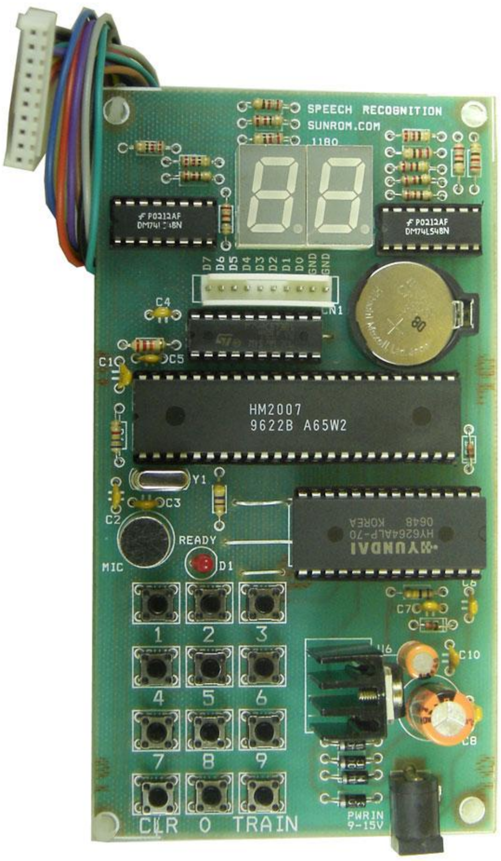
SUNROM TECHNOLOGIES
Your Source for Embedded Systems

Email: info@sunrom.com or sunrom@gmail.com

Visit us at http://www.sunrom.com

Document: Datasheet Date: 2-Oct-09 Model #: 1180 Product's Page: www.sunrom.com/p-762.html

Speech Recognition System



The speech recognition system is a completely assembled and easy to use programmable speech recognition circuit. Programmable, in the sense that you train the words (or vocal utterances) you want the circuit to recognize. 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 for further development. Some of interfacing applications which can be made are controlling home appliances, robotics movements, Speech Assisted technologies, Speech to text translation, and many more.

Features

- Self-contained stand alone speech recognition circuit
- User programmable
- Up to 20 word vocabulary of duration two second each
- Multi-lingual
- Non-volatile memory back up with 3V battery onboard. Will keep the speech recognition data in memory even after power off.
- Easily interfaced to control external circuits & appliances

Specification

Parameter	Value	Note
Input Voltage	9 to 15 V DC	Use a commonly available 12V 500ma DC Adapter
Output Data	8 bits at 5V Logic Level	Any microcontroller like 8051, PIC or AVR can be interfaced to data port to interpret and implement specialized applications

Applications

There are several areas for application of voice recognition technology.

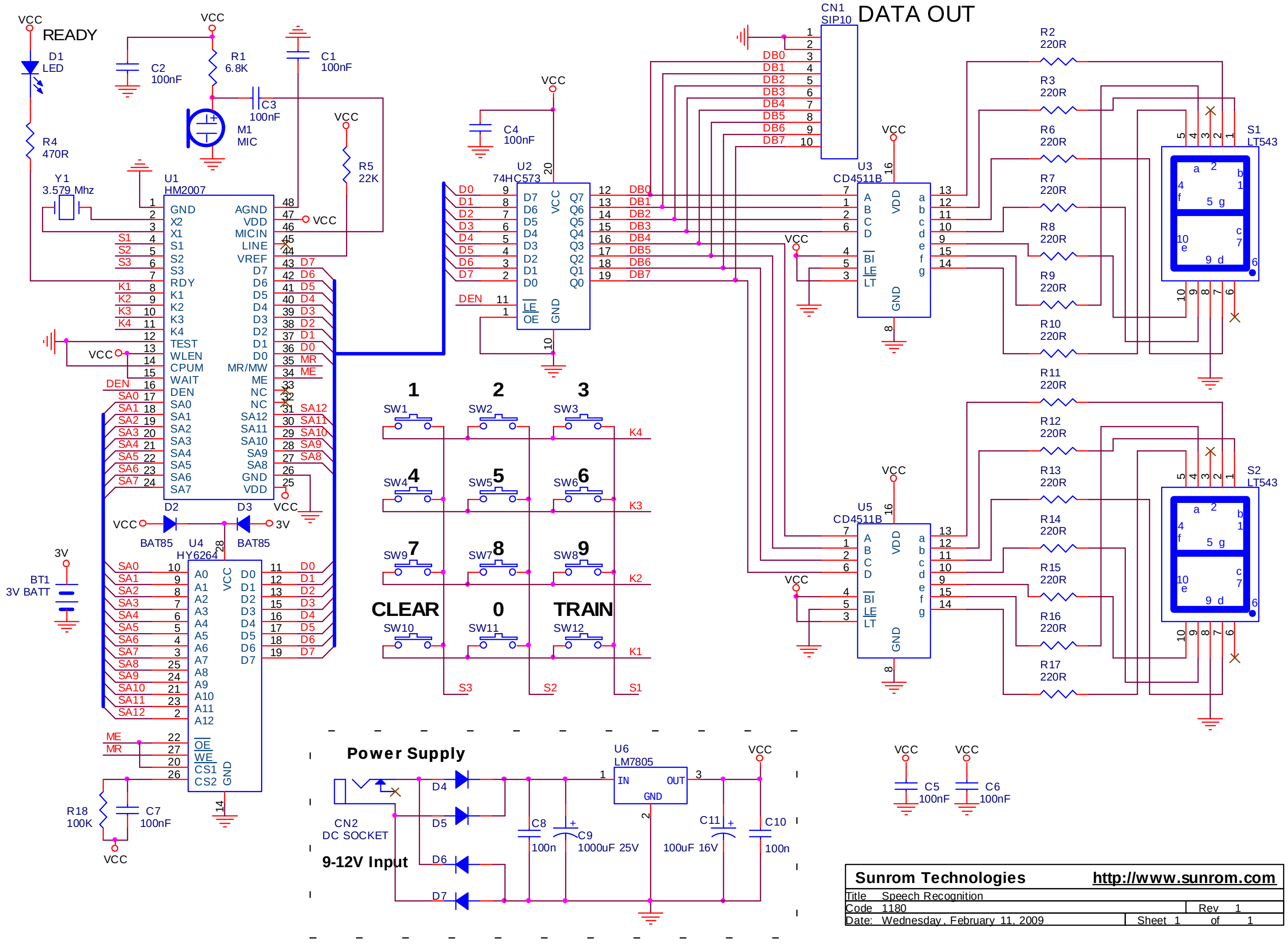
- Speech controlled appliances and toys
- Speech assisted computer games
- Speech assisted virtual reality
- Telephone assistance systems
- Voice recognition security
- Speech to speech translation

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.

Complete Schematic of System



Using the System

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.

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 onboard microphone (near LED) 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.

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.

Error Codes:
The chip provides the following error codes.

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

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.

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.

Simulated Independent Recognition
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. Therefore we obtain four different enunciation’s of each target word. (speaker independent). The word spaces 01, 02, 03 and 04 are allocated to the first target word. We continue do this for the remaining word space. For instance, the second target word will use the word spaces 05, 06, 07 and 08. We continue in this manner until all the words are programmed.

If you are experimenting with speaker independence use different people when training a target word. This will enable the system to recognize different voices, inflections and enunciation's of the target word. The more system resources that are allocated for independent recognition the more robust the circuit will become.

If you are experimenting with designing the most robust and accurate system possible, train target words using one voice with different inflections and enunciation's of the target word.

Homonyms

Homonyms are words that sound alike. For instance the words cat, bat, sat and fat sound alike. Because of their like sounding nature they can confuse the speech recognition circuit. When choosing target words for your system do not use homonyms.

The Voice with Stress & Excitement

Stress and excitement alters ones voice. This affects the accuracy of the circuit's recognition. For instance assume you are sitting at your workbench and you program the target words like fire, left, right, forward, etc., into the circuit. Then you use the circuit to control a flight simulator game, Doom or Duke Nukem. Well, when you're playing the game you'll likely be yelling "FIRE! …Fire! ...FIRE!! ...LEFT …go RIGHT!". In the heat of the action you're voice will sound much different than when you were sitting down relaxed and programming the circuit. To achieve a higher accuracy word recognition one needs to mimic the excitement in ones voice when programming the circuit.

These factors should be kept in mind to achieve the high accuracy possible from the circuit. This becomes increasingly important when the speech recognition circuit is taken out of the lab and put to work in the outside world.

Error Codes

When interfacing the external circuit through its data bus, The decoding circuit must recognize the word numbers from error codes. So the circuit must be designed to recognize error codes 55, 66 and 77 and not confuse them with word spaces 5, 6 and 7.

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.

Aural Interfaces

It's been found that mixing visual and aural information is not effective. Products that require visual confirmation of an aural command grossly reduces efficiency. To create an effective AUI products need to understand (recognize) commands given in an unstructured and efficient methods. The way in which people typically communicate verbally.

Learning to Listen

The ability to listen to one person speak among several at a party is beyond the capabilities of today's speech recognition systems. Speech recognition systems can not (as of yet) separate and filter out what should be considered extraneous noise.

Speech recognition is not understanding speech. Understanding the meaning of words is a higher intellectual function. Because a circuit can respond to a vocal command doesn't mean it understands the command spoken. In the future, voice recognition systems may have the ability to distinguish nuances of speech and meanings of words, to "Do what I mean, not what I say!"

Speaker Dependent / Speaker Independent

Speech recognition is divided into two broad processing categories; speaker dependent and speaker independent.

Speaker dependent systems are trained by the individual who will be using the system. These systems are capable of achieving a high command count and better than 95% accuracy for word recognition. The drawback to this approach is that the system only responds accurately only to the individual who trained the system. This is the most common approach employed in software for personal computers.

Speaker independent is a system trained to respond to a word regardless of who speaks. Therefore the system must respond to a large variety of speech patterns, inflections and enunciation's of the target word. The command word count is usually lower than the speaker dependent however high accuracy can still be maintain within processing limits. Industrial applications more often require speaker independent voice recognition systems.

Recognition Style

In addition to the speaker dependent/independent classification, speech recognition also contends with the style of speech it can recognize. They are three styles of speech: isolated, connected and continuous.

Isolated: Words are spoken separately or isolated. This is the most common speech recognition system available today. The user must pause between each word or command spoken.

Connected: This is a half way point between isolated word and continuous speech recognition. It permits users to speak multiple words. The HM2007 can be set up to identify words or phrases 1.92 seconds in length. This reduces the word recognition dictionary number to 20.

Continuous: This is the natural conversational speech we use to in everyday life. It is extremely difficult for a recognizer to sift through the sound as the words tend to merge together. For instance, "Hi, how are you doing?" to a computer sounds like "Hi,.howyadoin" Continuous speech recognition systems are on the market and are under continual development.

More On The HM2007 Chip

The HM2007 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:

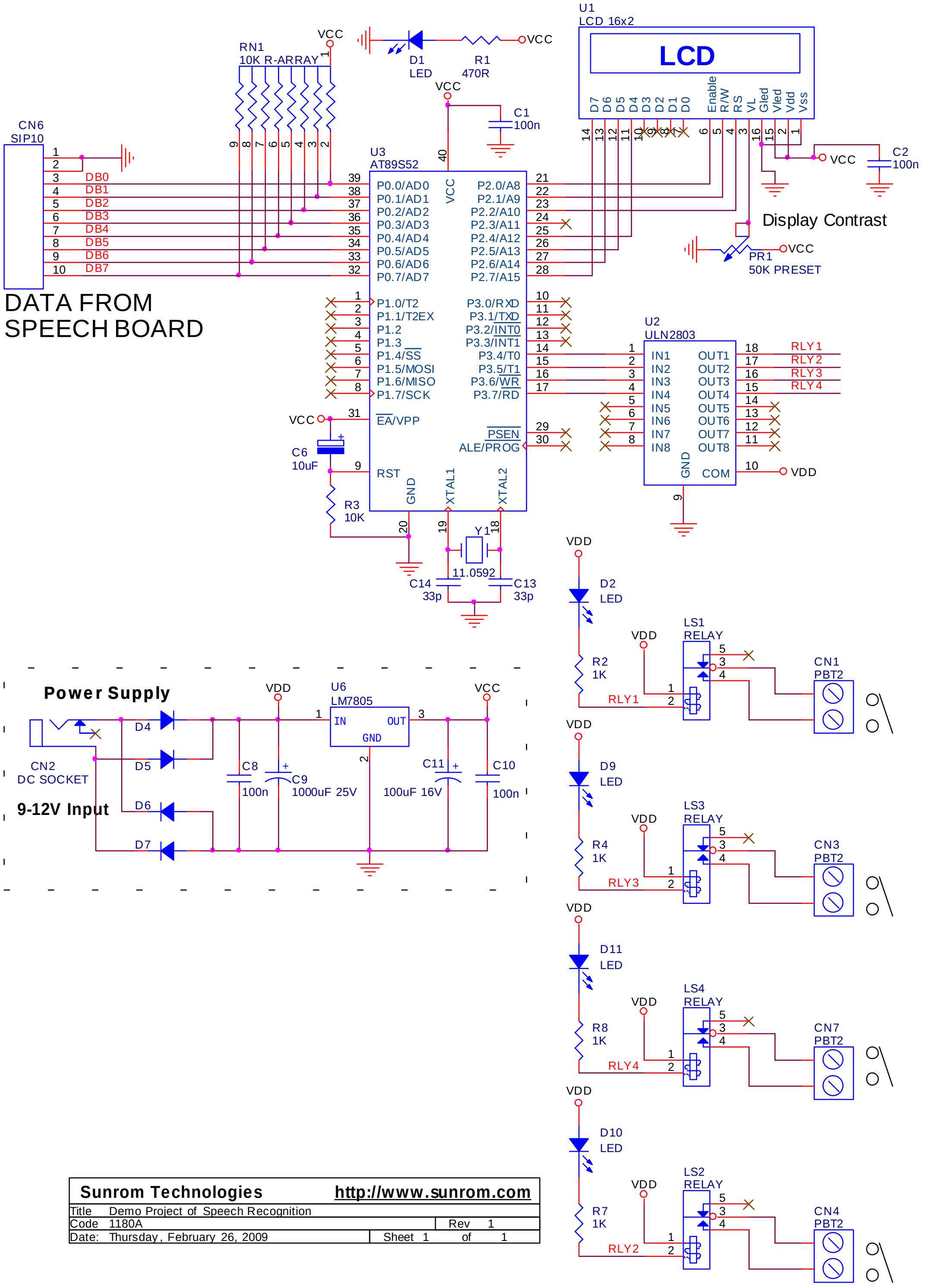
- Single chip voice recognition CMOS LSI
- Speaker dependent
- External RAM support
- Maximum 40 word recognition (.96 second)
- Maximum word length 1.92 seconds (20 word)
- Microphone support
- Manual and CPU modes available
- Response time less than 300 milliseconds
- 5V power supply

More information on the HM2007 chip is available in the HM2007 data booklet (DS-HM2007) which can be downloaded below.
http://www.sunrom.com/files/HM2007.pdf

Interfacing external circuits through data bus

This sample project will show how a circuit can be interfaced through the data bus of speech recognition circuit. It will show messages and error codes on LCD. It will also operate four relays as per data from speech circuit.

Schematic of interfacing project



Sample Code of interfacing project

```c
//main.c
#include <REGX51.H> // standard 8051 defines
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=- Include files -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
#include "lcd.h"
#include "utils.h"
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=- Hardware Defines -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
sfr DATA = P0;
sbit OUT1 = P3^4;
sbit OUT2 = P3^5;
sbit OUT3 = P3^6;
sbit OUT4 = P3^7;

// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=- Variables -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
char buf[20];
char code M1[] ="SPEECH: ONE";
char code M2[] ="SPEECH: TWO";
char code M3[] ="SPEECH: THREE";
char code M4[] ="SPEECH: FOUR";
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=- Main Program -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
void main()
{
      unsigned char lastdata, datanow;
      // -=-=- Intialize variables -=-=-=
      OUT1 = 0;
      OUT2 = 0;
      OUT3 = 0;
      OUT4 = 0;
      // -=-=- Intialise -=-=-=
      lcdInit();
      // -=-=- Welcome LCD Message -=-=-=
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint("Speech Test");
      lcdGotoXY(0,1); // 2nd Line of LCD
      lcdPrint("System");
      delayms(5000); // 5 sec
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint("Train: 1-4 key >");
      lcdGotoXY(0,1); // 2nd Line of LCD
      lcdPrint("Train>Speak Now");
      // -=-=- Program Loop -=-=-=
      lastdata=0xff;
      while(1)
      {
            datanow=DATA; // read data from speech board

            if(lastdata!=datanow) // if there is new data then,
            {
```

```
lastdata=datanow;
switch(lastdata)
{
case 0x55:
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint("Speech too Long");
      lcdGotoXY(0,1); // 2nd Line of LCD
      lcdPrint("Try Again!");
break;
case 0x66:
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint("Speech too Short");
      lcdGotoXY(0,1); // 2nd Line of LCD
      lcdPrint("Try Again!");
break;
case 0x77:
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint("No Match");
      lcdGotoXY(0,1); // 2nd Line of LCD
      lcdPrint("Try Again!");
break;
case 0x01:
      if(OUT1==1)
            OUT1 = 0;
      else
            OUT1 = 1;
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint(M1);
break;
case 0x02:
      if(OUT2==1)
            OUT2 = 0;
      else
            OUT2 = 1;
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint(M2);
break;
case 0x03:
      if(OUT3==1)
            OUT3 = 0;
      else
            OUT3 = 1;
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint(M3);
break;
case 0x04:
      if(OUT4==1)
            OUT4 = 0;
      else
            OUT4 = 1;
      lcdClear();
      lcdGotoXY(0,0); // 1st Line of LCD
      lcdPrint(M4);
```

```
                  break;
                  }
            }
      }
}
```