Embedded Voice Messenger with Password Protection

Supriya Vaidya[1]; Jyoti Yadav[2]; Faizan Siddiqui[3]; Kamlakar singh[4]
Department of Electronics Engineering,
K.C. College of Engineering & Management studies & Research,
Kopri,Thane(E)-400 603, India.

***Abstract* — The purpose of this project is to implement the various concepts of microcontroller and embedded designing environment. Our project "EMBEDDED VOICE MESSENGER WITH PASSWORD PROTECTION" fully utilizes microcontroller futures &embedded technology concepts to minimize the complication of digital gates, size and cost too. This system can store 8/16 minutes of voice and manages data reprogrammable voice IC. User can store as many tracks he wants of any length. This project would develop keeping time and ease of human ability. It also gives future provision to users. This project would be useful for spy purpose and fan operators.**

Keywords—8051microcontroller, LCD display, voice IC, Digital Gates,

I. INTRODUCTION

This system can store 8/16 minutes of voice and manages data reprogrammable IC. User can store as many tracks he wants of any length.Voice is being recorded and stored in the microcontroller via this voice IC through microphone amplifier and then this voice IC transmit all the recorded data as it is to the speaker amplifier and then to speaker to get the voice of the entered password for the protection purpose.

Initially from the transformer all the input power is being sent to the power supply and via power supply to the microcontroller, LCD display and voice IC, 5V input power is distributed. Keys are pressed to get the recorded password from microcontroller on the screen and display that particular password. There is an interfacing between keypad - microcontroller and LCD – microcontroller. Then via microphone the amplification of voice is done and recorded voice from the microcontroller is given out from voice IC to the speaker amplifier and finally the whole voice messenger gives the recorded voice data from Microphone with displaying password.[1] The embedded system is real time technology for computing constraints and special-purpose computer system designed to perform, a few dedicated functions. It is being embedded as a part of a whole device which includes hardware and mechanical parts also. [2]In contrast our Personal Computer can do many different jobs depending on computing constraints and programming. [2]Embedded systems have become very important for today as they are being controlling many of the common devices of we use in our day to day life. [3] Since the embedded system is dedicated to design engineers who can optimize it and reduces the size and reduces cost of the product, or increases the reliability and also increases the performance of the system.

Physically, an embedded system ranges from devices which are portable such as digital watches, MP3 players and extends to large stationary installations such as traffic lights, factory controlled plants also the systems controlling nuclear power plants. Also their Complexity varies from the very low in size, with a so small single microcontroller chip comparing very high multiple units and networks mounted inside the enclosure.[3]For example, handheld computers share a few elements with embedded systems because they allow different applications to be loaded also they connect different peripherals. An embedded system is nothing but the combination of both computer hardware and software which are either fixed in capability or it can be fixed in its programmability. [5]Industrial machines, automobiles medical equipment, cameras, household appliances, airplanes, vending machines, toys are many examples of an embedded system.

Certain operating systems or language platforms such as a java script in embedded system and windows XP embedded are low-end consumer products which are very inexpensive and they are having limited storage.

II. HARDWARE ARCHITECHTURE AND IMPLEMENTATION

In the interfacing of the keyboard here we are interfacing is a matrix keyboard. This key board is designed with a particular rows and columns. These rows and columns are connected to the microcontroller through its ports of the micro controller 8051. We normally use 8*8 matrix keyboard. So only two ports of 8051 can be easily connect to the rows and columns of the keyboard. Whenever a key is pressed, a row and a column gets shorted through that pressed key and all the other keys are left open. When a key is pressed next our aim is to identify that key. To do this we firstly check for particular row and then we check the corresponding column of the keyboard. To check the row of the pressed key in the keyboard, one of these rows are made high by making one of the bit in the output port of 8051 high. This is done until the row is found out once we get the row next our job is to find out column of the pressed key. The column is detected by the contents in the input port is rotated with carry until the carry bit is set. The contents of the counter is then compared and displayed in the display. This display is designed using a seven segment display and BCD to seven segment decoder IC 7447. The BCD equivalent number of counter is sent through output port of 8051

displays the number of pressed keys. The circuit diagram is as shown in Fig 1.1.

1) 8051 Architechture and its specifications:

With 64K Program Memory and data memory,128 bytes of on chip data memory, EPROM of size 4K X 8 and APR6016/6008. [5]The generic 8051 architecture supports a Harvard architecture, which contains two separate buses for both program and data. So, it has two distinctive memory spaces of 64K X 8 size for both programmed and data.Each 8051 device has some amount of data RAM built in the device for internal processing. This area is used for stack operation and temporary storage of the data.This bus architecture is supported with on-chip peripheral functions like I/O ports, timers/counters, versatile serial communication port.So it is clear that 8051 architecture was designed to cater many real time embedded needs.

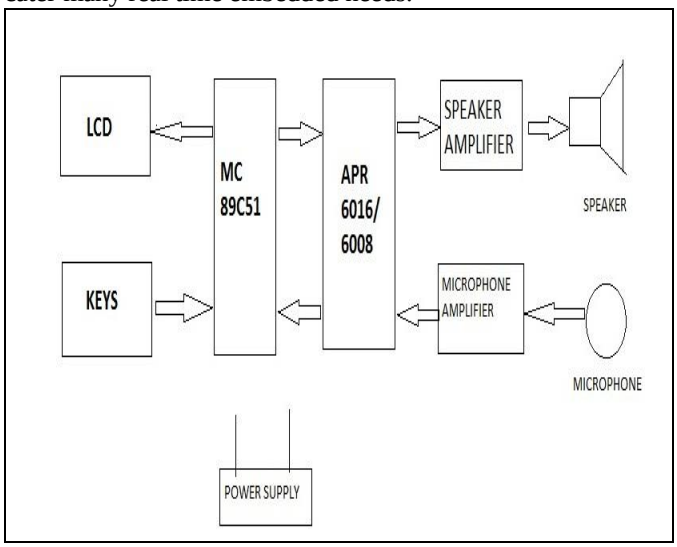


Fig 1: Functional block diagram

WORKING:
Transformer gives 230volts of the input power to the power supply and the power supply gives 5v to the microcontroller AT89C51, LCD 16*2 and voice IC APR6016, 12V power to the speaker amplifier. As soon as keys are pressed to save the password gets input to the microcontroller to record it and displays the password on the screen. Microphone amplifier gets voice via microphone and sends it to the voice IC. This IC is interfaced to the microphone 8051 AT89C51 then via microphone voice is being recorded and amplified voice is being sent. Power supply distributes the power to the entire blocks of the microcontroller 8051, LCD 16*2, voice IC and speaker amplifier LM386.

2) Regulated power supply:

A Variable regulated power supply, also called a variable bench power supply, is one where you can continuously adjust the output voltage to your requirements. Varying the output of the power supply is the recommended way to test a project after having double checked parts placement against circuit drawing and the parts placement guide. This type of regulation is ideal for having a simple variable bench power supply. Actually this is quite important because one of the first projects a hobbyist should undertake is the construction of a variable regulated power supply. While a dedicated supply is quite handy e.g. 5V or 12V, it's much handier to have a variable supply on hand, especially for testing.

Most digital logic circuits and processors need a 5 volt power supply. To use these parts we need to build a regulated 5 volt source. Usually you start with an unregulated power To make a 5 volt power supply, we use a LM7805 voltage regulator IC

3) Liquid crystal display:

Liquid crystal display displays temperature of the measured element, which is calculated by the microcontroller also the CMOS technology makes the device ideal for application in hand held, portable and other battery instruction with low power consumption.LCD display can be interfaced with microcontroller to read the output directly. In our project we use a two line LCD display with 16 characters each.

4) Keypad Theory:

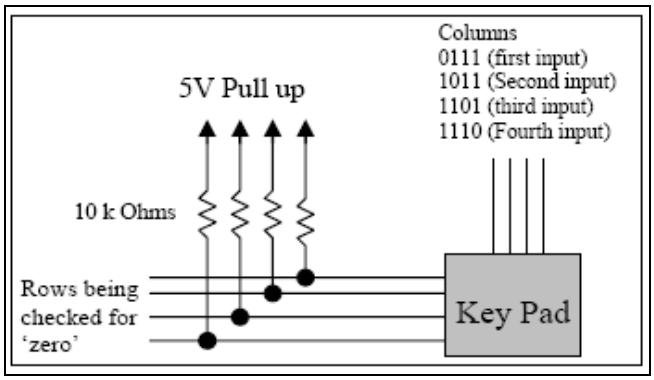


Fig 1.1 Keypad connections

Keypad is basically a matrix of many rows and columns. The number of pin outs of keyboard, with each pin attached to a particular, unique, entire row or entire column via a naked tiny conducting wire. . All the wires are properly insulated from each other, however, at the places where a wire for a column crosses a wire representing a row, there is a short air-spacing between the two, instead of some insulated material, so that the two wires do not touch each other at their own, but may be made to do so on key press. That is where we press the keys on the keypad. As we press a particular key on the keypad, the spacing between two wires reduces until the two wires, of one particular row and one particular column, touch each other. Different scanning techniques may be developed to determine which key is pressed.

KEYPAD SCANNING

The wires from four columns and four rows on the keypad form a full byte, and a particular port on the master 8051 may be dedicated for interfacing with keypad and scanning it [11]. In our configuration, the more significant nibble on this port is attached to columns portion of keypad, and is selected as the input nibble. The less significant nibble is for the four rows, which, normally are pulled up to logic "1", and if any

key is pressed on keypad, a particular pin in this nibble captures the data sent to a particular column. Our goal is to search that pin, and also to find out as to data of which pin has been captured there. The scanning scheme is elaborated below, and keypad connections are shown in Fig. 5.

The start is taken from column 0, logic 1 is sent on all other three columns and a 0 on this column 0 through the input nibble 0111.

The output nibble coming from the rows is checked for a zero on anyone pins, which are normally pulled up to a 1, and can reads a 0 only if any key in the column 0 is pressed. If no pin in the nibble read is 0, then the input nibble 0111 is shifted to right by one place to yield 1011, and is re-sent on columns, and rows are again checked for any zero. This idle process is continued till user gives some input to the keypad.

Each time we shift the input nibble, a particular variable already set for this purpose, is incremented, starting from zero for first column, to a maximum of three for the fourth column. It represents the column number in which logic 0 is detected, and is saved in some register, say R0, for future reference. At the moment any zero is received in any row, the value of the nibble coming from the rows is also stored in some another register, say R1. Now to find out that which the key pressed, shift R0 to the right through carry, using RRC instruction, and keep a check on carry, incrementing another variable by 4, every time the RRC instruction is used, until a 0 in the carry appears. At this moment, the variable incremented by 4 every time, would be reading the offset starting from the very first key on keypad to the start of this particular row. Now once the row has been detected and the variable incremented, it is incremented again by the value of the column number, R1 [12]. The ultimate answer we have at the end is the final value of the variable, which is the offset starting from zero, having a maximum possible value of 15 for a 4*4 matrix. Once the offset is obtained, the corresponding ASCII code is selected from the lookup table formulated at the end of the software of master controller. It is selected using the already calculated offset from the starting location of the lookup table. This ASCII code is sent to the data byte of LCD to displayed keypad.

5) Voice IC APR6016/6008:

This IC can store 8/16 minutes of voice and manages data reprogrammable IC. User can store as many tracks he wants of any length. Voice is being recorded and stored in the microcontroller via this voice IC through microphone amplifier and then this voice IC transmit all the recorded data as it is to the speaker amplifier and then to speaker to get the voice of the entered password for the protection purpose.

The APR6008 offers non-volatile storage of voice and/or data in advanced Multi-Level Flash memory. Up to 8 minutes of audio recording and playback can be accommodated. A maximum of 30K bits of digital data can be stored.

Devices can be cascaded for longer duration recording or greater digital storage. Device control is accomplished through an industry standard SPI interface that allows a microcontroller to manage message recording and playback. This flexible arrangement allows for the widest variety of messaging options. The APR6008 is ideal for use in cellular and cordless phones, telephone answering devices, personal digital assistants, personal voice recorders, and voice pagers. APLUS Integrated achieves this high level of storage capability by using a proprietary analog multi-level storage technology implemented in an advanced non-volatile Flash memory process. Each memory cell can typically store 256 voltage levels. This allows the APR6008 voice to reproduce audio signals in their natural form, eliminating the need for encoding and compression which can introduce distortion.

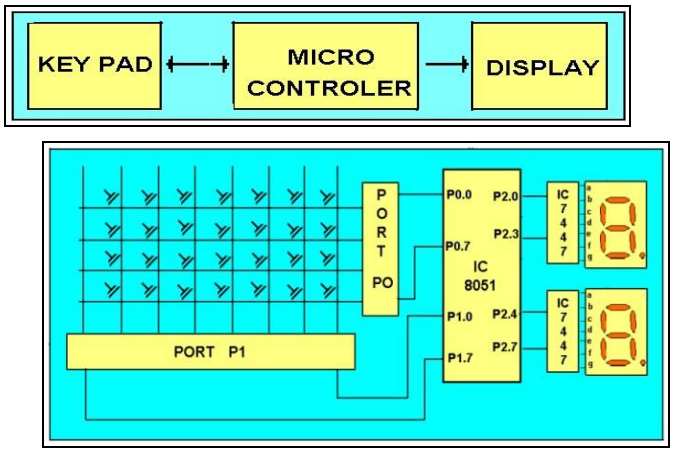


Fig.1.2 Circuit diagram of INTERFACING KEY BOARD TO 8051

1)The 8051 has 4 I/O ports P0 to P3 each with 8 I/O pins, P0.0 to P0.7, P1.0 to P1.7, P2.0 to P2.7, P3.0 to P3.7. The one of the port P1 (it understood that P1 means P1.0 to P1.7) as an I/P port for microcontroller 8051, port P0 as an O/P port of microcontroller 8051 and port P2 is used for displaying the number of pressed key.

2) Make all rows of port P0 high so that it gives high signal when key is pressed.

3) See if any key is pressed by scanning the port P1 by checking all columns for non zero condition.

4) If any key is pressed, to identify which key is pressed make one row high at a time.

5) Initiate a counter to hold the count so that each key is counted.

6) Check port P1 for nonzero condition. If any nonzero number is there in [accumulator], start column scanning by following step 9.

7) Otherwise make next row high in port P1.

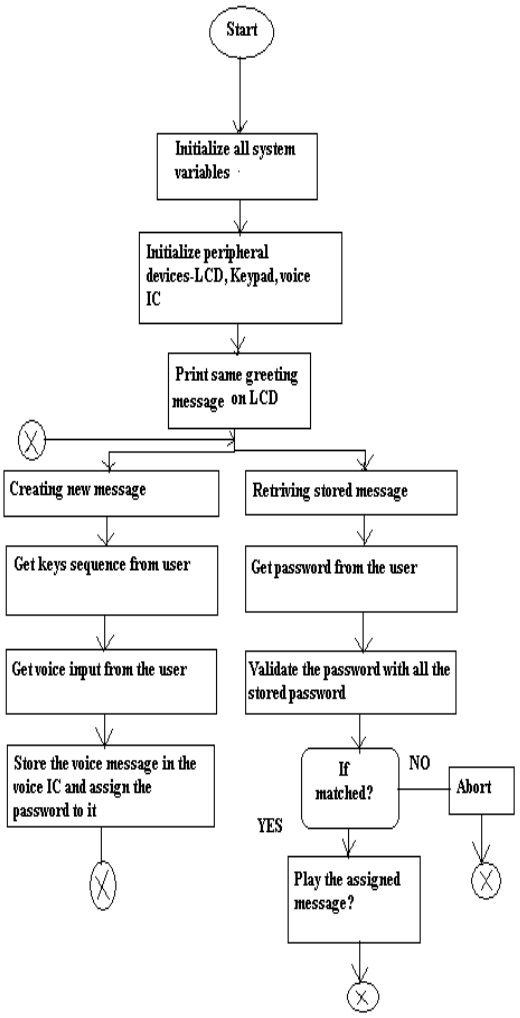
8) Add a count of 08h to the counter to move to the next row by repeating steps from step 6.

9)If any key pressed is found, the [accumulator] content is rotated right through the carry until carry bit sets, while doing this increment the count in the counter till carry is found.

10) Move the content in the counter to display in data field or to memory location

To repeat the procedures go to step 2.

III. SOFTWARE DEVELOPMENT



1] First we initialize all the system variables
2] Then initialize the peripheral devices like LCD, Keypad, voice IC
3] Next we print the same greeting message on the LCD screen.
4] Then comes two conditions those are whether it is to be created new message or the stored message is to be retrieved.
5] If there is new created message then it is our job to get the sequence from the user or get the password from the user.
6] Get the voice input from the user. In case of retrieving stored message then we validate the password with the stored password.
7] In creating new message next step is to store the voice message in the voice IC and assign the password to it.
8] In retrieving steps once we validate the password with stored one then next job is to match those two passwords and if yes the play the assigned message and if not then abort.

IV. CONCLUSIONS

This system can store 8/16 minutes of voice and manages data reprogrammable IC. User can store as many tracks he wants of any length. With recommended password system and ease of voice amplifier for amplification of recorded data, embedded technology can be the most promising future for the next generation.

REFERENCES

[1] Myke Predko, Programming and Customizing the 8051 Microcontroller, Edition 1999, Tata McGraw-Hill, Page: 157-167.
[2] Muhammad Ali Mazidi, Janice Gillispie Mazidi, 8051 Microcontroller and Embedded Systems, Prentice-Hall, Page: 183-193, 236, 243.
[3] Dogan Ibrahim, Microcontroller Projects in C for the 8051, Newnes, Page: 29-161.
[4] Kenneth J. Ayala, The 8051 Microcontroller ARCHITECTURE, PROGRAMMING and APPLICATIONS, WEST PUBLISHING COMPANY, Page: 131-197.
[5] Michael J. Pont, Embedded C, Edition 2002, Addison Wesley, Page: 57-87,217.