition will become the method of choice for controlling appliances, toys, tools and computers. At its most basic level, speech controlled appliances and tools allow the user to perform parallel tasks (i.e. hands and eyes are busy elsewhere) while working with the tool or appliance. The heart of the circuit is the HM2007 speech recognition IC. The IC can recognize 20 words, each word a length of 1.92 seconds.

This document is based on using the Speech recognition kit for operating the mouse in manual mode using MATLAB codes. The problem in using this circuit was the installing the drivers of the mouse. Hence we have used MATLAB coding to interface the mouse. This appendum is giving our experience in solving the problems when operating the HM2007 in manual mode. A generic implementation of a HM2007 driver is appended as reference.[12]

II. Overview

The keypad and digital display are used to communicate with and program the HM2007 chip. The keypad is made up of 12 normally open momentary contact switches. When the circuit is turned on, "00" is on the digital display, the red LED (READY) is lit and the circuit waits for a command[23]

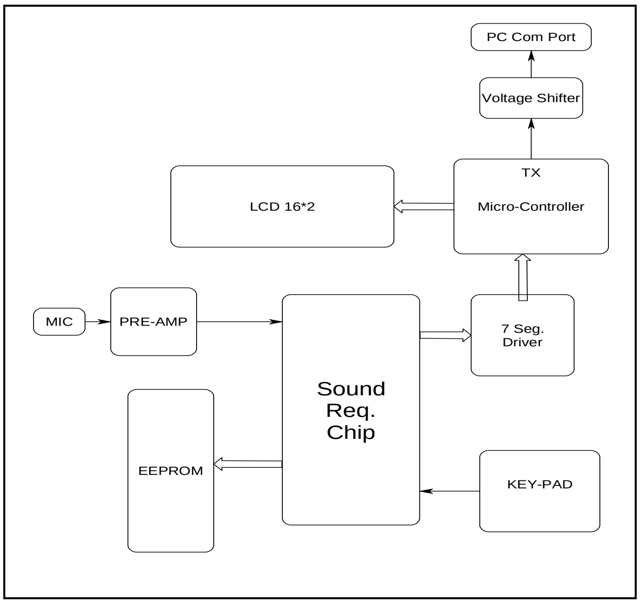

Figure 1 : Basic Block Diagram

1. Training Words for Recognition

Press "1" (display will show "01" and the LED will turn off) on the keypad, then press the TRAIN key (the LED will turn on) to place circuit in training mode, for word one. Say the target word into the headset microphone clearly. The circuit signals acceptance of the voice input by blinking the LED off then on. The word (or utterance) is now identified as the "01" word. If the LED did not flash, start over by pressing "1" and then "TRAIN" key. You may continue training new words in the circuit. Press "2" then TRN to train the second word and so on. The circuit will accept and recognize up to 20 words (numbers 1 through 20). It is not necessary to train all word spaces. If you only require 10 target words that's all you need to train.

2. Testing Recognition:

Repeat a trained word into the microphone. The number of the word should be displayed on the digital display. For instance, if the word "directory" was trained as word number 20, saying the word "directory" into the microphone will cause the number 20 to be displayed[5].

3. Error Codes:

The chip provides the following error codes.

55 = word to long
66 = word to short
77 = no match

4. Clearing Memory

To erase all words in memory press "99" and then "CLR". The numbers will quickly scroll by on the digital display as the memory is erased[11].

5. Changing & Erasing Words

Trained words can easily be changed by overwriting the original word. For instances suppose word six was the word "Capital" and you want to change it to the word "State". Simply retrain the word space by pressing "6" then the TRAIN key and saying the word "State" into the microphone. If one wishes to erase the word without replacing it with another word press the word number (in this case six) then press the CLR key.Word six is now erased.

6. Voice Security System

This circuit isn't designed for a voice security system in a commercial application, but that should not prevent anyone from experimenting with it for that purpose. A common approach is to use three or four keywords that must be spoken and recognized in sequence in order to open a lock or allow entry[13].

III. More On The HM2007 Chip

The HM2007[25] is a CMOS voice recognition LSI (Large Scale Integration) circuit. The chip contains an analog front end, voice analysis, regulation, and system control functions. The chip may be used in a stand alone or CPU connected.

Features:

- Single chip voice recognition CMOS LSI
- Speaker dependent
- External RAM support
- Maximum 40 word recognition (.96 second)
- Maximum word length 1.92 seconds (20 word)
- Microphone support
- Manual and CPU modes available
- Response time less than 300 milliseconds
- 5V power supply

The speech recognition system is speaker dependant, meaning that the voice that trained the system has the highest recognition accuracy. But you can simulate independent speech recognition.

To make the recognition system simulate speaker independence one uses more than one word space for each target word. Now we use four word spaces per target word.

a. Pin Configuration :

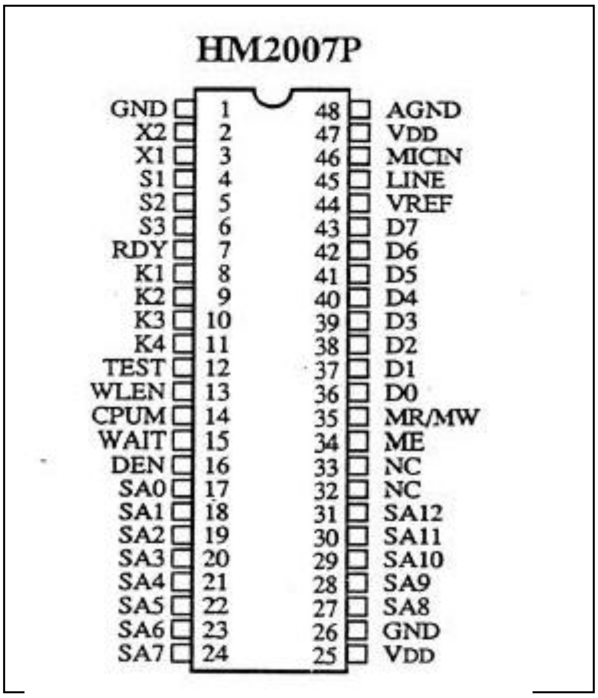

Figure 2 : Pin configuration of HM2007P

IV. Coding

```
void main() {
  TRISB = 0xFF;                    // Set PORTB as input
 Usart_Init(9600);// Initialize UART module at 9600
 Delay_ms(100);// Wait for UART module to stabilize
  Usart_Write('S');
  Usart_Write('T');
  Usart_Write('A');
  Usart_Write('R');
  Usart_Write('T');          // and send data via UART
  Usart_Write(13);
 while(1)
 {
  if(PORTB==0x01)
  {
    Usart_Write('1');     // and send data via UART
    while(PORTB==0x01)
     {}
  }
  if(PORTB==0x02)
  {
           Usart_Write('2'); // and send data via UART
            while(PORTB==0x02)
             {}
  }
  if(PORTB==0x03)
  {
        Usart_Write('3');      // and send data via UART
         while(PORTB==0x03)
             {}
  }
```

```
if(PORTB==0x04)
 {
       Usart_Write('4');     // and send data via UART
      while(PORTB==0x04)
            {}
}
if(PORTB==0x05)
 {
      Usart_Write('5');      // and send data via UART
     while(PORTB==0x05)
            {}
 }
if(PORTB==0x06)
 {
     Usart_Write('6');       // and send data via UART
     while(PORTB==0x06)
            {}
 }
if(PORTB==0x07)
 {
      Usart_Write('7');     // and send data via UART
    while(PORTB==0x07)
            {}
 }
 if(PORTB==0x08)
 {
      Usart_Write('8');      // and send data via UART
   while(PORTB==0x08)
            {}
 }
 if(PORTB==0x09)
 {
      Usart_Write('9');      // and send data via UART
 while(PORTB==0x09)
            {}
 }
  if(PORTB==0x0A)
 {
     Usart_Write('1');      // and send data via UART
      Usart_Write('0');
      while(PORTB==0x0A)
            {}
 }
  if(PORTB==0x0B)
 {
        Usart_Write('1');   // and send data via UART
        Usart_Write('1');
      while(PORTB==0x0B)
   {}
 }
 if(PORTB==0x0C)
 {
        Usart_Write('1');  // and send data via UART
        Usart_Write('2');   // and send data via UART
        while(PORTB==0x0C)
            {}
   }
   if(PORTB==0x0D)
   {

        Usart_Write('1');    // and send data via UART
        Usart_Write('3');    // and send data via UART
     while(PORTB==0x0D)
            {}
   }
   if(PORTB==0x0E)
   {
        Usart_Write('1');     / and send data via UART
        Usart_Write('4');    // and send data via UART
     while(PORTB==0x0E)
            {}
   }
    if(PORTB==0xFF)
    {
        Usart_Write('1');   // and send data via UART
        Usart_Write('5');   // and send data via UART
    while(PORTB==0xFF)
            {}
    }
   }
 }
```

V. Result

Experiment conducted in following circumstances

1. The room should be sound proof.
2. Humidity should not be greater than 50%.
3. The person speaking, should be close to microphone.
4. It does not depend on temperature.

VI. Analysis

1. Sensitivity of system:-
   Testing signal :- impulse of 10 Hz
   Testing on : DSO 100 Hz bandwidth
   Response time :- 50 ms

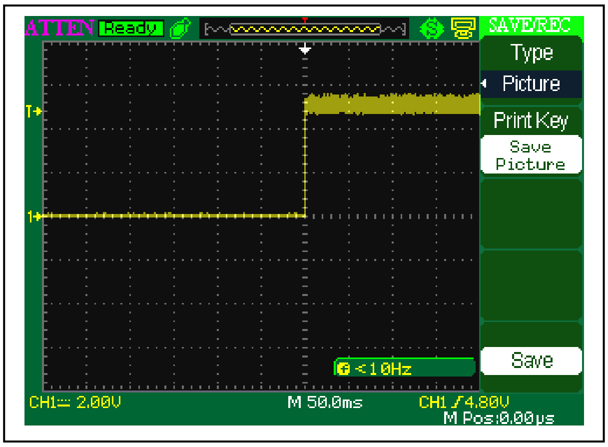

Figure 3 : Impulse response graph

2. Sinusoidal Response :-
   Testing signal :- sinusoidal of 3.9999 Mhz
   Response time :- 10 ms

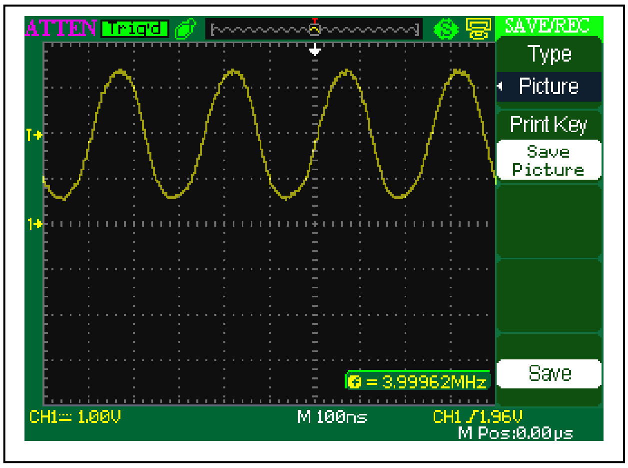

Figure 4 : Rresponse graph

3. Voice recognition analysis
   Testing Signal :- human voice
   Error produced:- 0.4 %
   Response time :- 15 ms

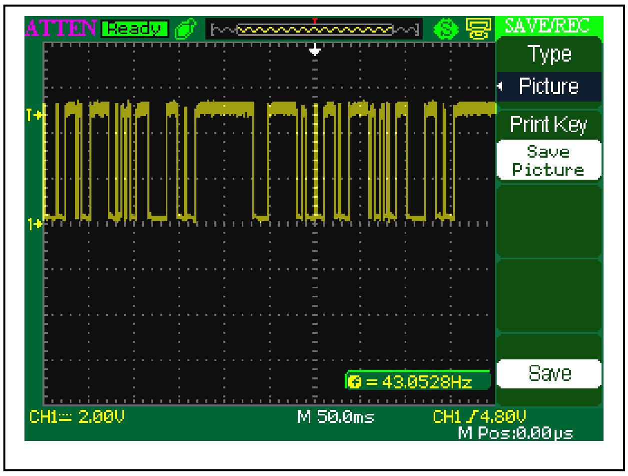

Figure 5 : Voice recognition

VII. Conclusion

From above explanation we conclude that HM2007 can be used to detect voice signals accurately. After detecting voice signals these can be used to operate the mouse as explained earlier. Thus, we can implement microcontroller in voice recognition system for human machine interface in embedded system. The MATLAB coding is not specified in this document but without its use mouse cannot be operated.

VIII. References

1. H. Dudley, *The Vocoder*, Bell Labs Record, Vol. 17, pp. 122-126, 1939.
2. H. Dudley, R. R. Riesz, and S. A. Watkins, *A Synthetic Speaker*, J. Franklin Institute, Vol. 227, pp. 739-764, 1939.
3. J. G. Wilpon and D. B. Roe, *AT&T Telephone Network Applications of Speech Recognition*, Proc. COST232 Workshop, Rome, Italy, Nov. 1992.
4. C. G. Kratzenstein, *Sur la raissance de la formation des voyelles*, J. Phys., Vol 21, pp. 358-380, 1782.
5. H. Dudley and T. H. Tarnoczy, *The Speaking Machine of Wolfgang von Kempelen*, J. Acoust. Soc. Am., Vol. 22, pp. 151-166, 1950.
6. Sir Charles Wheatstone, *The Scientific Papers of Sir Charles Wheatstone*, London: Taylor and Francis, 1879.
7. J. L. Flanagan, *Speech Analysis, Synthesis and Perception*, Second Edition, Springer-Verlag, 1972.
8. H. Fletcher, *The Nature of Speech and its Interpretations*, Bell Syst. Tech. J., Vol 1, pp. 129-144, July 1922.
9. K. H. Davis, R. Biddulph, and S. Balashek, *Automatic Recognition of Spoken Digits*, J. Acoust. Soc. Am., Vol 24, No. 6, pp. 627-642, 1952.
10. H. F. Olson and H. Belar, *Phonetic Typewriter*, J. Acoust. Soc. Am., Vol. 28, No. 6, pp. 1072-1081, 1956.
11. J. W. Forgie and C. D. Forgie, *Results Obtained from a Vowel Recognition Computer Program*, J. Acoust. Soc. Am., Vol. 31, No. 11, pp. 1480-1489, 1959.
12. J. Suzuki and K. Nakata, *Recognition of Japanese Vowels—Preliminary to the Recognition of Speech*, J. Radio Res. Lab, Vol. 37, No. 8, pp. 193-212, 1961.
13. J. Sakai and S. Doshita, *The Phonetic Typewriter*, Information Processing 1962, Proc. IFIP Congress, Munich, 1962.
14. K. Nagata, Y. Kato, and S. Chiba, *Spoken Digit Recognizer for Japanese Language*, NEC Res. Develop., No. 6, 1963.
15. D. B. Fry and P. Denes, *The Design and Operation of the Mechanical Speech Recognizer at University College London*, J. British Inst. Radio Engr., Vol. 19, No. 4, pp. 211-229, 1959.
16. T. B. Martin, A. L. Nelson, and H. J. Zadell, *Speech Recognition by Feature Abstraction Techniques*, Tech. Report AL-TDR-64-176, Air Force Avionics Lab, 1964.
17. T. K. Vintsyuk, *Speech Discrimination by Dynamic Programming*, Kibernetika, Vol. 4, No. 2, pp. 81-88, Jan.-Feb. 1968.
18. H. Sakoe and S. Chiba, *Dynamic Programming Algorithm Quantization for Spoken Word Recognition*, IEEE Trans. Acoustics, Speech and Signal Proc., Vol. ASSP-26, No. 1, pp. 43-49, Feb. 1978.
19. A. J. Viterbi, *Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm*, IEEE Trans. Informaiton Theory, Vol. IT-13, pp. 260-269, April 1967.

20. B. S. Atal and S. L. Hanauer, *Speech Analysis and Synthesis by Linear Prediction of the Speech Wave*, J. Acoust. Soc. Am. Vol. 50, No. 2, pp. 637-655, Aug. 1971.
21. F. Itakura and S. Saito, *A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies,* Electronics and Communications in Japan, Vol. 53A, pp. 36-43, 1970.
22. F. Itakura, *Minimum Prediction Residual Principle Applied to Speech Recognition*, IEEE Trans. Acoustics, Speech and Signal Proc., Vol. ASSP-23, pp. 57-72, Feb. 1975.
23. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg and J. G. Wilpon, *Speaker Independent Recognition of Isolated Words Using Clustering Techniques*, IEEE Trans. Acoustics, Speech and Signal Proc., Vol. Assp-27, pp. 336-349, Aug. 1979.
24. B. Lowerre, *The HARPY Speech Understanding System*, Trends in Speech Recognition, W. Lea, Editor, Speech Science Publications, 1986, reprinted in Readings in Speech Recognition, A. Waibel and K. F. Lee, Editors, pp. 576-586, Morgan Kaufmann Publishers, 1990.
25. M. Mohri, *Finite-State Transducers in Language and Speech Processing,* Computational Linguistics, Vol. 23, No. 2, pp. 269-312, 1997.
26. Dennis H. Klatt, *Review of the DARPA Speech Understanding Project (1)*, J. Acoust. Soc. Am., 62, 1345-1366, 1977.
27. F. Jelinek, L. R. Bahl, and R. L. Mercer, *Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech*, IEEE Trans. On Information Theory, Vol. IT-21, pp. 250-256, 1975.
28. C. Shannon, *A mathematical theory of communication*, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
29. S. K. Das and M. A. Picheny, Issues in practical large vocabulary isolated word recognition: The IBM Tangora system, in *Automatic Speech and Speaker Recognition Advanced Topics*, C.H. Lee, F. K. Soong, and K. K. Paliwal, editors, p. 457-479, Kluwer, Boston, 1996.
30. B. H. Juang, S. E. Levinson and M. M. Sondhi, *Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains,* IEEE Trans. Information Theory, Vol. It-32, No. 2, pp. 307-309, March 1986.