MULTIUSER GSM BASED ELECTRONIC NOTICE BOARD

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Electronics and Communication Engineering

under the Supervision of

Mr.Munish Sood

By

Pankaj Sharma (091052)

Harinderjeet Singh (091058)

Jeet Yadav (091060)

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled "Multiuser GSM Based Electronic Notice Board", submitted by Pankaj Sharma (091052), Harinderjeet Singh (091058), Jeet Yadav (091060) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 18-05-2013

Supervisor's Name: Mr. Munish Sood

Designation: Lecturer

ACKNOWLEDGEMENT

We are very grateful and highly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide Mr. Munish Sood, Department of Electronics and Communications Engineering, Jaypee University of Information Technology Waknaghat, in bringing this report to a successful completion.

We are grateful to Prof. Sunil Bhooshan, Head of the Department of Electronics and Communications Engineering, for permitting us to make use of the facilities available in the department to carry out the project successfully. Last but not the least we express our sincere thanks to all of our teachers and friends who have patiently extended all sorts of help for accomplishing this undertaking.

Date: 18-05-2013

Pankaj Sharma (091052)
Harinderjeet Singh (091058)
JeetYadav(091060)

Table of Content

List of Figures

List of Tables

Abstract

Wireless communication has announced its arrival on big stage and the world is going mobile. We want to control everything and without moving an inch. This remote control of appliances is possible through Embedded Systems. The use of "Embedded System in Communication" has given rise to many interesting applications that ensures comfort and safety to human life.

The main aim of this project will be to design a SMS driven automatic display board which can replace the currently used programmable electronic display. It is proposed to design receiver cum display board which can be programmed from an authorized mobile phone. The message to be displayed is sent through a SMS from an authorized transmitter. The microcontroller receives the SMS, validates the sending Mobile Identification Number (MIN) and displays the desired information. Started off as an instantaneous News display unit, we have improved upon it and tried to take advantage of the computing capabilities of microcontroller. Looking into current trend of information transfer in the campus, it is seen that important notice take time to be displayed in the notice boards. This latency is not expected in most of the cases and must be avoided. It is proposed to implement this project at the institute level. It is proposed to place display boards in major access points.

The electronics displays which are currently used are programmable displays which need to be reprogrammed each time. This makes it inefficient for immediate information transfer, and thus the display board loses its importance. The GSM based display board can be used as an add-on to these display boards and make it truly wireless. The display board programs itself with the help of the incoming SMS with proper validation. Such a system proves to be helpful for immediate information transfer.

The system required for the purpose is nothing but a Microcontroller based SMS box. The main components of the kit include microcontroller, GSM modem. These components are integrated with the display board and thus incorporate the wireless features. The GSM modem receives the SMS. Since it is a multiuser system, various users are authenticated to use the system or display the messages on the notice board.

In this project, we have authorized only three users. Their mobile number is entered through the keypad matrix and saved into the memory. Only the messages received through these three numbers will be displayed by the LCD and rest are not accepted.

A 16x2 Character LCD display is attached in byte mode to port 0 of microcontroller. This display will be used to display the messages /advertisements. Microcontroller coding will be done using Embedded C and Kiel. The modem transmits the stored message through the COM port. The microcontroller displays the message in the LCD display board. The microcontroller used in this case is AT89s52. In the prototype model, LCD display is used for simulation purpose. While implementation this can be replaced by actual display boards.

CHAPTER 1: INTRODUCTION

This project aims at integrating the expansiveness of a wireless cellular network and the ease of information transfer through the SMS with the coverage of public display boards. It is thereby a modest effort to realize the complete potential of public display boards in instantaneous information broadcast in swift response to events of interests.

As the project uniqueness is mainly influenced by the wireless cellular networks. Therefore it is essential to first provide details about the GSM Technology and then about the remaining circuitry.

1.1 Global System for Mobile Communications (GSM):

1.1.1 Introduction:

In 1982, the European Conference of Postal and Telecommunications Administrations (CEPT) created the Group Special Mobile (GSM) to develop a standard for a mobile telephone system that could be used across Europe. In 1987, a memorandum of understanding was signed by 13 countries to develop a common cellular telephone system across Europe. The first GSM network was launched in 1991 by Radiolinja in Finland with joint technical infrastructure maintenance from Ericsson.

Global System for Mobile Communications, or **GSM** (originally from **Group Special Mobile**), is the world's most popular standard for mobile telephone systems. The GSM estimates that 80% of the global mobile market uses the standard. Over 1.5 billion people use GSM across more than 212 countries and territories. This ubiquity means that subscribers can use their phones throughout the world, enabled by international roaming arrangements between mobile network operators. GSM differs from its predecessor technologies in that both signaling and speech channels are digital, and thus GSM is considered a second generation (2G) mobile phone system.

This also facilitates the wide-spread implementation of data communication applications into the system. The GSM standard has been an advantage to both consumers, who may benefit from the ability to roam and switch carriers without replacing phones, and also to network operators, who can choose equipment from many GSM equipment vendors. GSM also pioneered low-cost implementation of the short message service (SMS), also called text messaging, which has since been supported on other mobile phone standards as well. The standard includes a worldwide emergency telephone number feature.

GSM is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. The coverage area of each cell varies according to the implementation environment. Macro cells can be regarded as cells where the Base station antenna is installed on a mast or a building above average roof top level. Micro cells are cells whose antenna height is under average rooftop level; they are typically used in urban areas. Pico cells are small cells whose coverage diameter is a few dozen meters; they are mainly used indoors

Cell horizontal radius varies depending on antenna height, antenna gain and propagation conditions from a couple of hundred meters to several tens of kilometers. The longest distance the GSM specification supports in practical use is 35 kilometers (22 mi). There are also several implementations of the concept of an extended cell, where the cell radius could be double or even more, depending on the antenna system, the type of terrain and the timing advance.

Indoor coverage is also supported by GSM and may be achieved by using an indoor picocell base station, or an indoor repeater with distributed indoor antennas fed through power splitters, to deliver the radio signals from an antenna outdoors to the separate indoor distributed antenna system.

These are typically deployed when a lot of call capacity is needed indoors; for example, in shopping centers or airports. However, this is not a prerequisite, since indoor coverage is also provided by in-building penetration of the radio signals from any nearby cell.

1.1.2 Subscriber Identity Module (SIM)

One of the key features of GSM is the Subscriber Identity Module, commonly known as a SIM card. The SIM is a detachable smart card containing the user's subscription information and phone book. This allows the user to retain his or her information after switching handsets. Alternatively, the user can also change operators while retaining the handset simply by changing the SIM. Some operators will block this by allowing the phone to use only a single SIM, or only a SIM issued by them; this practice is known as SIM locking.

Sometimes mobile network operators restrict handsets that they sell for use with their own network. This is called locking and is implemented by a software feature of the phone. Because the purchase price of the mobile phone to the consumer is typically subsidized with revenue from subscriptions, operators must recoup this investment before a subscriber terminates service. A subscriber may usually contact the provider to remove the lock for a fee, utilize private services to remove the lock, or make use of free or fee-based software and web sites to unlock the handset themselves.

1.1.3 Advantages & Uses of GSM:

1. Roaming with GSM phones is a major advantage over the competing technology as roaming across CDMA networks.

2. Another major reason for the growth in GSM usage, particularly between 1998 and 2002, was the availability of prepaid calling from mobile phone operators. This allows people who are either unable or unwilling to enter into a contract with an operator to have mobile phones. Prepaid also enabled the rapid expansion of GSM in many developing countries where large sections of the population do not have access to banks or bank accounts and countries where there are no effective credit rating agencies. (In the USA, starting a non-prepaid contract with a cellular phone operator is almost always subject to credit verification through personal information provided by credit rating agencies).

3. The architecture of GSM allows for rapid flow of information by voice or data messaging (SMS). Users now have access to more information, whether personal, technical, economic or political, more quickly than was possible before the global

presence of GSM. Even remote communities are able to integrate into networks (sometimes global) thereby making information, knowledge and culture accessible, in theory, to anyone.

4. One of the most appealing aspects of wireless communications is its mobility. Much of the success of GSM is due to its mobility management, offering users the freedom and convenience to conduct business from almost anywhere at any time.

5. GSM has been the catalyst in the tremendous shift in traffic volume from fixed networks to mobile networks. This has resulted in the emergence of a mobile paradigm, whereby the mobile phone has become the first choice of personal phone.

6. Higher digital voice quality.

CHAPTER 2: DESIGN OVERVIEW

2.1 Brief design overview

The block diagram For Wireless GSM based electronic notice display is shown in figure 1.0

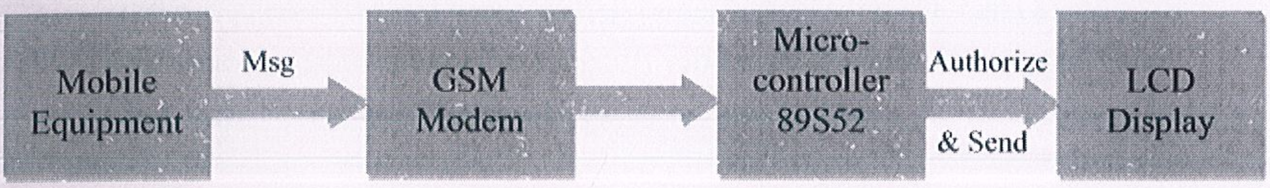


Fig 1.0: Block Diagram

The design of the project is basically divided into :

1.Transmitting Section.

2.Receiving Section.

Transmitting section consists of just mobile which has inbuilt GSM for wireless data transfer through GSM whereas receiving section consists of microcontroller 89s52 and LCD display.

Since it is a multiuser system, various users are authenticated to use the system or display the messages on the notice board. In this project, we have authorized only three users. Their mobile number is entered through the keypad matrix and saved into the memory. Only the messages received through these three numbers will be displayed by the LCD and rest are not accepted.

CHAPTER 3: GSM MODEM AND AT COMMANDS

3.1 GSM Modem

A GSM modem is a wireless modem that works with a GSM wireless network. wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate. GSM Modem is a compact and portable terminal that can satisfy various data communication needs over GSM. It can be interfaced directly with the microcontroller 89s52.

Microcontroller use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. GSM modem can be used just like a dial-up modem. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, various things can be done:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low -- only about six to ten SMS messages per minute.

Since microcontrollers use AT commands to control modems, so we will be using AT commands in programming the microcontroller so that microcontroller will be able to receive messages from the GSM modem. In this section we are going to discuss some important AT commands.

3.2 AT-Command set

The following section describes the AT-Command set. The commands can be tried out by connecting a GSM modem to one of the PC's COM ports. Type in the test-command, adding CR + LF (Carriage return + Line feed = \r\n) before executing. Table gives an overview of the implemented AT Commands in this application. The use of the commands is described in the later sections.

AT-Command set overview

Table 1.0: AT-Command set

Command	Description
AT	Check if serial interface and GSM modem is working.
ATE0	Turn echo off, less traffic on serial line.
AT+CNMI	Display of new incoming SMS.
AT+CPMS	Selection of SMS memory.
AT+CMGF	SMS string format, how they are compressed.
AT+CMGR	Read new message from a given memory location.
AT+CMGS	Send message to a given recipient.
AT+CMGD	Delete message.

3.3 Important AT Commands

After successfully testing the MODEM for its correct operational state, we need to set the MODEM parameters like Baud rate, Echo off etc to enable easier access via a microcontroller which we used in this project. Following is a list of the important AT commands

Example: Changing and saving parameters

AT+IPR=2400[Enter] Transfer rate to 2400 bps

AT&W[Enter] save parameters

Send and Receive

Example: Send SMS with GSM modem / module

Enter SMS-Center

AT+CSCA=+491722270000[Enter] Enter SMS-Center for Vodafone Germany (T-Mobile Germany = +491710760000)

Example: Send SMS

AT+CMGF=1[Enter]

AT+CMGS="+491711234567"[Enter]

>Please call office^Z

+CMGF=1 - set modem in text mode

Send SMS (^Z equals StrgZ). At D2 you can send without international and local code. If you dial with Int. and local code the transmission from foreign networks are ensured.

Example: Receive SMS

A SMS will be stored in the GSM modem / module and being sent via RS232 to the peripherals. The peripherals have to send commands to the GSM unit to receive SMS and to erase SMS from the device in order to free memory.

+CMTI:"SM",x X stands for the memory number of received SMS

AT+CMGR=X[Enter] Read SMS on memory number X

AT+CMGD=X[Enter] Erase SMS on memory number X

ATZ;E[Enter] Echo OFF

ATZ;E1[Enter] Echo ON

AT+COPS?[Enter] Shows if network of SIM-card is available. 0,2,26201= D1 availiable

AT+COPS=?[Enter] Shows all available networks

CHAPTER 4: HARDWARE DESCRIPTION

4.1 Components Used in the Project

The following major components have been used in the making of the project :

a) Microcontroller (89s52)
b) Transformer (12V)
c) Regulator (7805)
d) Diodes (IN007)
e) Capacitors (100/50, 100/25)
f) Resistance (1K)
g) LCD
h) LEDs
i) Switches
j) GSM Module

4.2 Power Supply

- Power Supply is an important part of a circuit. It provides required supply to different blocks of the circuit from input 230 V AC.
- The main blocks of power supply circuit include transformer, rectifier circuit, filter circuit, and regulator circuit.
- In this project we are using two power supplies. One provides voltage (5V) to microcontroller and lcd display whereas the other provides voltage (10V) to the GSM modem.

  The main components of Power supply are:

4.2.1Transformer:

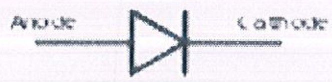
A transformer is a static electrical device that transfers energy by inductive coupling between its winding circuits. A varying current in the primary winding creates a varying magnetic flux in the transformer's core and thus a varying magnetic flux through the secondary winding. This varying magnetic flux induces a varying electromotive force (emf) or voltage in the secondary winding.

Transformers range in size from thumbnail-sized used in microphones to units weighing hundreds of tons interconnecting the power grid. A wide range of transformer designs are used in electronic and electric power applications. Transformers are essential for the transmission, distribution, and utilization of electrical energy.

- Here we have used two step down transformers. It is used to step down the ac 230V supply to 12V.
- One provides power supply to microcontroller and the other provides power supply to the GSM modem.

4.2.2 Diodes:

Here we have used In 4007 diodes are used in power supply section. In electronics, diode is a two-terminal electronic component with an asymmetric transfer characteristic, with low (ideally zero) resistance to current flow in one direction, and high (ideally infinite) resistance in the other. A semiconductor diode, the most common type today, is a crystalline piece of semiconductor material with a p–n junction connected to two electrical terminals. The most common function of a diode is to allow an electric current to pass in one direction (called the diode's forward direction), while blocking current in the opposite direction (the reverse direction). Thus, the diode can be viewed as an electronic version of a check valve. However, diodes can have more complicated behavior than this simple on–off action. Semiconductor diodes begin conducting electricity only if a certain threshold voltage or cut-in voltage is present in the forward direction (a state in which the diode is said to be forward biased). The voltage drop across a forward-biased diode varies only a little with the current, and is a function of temperature; this effect can be used as a temperature sensor or voltage reference.



Symbol of diode

4.2.3 Rectifiers:

A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), which flows in only one direction. The process is known as rectification. There are two types of rectifiers mainly half wave and full wave rectifiers.

1. Half wave rectifier: In half wave rectifier only half cycle of applied AC voltage is used. Another half cycle of AC voltage (negative cycle) is not used. Only one diode is used which conducts during positive cycle. The circuit diagram of half wave rectifier without capacitor is shown in the following figure. During positive half cycle of

the input voltage anode of the diode is positive compared with the cathode. Diode is in forward bias and current passes through the diode and positive cycle develops across the load resistance RL. During negative half cycle of input voltage, anode is negative with respected to cathode and diode is in reverse bias. No current passes through the diode hence output voltage is zero.

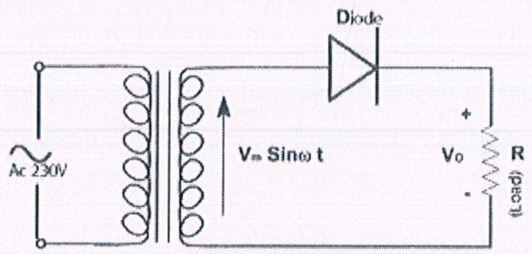


Fig2.0: Circuit diagram of Half wave rectifier

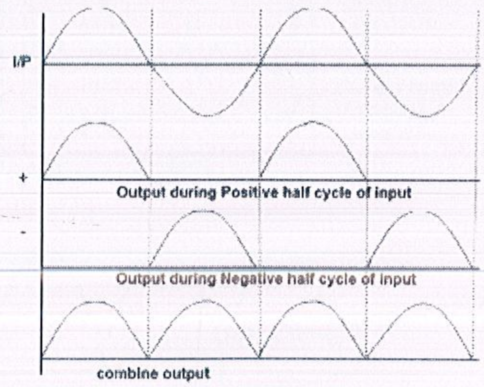


Fig 2.1: Input and Output waveforms of half wave rectifier

2.Full Wave Rectifiers : These are again of two types.

Center Tap rectifier.

Bridge rectifier.

Center Tap Rectifier: A center-tapped rectifier is a type of full-wave rectifier that uses two diodes connected to the secondary of a center-tapped transformer. The input voltage is coupled through the transformer to the center-tapped secondary. Half of the total secondary voltage appears between the center tap and each end of the secondary winding.

For a positive half-cycle of the input voltage, the polarities of the secondary voltages will include A as positive terminal and C as negative terminal while the B terminal will be positive terminal. This condition forward-biases diode D1 and reverse-biases diode D2. The current path is through D1 and the load resistor RL, as indicated. For a negative half-cycle of the input voltage, the voltage polarities on the secondary reverse completely. This condition reverse-biases D1 and forward-biases D2. The current path is through D2 and RL, as indicated. Because the output current during both the positive and negative portions of the input cycle is in the same direction through the load, the output voltage developed across the load resistor is a full-wave rectified dc voltage, as shown in the following figure.

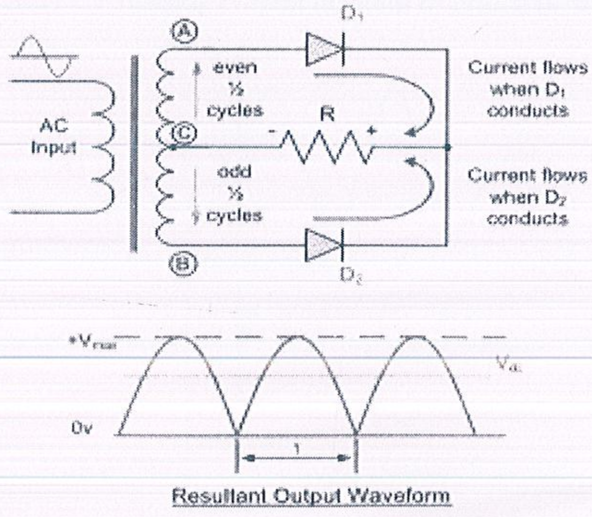


Fig3.0: Waveforms of center tapped rectifier.

Bridge rectifier:

The Bridge rectifier is a circuit, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the following figure.

The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge. For the positive half cycle of the input ac voltage, diodes B and D conduct, whereas diodes A and C remain in the OFF state. The conducting diode will be in series with the load resistance RL and hence the load current flows through R. For the negative half cycle of the input ac voltage, diodes A and C conduct whereas, B and D remain OFF. The conducting diodes A and C will be in series with the load resistance RL and hence the current flows through RL in the same direction as in the previous half cycle. Thus a bi-directional wave is converted into unidirectional wave.

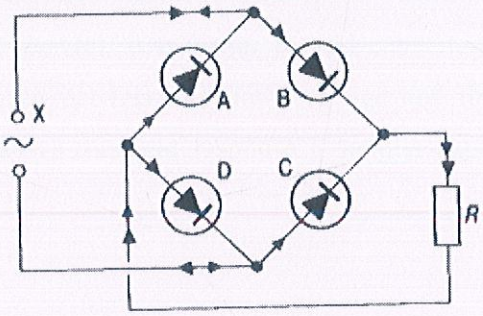


Fig4.0: Circuit diagram of bridge rectifier

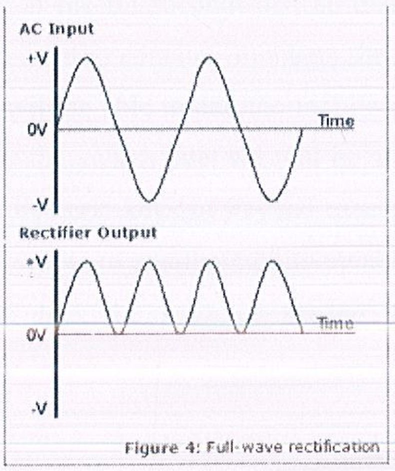


Fig 4.1: Waveform of Bridge rectifier

4.2.4 Filters and Regulators: The output of the rectifier is a pulsating dc wave. We need a constant dc output. To do this, we need to filter out the oscillations from the pulsating dc wave. This is obtained with a diode capacitor combination. A capacitor-input filter will charge and discharge such that it fills in the "gaps" between each peak. This reduces variations of voltage. This voltage variation is called ripple voltage. We are using a full wave rectifier in this project. The advantage of a full-wave rectifier over a half-wave is quite clear. The capacitor can more effectively reduce the ripple when the time between peaks is shorter. While charging, the capacitor appears as a short. This causes large currents through the diodes. To avoid damaging the devices, a surge resistor is added. Surge Resistance should be small in comparison to the load resistor (R_L). -To most effectively reduce the ripple occurring after filtering we use and IC voltage regulator.

We are using LM 7805 Regulator here. A regulator has 3 terminals: input, output and reference (or adjust). In general, it is better to add capacitors after (and before) the regulator. A large capacitor between the input voltage and the input terminal further filters the signal. To improve transient response a smaller capacitor is added after the regulator.

4.3 Switches:

We have used switches in designing the keypad matrix. As we had already discussed that we can store three numbers in the microcontroller, so there are three switches in the keypad matrix. The keypad matrix contains numbers from 0 to 9. By pressing the corresponding switches, we will be able to add the mobile numbers through the keypad matrix. For example, by pressing switch one, we will be able to add mobile number of first person by pressing the numeric keys of keypad matrix. Similarly by pressing the other two switches, we will be able to add the mobile numbers of the other two persons. In this way these switches does the work of storing the mobile numbers in the microcontroller.

4.4 Microcontroller

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

4.4.1 Description

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption (single-digit milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

A microcontroller can be considered a self-contained system with a processor, memory and peripherals and can be used as an embedded system.

<u>Interrupts</u>

Microcontrollers must provide real time (predictable, though not necessarily fast) response to events in the embedded system they are controlling. When certain events occur, an interrupt system can signal the processor to suspend processing the current instruction sequence and to begin an interrupt service routine (ISR, or "interrupt handler"). The ISR will perform any processing required based on the source of the interrupt before returning to the original instruction sequence. Possible interrupt sources are device dependent, and often include events such as an internal timer overflow, completing an analog to digital conversion, a logic level change on an input such as from a button being pressed, and data received on a communication link. Where power consumption is important as in battery operated devices, interrupts may also wake a microcontroller from a low power sleep state where the processor is halted until required to do something by a peripheral event.

Programs

Typically microcontroller programs must fit in the available on-chip program memory, since it would be costly to provide a system with external, expandable, memory. Compilers and assemblers are used to convert high-level language and assembler language codes into a compact machine code for storage in the microcontroller's memory. Depending on the device, the program memory may be permanent, read-only memory that can only be programmed at the factory, or program memory may be field-alterable flash or erasable read-only memory.

Manufacturers have often produced special versions of their microcontrollers in order to help the hardware and software development of the target system. Originally these included EPROM versions that have a "window" on the top of the device through which program memory can be erased by ultraviolet light, ready for reprogramming after a programming ("burn") and test cycle. Since 1998, EPROM versions are rare and have been replaced by EEPROM and flash, which are easier to use (can be erased electronically) and cheaper to manufacture.

Other microcontroller features

Microcontrollers usually contain from several to dozens of general purpose input/output pins (GPIO). GPIO pins are software configurable to either an input or an output state. When GPIO pins are configured to an input state, they are often used to read sensors or external signals. Configured to the output state, GPIO pins can drive external devices such as LEDs or motors.

A common feature on some microcontrollers is a digital-to-analog converter (DAC) that allows the processor to output analog signals or voltage levels.

In addition to the converters, many embedded microprocessors include a variety of timers as well. One of the most common types of timers is the Programmable Interval Timer (PIT). A PIT may either count down from some value to zero, or up to the capacity of the count register, overflowing to zero. Once it reaches zero, it sends an interrupt to the processor indicating that it has finished counting. This is useful for devices such as thermostats, which periodically test the temperature around them to see if they need to turn the air conditioner on, the heater on, etc.

A dedicated Pulse Width Modulation (PWM) block makes it possible for the CPU to control power converters, resistive loads, motors, etc., without using lots of CPU resources in tight timer loops.

Universal Asynchronous Receiver/Transmitter (UART) block makes it possible to receive and transmit data over a serial line with very little load on the CPU. Dedicated on-chip hardware also often includes capabilities to communicate with other devices (chips) in digital formats such as I^2C and Serial Peripheral Interface (SPI).

A micro-controller is a single integrated circuit, commonly with the following features:

- Central processing unit - ranging from small and simple 4-bit processors to complex 32- or 64-bit processors.
- Volatile memory (RAM) for data storage.
- ROM, EPROM, EEPROM or Flash memory for program and operating parameter storage.
- Discrete input and output bits, allowing control or detection of the logic state of an individual package pin.
- Serial input/output such as serial ports (UARTs).

- Other serial communications interfaces like I^2C, Serial Peripheral Interface and Controller Area Network for system interconnect.
- Peripherals such as timers, event counters, PWM generators.
- Clock generator - often an oscillator for a quartz timing crystal, resonator or RC circuit.
- Many include analog-to-digital converters, some include digital-to-analog converters.
- In-circuit programming and debugging support.

4.4.2 Programming Environments

Microcontrollers were originally programmed only in assembly language, but various high-level programming languages are now also in common use to target microcontrollers. These languages are either designed specially for the purpose, or versions of general purpose languages such as the C programming language. Compilers for general purpose languages will typically have some restrictions as well as enhancements to better support the unique characteristics of microcontrollers. Some microcontrollers have environments to aid developing certain types of applications. Microcontroller vendors often make tools freely available to make it easier to adopt their hardware.

Many microcontrollers are so quirky that they effectively require their own non-standard dialects of C, such as SDCC for the 8051, which prevent using standard tools (such as code libraries or static analysis tools) even for code unrelated to hardware features. Interpreters are often used to hide such low level quirks.

Interpreter firmware is also available for some microcontrollers. For example, BASIC on the early microcontrollers Intel 8052; BASIC and FORTH on the Zilog Z8 as well as some modern devices. Typically these interpreters support interactive programming.

Simulators are available for some microcontrollers. These allow a developer to analyze what the behavior of the microcontroller and their program should be if they were using the actual part. A simulator will show the internal processor state and also that of the outputs, as well as allowing input signals to be generated. While on the one hand most simulators will be limited from being unable to simulate much other hardware in a system, they can exercise conditions that may otherwise be hard to reproduce at will in the physical implementation, and can be the quickest way to debug and analyze problems.

Recent microcontrollers are often integrated with on-chip debug circuitry that when accessed by an in-circuit emulator via JTAG, allow debugging of the firmware with a debugger.

4.4.3 89S52 Microcontroller

8051 is the name of a big family of microcontrollers. The device which we used in this project is the 'AT89S52' which is a typical 8051 microcontroller manufactured by Atmel. The 89S52 has 4 different ports, each one having 8 Input/output lines providing a total of 32 I/O lines. Those ports can be used to output DATA and orders do other devices, or to read the state of a sensor, or a switch. Most of the ports of the 89S52 have 'dual function' meaning that they can be used for two different functions.

The first one is to perform input/output operations and the second one is used to implement special features of the microcontroller like counting external pulses, interrupting the execution of the program according to external events, performing serial data transfer or connecting the chip to a computer to update the software. Each port has 8 pins, and will be treated from the software point of view as an 8-bit variable called 'register', each bit being connected to a different Input/output pin.

There are two different memory types: RAM and EEPROM. Shortly, RAM is used to store variable during program execution, while the EEPROM memory is used to store the program itself, that's why it is often referred to as the 'program memory'. It is clear that the CPU (Central Processing Unit) is the heart of the micro controllers. It is the CPU that will Read the program from the FLASH memory and execute it by interacting with the different peripherals.

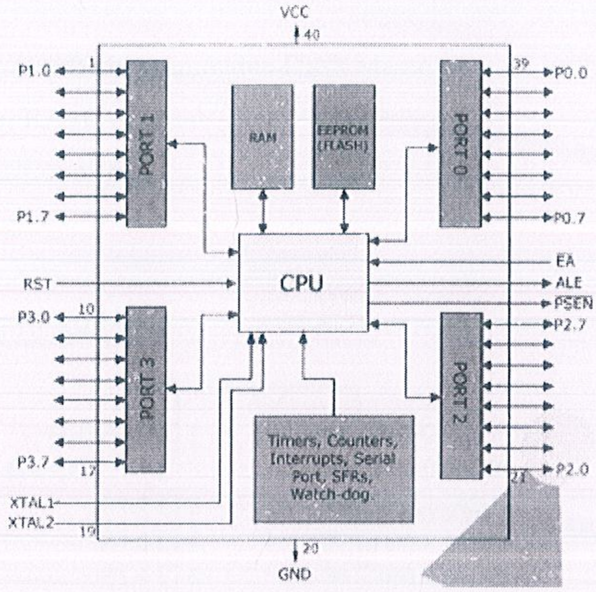


Fig 5.0 : Simple architecture of 89S52

Additional Features of 89S52

The AT89S52 is a low-power, high-performance CMOS 8-bit microcomputer with 8Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80s51 and 80s52 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89s52 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications. One of the main features of this microcontroller is that it has 1,000 Write/Erase Cycles. It has 256 x 8-bit Internal RAM also.

89S51

- 4K Bytes of In-System Reprogrammable Flash Memory
- 128 x 8-bit Internal RAM
- Two 16-bit Timer/Counters
- Six Interrupt Sources

89S52

• 8K Bytes of In-System Reprogrammable Flash Memory

• 256 x 8-bit Internal RAM

• Three 16-bit Timer/Counters

• Eight Interrupt Source

Pin configuration of 89S52

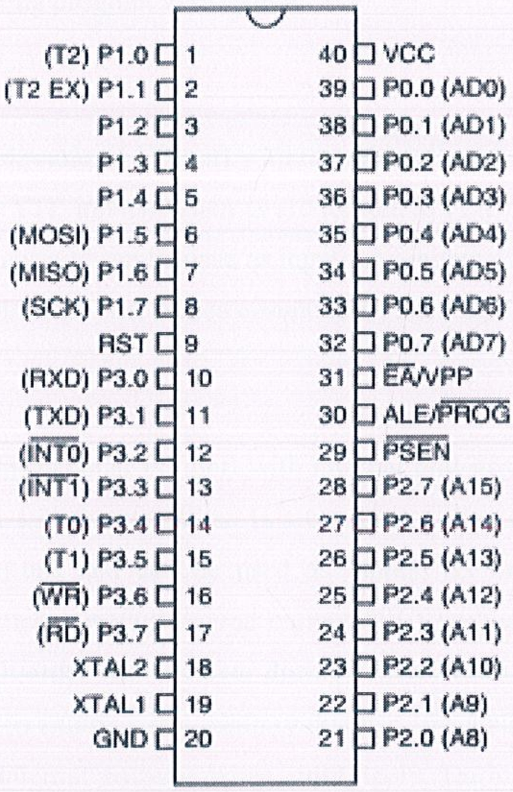


Fig 6.0: Pin diagram of 89S52

Pin Functions:

- VCC: Supply voltage (5V)
- GND: Ground.
- Port 0:

- Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

  Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1:

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups.

Port 2:

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses. In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that uses 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3:

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

RST:

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG:

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode

PSEN:

Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP:

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset.EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

XTAL1:

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting oscillator amplifier.

Circuit Diagram

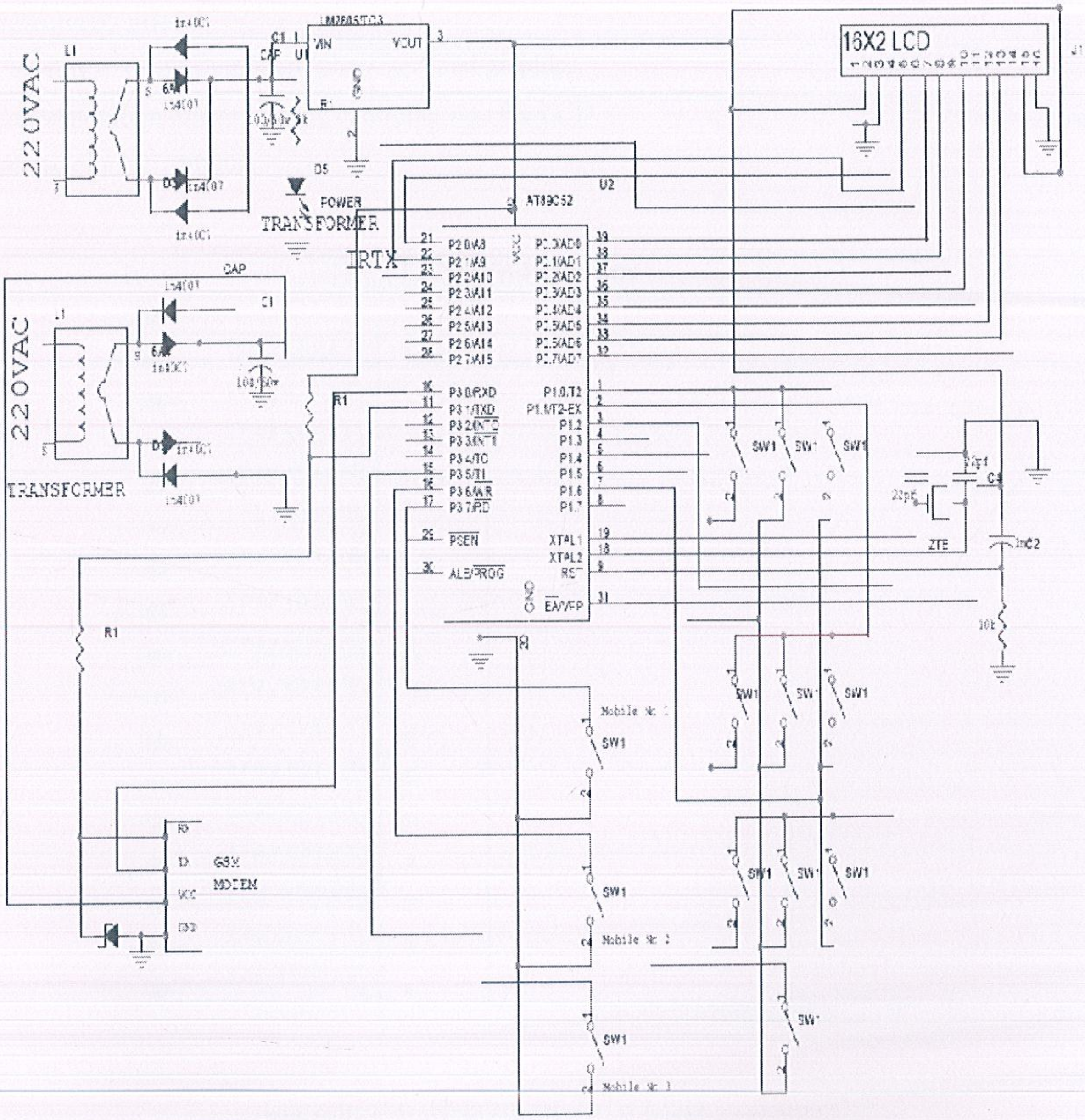


Fig 7.0: Circuit diagram of the hardware

4.5 LCD Display

In this project we are using 16*2 LCD to display the messages. Here 16*2 LCD means that it can accommodate 16 characters per line and there will be two lines. In recent years the LCD is finding widespread use replacing LED's. This is due to the following reasons

1. Declining prices
2. Ability to display numbers, characters and graphics.
3. Incorporation of a refreshing controller into the LCD.
4. Ease of programming.

16*2 LCD Pin Diagram

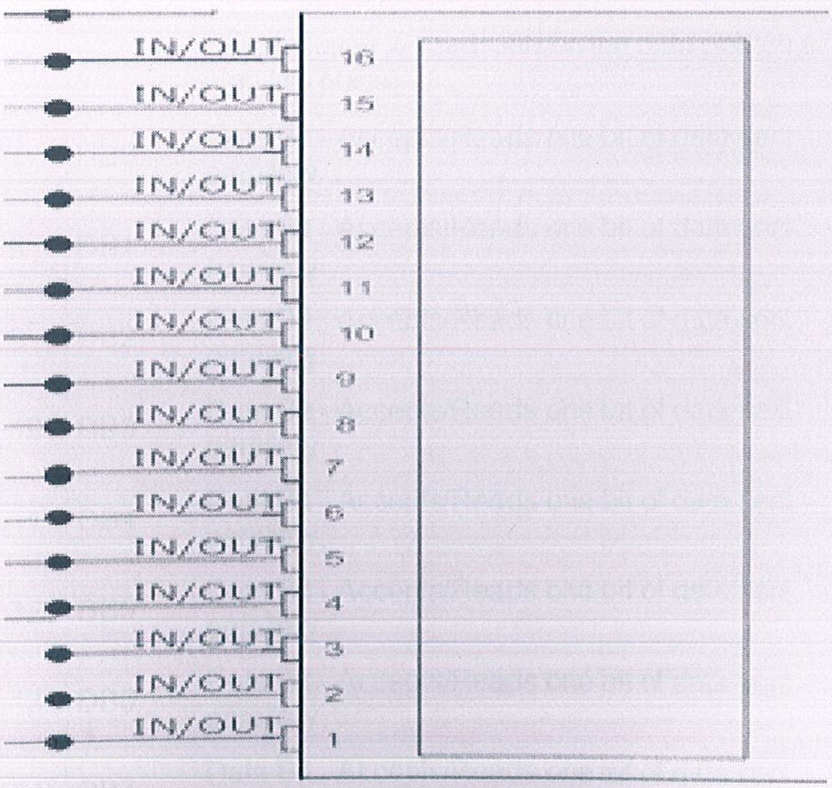


Fig 8.0: Pin diagram of 16*2 LCD

Table 2: Pin Description of 16*2 LCD

Pin	Symbol	Description
1	GND	Ground
2	Vcc	Connected to +5V Power supply
3	Vee	Sets Contrast when voltage on this pin is varied (usually by a potentiometer)
4	RS	Register Select (rs=0 selects command register, rs=1 selects data register)
5	R/W	Used to Read or Write information on the LCD (0 while writing, 1 while reading)
6	EN	Enable (used to read data on the data port on a high-low pulse)
7	DB0	Data Bit - Accepts/Reads one bit of data sent parallelly
8	DB1	Data Bit - Accepts/Reads one bit of data sent parallelly
9	DB2	Data Bit - Accepts/Reads one bit of data sent parallelly
10	DB3	Data Bit - Accepts/Reads one bit of data sent parallelly
11	DB4	Data Bit - Accepts/Reads one bit of data sent parallelly
12	DB5	Data Bit - Accepts/Reads one bit of data sent parallelly
13	DB6	Data Bit - Accepts/Reads one bit of data sent parallelly
14	DB7	Data Bit - Accepts/Reads one bit of data sent parallelly

The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus). Here we are using 8bit mode as it is simple for programming.

CONTROL LINES:

1. EN (PIN6): This is called "enable ". This is used to tell lcd that you are sending data to lcd. A low (0) to high (1) pulse should be give to en pin to latch data.

2. RS (PIN4): This is called "Register Select". It is used to select data or command register. When RS=0 command register is selected and when RS=1 data register is selected.

3. RW (PIN5): This is called as "read/write" pin. When RW=0 data is written on lcd. And when RW=1 data can be efficiently read from lcd. Data reading is used to check busy status of lcd.

DATALINES (D0-D7):

The data bus consists of 8 or 4 lines (depending upon operation selected.). In case of 8bit mode all D0-D7 pins are used. D7 pin is busy flag when D7=1 lcd is busy and when D7=0 lcd is ready.

4.5.1 Algorithm for Programming LCD

To initialize lcd:

- Specify Function 8bit or 4bit(38h for 8bit mode 5×7 dot character format)
- Display on-off control (0eh for display cursor blinking).
- Entry mode set (06h for automatic right shifting).
- Clear display (01h for clear display).

- Select line 1(80h for selecting line 1).

To write lcd:

- Check busy flag of lcd.
- Make RW low.
- Make RS=0 if data is command and RS=1 if data is data to be display.
- Give Enable pulse (H to L) on EN pin to latch data.

INTERFACING LCD WITH 89S52

1. Make Hardware Connections.
2. Control Lines:
    - RS =P2.0 (sbit rs =P2^0)
    - RW =P2.2 (sbit rw=P2^2)
    - EN =P2.1 (sbit en =P2^1)
3. Datelines: DB0-DB7 D7 are connected to P0.0 to P0.7 respectively.
4. LCD Initialization.
5. Sending Data to the LCD.
6. Display Data on the LCD.
7. Preset is used to change contrast of lcd.
8. Pin 15&16 are used for backlight of lcd.

CHAPTER 5: SOFTWARE DESCRIPTION

5.1 Embedded C:

Embedded C is language for programming the microcontroller for embedded applications. There is a large and growing international demand for programmers with embedded skills and many desktop developers are starting to move in this developed area.

The reasons for writing programs in C are

- It is easier and less time consuming to write in C than assembly.
- C is easier to modify and update.
- We can use code available in function libraries.
- C is portable to other microcontroller also.

5.2 Software Used

5.2.1 Keil u-Vision 3.0

Keil Software is used to provide you with software development tools for 8051 based microcontrollers. With the Keil tools, you can generate embedded applications for virtually every 8051 derivative. The supported microcontrollers are listed in the μ-vision.

Keil is basically a compiler used to compile embedded C programs. The Keil 8051 Development Tools are designed to solve complex problems facing embedded software developers.

It is a compiler for debugging a embedded C program. It creates a hex file from the compiled C program which can be burned into the IC.

5.2.2 LAB PRO-51

The LAB PRO-51 is a single board computer based upon the popular 8051 family of microcontrollers. The particular device installed on the board is the P89S51RB2BA. It is a fully static CPU with optimized internal timing of 6 clocks per machine cycle. The processor can run at up to 20 MHz. Optionally, the processor can be configured for 12 clocks per machine cycle. In this mode, the processor runs at up to 33 MHz. Like other microcontrollers in its family, the 89S51 uses Flash for on-chip program storage. A major advantage of the 89S51, however, is its mechanism for in-system programming. Its Flash can be programmed by downloading user code over the on-chip serial port. This approach is referred to as "in-system" since the 89S52remains installed on the LAB PRO-51 during programming.

Typically, a host PC is used to perform the in-system programming. The PC is linked to the LAB PRO-51 by connecting the DB-09 connector on the board to a serial COM port on the PC. Programming software on the PC provides all necessary programming operations. Hence, no special programmer is required.

A major design feature of the LAB PRO-51 is its prototyping area. The prototype area contains a large number of solder pads (.060 pad with .036 hole) placed on a .1 inch grid. User circuitry can be built and tested in this area. Board connectors are used to wire 89S52 lines to circuitry in the prototype space. All in all, the LAB PRO-51 is great for building and testing designs which are based on the P89S52 processor.

5.3 Theory of Operation

In this project we interfaced 8952 microcontroller with the GSM Modem. The 89S52 microcontroller is used to decode the received message and do the required action. The protocol used for the communication between the two is AT command. The microcontroller pulls the SMS received by phone, decodes it, recognizes the Mobile no. and then switches on the relays attached to its port to control the appliances.

5.4 Flow Diagram

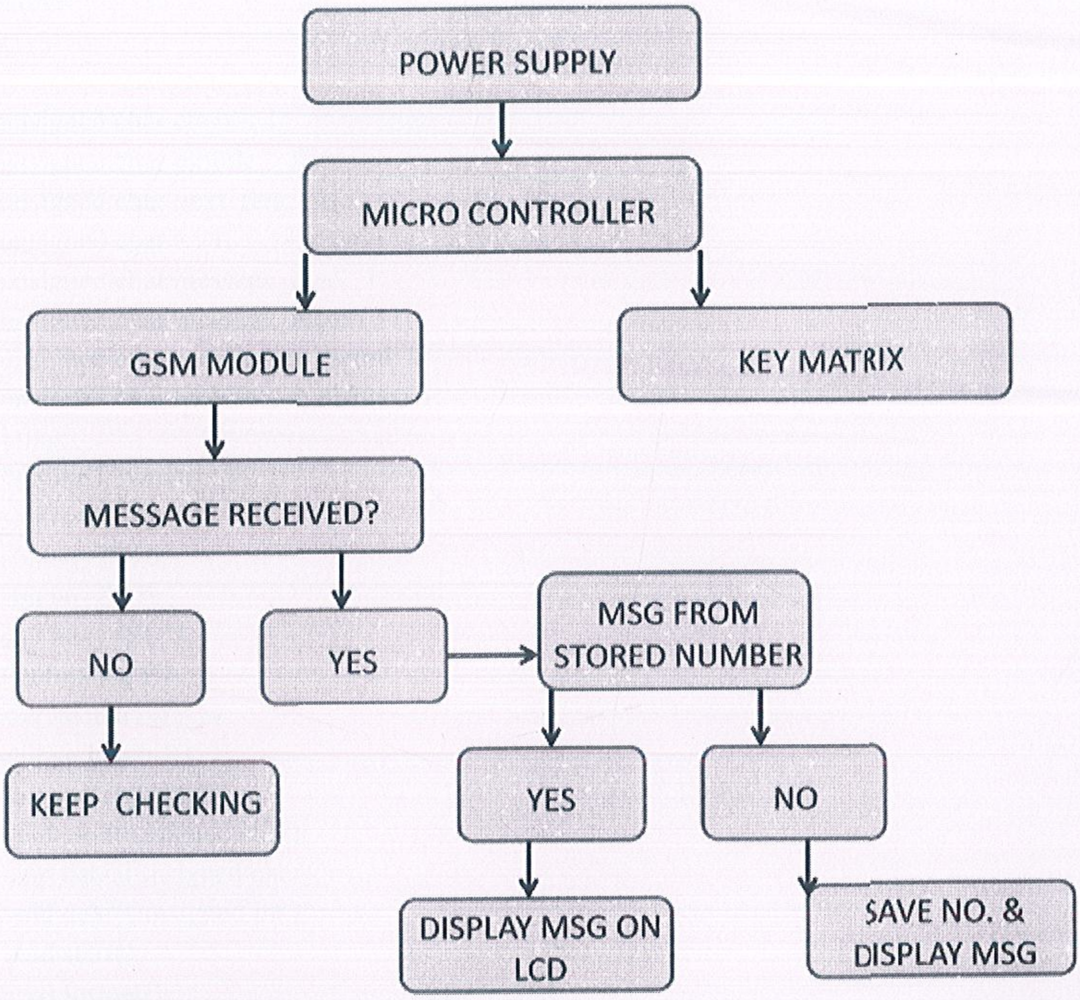


Fig 9.0: Flow Diagram

5.5 Programming of Microcontroller

```
#include<reg52.h>
struct
{
unsigned char cmti  : 1;
unsigned char cmgr  : 1;
unsigned char mobile  : 1;
unsigned char next_line  : 1;
unsigned char a : 1;
unsigned char message_read : 1;
unsigned char message_record : 1;
unsigned char mobile_no_record: 1;
unsigned char mobile_no_display: 1;
}flag;
#define keymatrix P2
sbit rs=P3^7;
sbit rw=P3^6;
sbit en=P3^5;
sbit busy=P1^7;
#define lcd P1
sbit switch1=P3^4;
sbit switch2=P3^3;
sbit switch3=P3^2;
void cmd(unsigned char);
void dataa(unsigned char);
void delay(unsigned int );
void start();
void format();
void messege_delete(unsigned char loc);
void display_string(const char *bb,unsigned char lenth);
unsigned char compare_string(char *,char *,unsigned char lenth);
void delete_string( char *bb,unsigned char lenth);
void _read_();
unsigned char key_matrix(void);
void _at(void);
void _sbuf(unsigned char);
void _formet(void);
code unsigned char compare[4][6]= {
                              {"+CMTI:"},
```

```
                {"+91"},
                {10},
                {"+CMGR"}
                };
code unsigned char message_clear[16]={"                "};
char specific_mobile_1[10]={"9814332470"};
char specific_mobile_2[10]={"9988087637"};
char specific_mobile_3[10]={"5252524444"};
unsigned char
mobile_count=0,abz,diamantion,a,ABCD,aabb,message_location,store_0[6],mobile_no_r
ec[10],mobile_message[16],zero,one,two,three,asa;
 void serial(void) interrupt 4
{
if(TI==1)
{
ABCD=0;
TI=0;
}
if(RI==1)
{
RI=0;
///{{{{{find meaasge location
if(flag.a)
{
flag.a=0;
message_location=SBUF;
flag.message_read=1;
}
if((flag.cmti)&&(SBUF==','))
{
flag.cmti=0;
flag.a=1;
asa=0;
}
//}}}}}find meaasge location
///{{{{{find mobile no
if((flag.mobile)&&(aabb<10)&&(flag.cmgr==1))
{
mobile_no_rec[aabb++]=SBUF;
if(aabb==10)
{
flag.cmgr=0;
flag.mobile_no_record=1;
flag.mobile_no_display=1;
flag.next_line=0;
        abz=0;
```

```
}
}
//}}}}find find mobile no
///{{{{{find message
if((flag.mobile_no_record)&&(flag.next_line))
{
if((SBUF!=10)&&(abz<16))
{
mobile_message[abz++]=SBUF;
}
else
{
flag.message_record=1;
flag.next_line=0;
}
}

//}}}}find find message
if(((SBUF=='+')||(SBUF==10)||(SBUF==','))||(a>19)||(SBUF==""))
{
store_0[0]=SBUF;
a=0;
if(zero!=6)
{
zero=0;
}
if(one!=3)
{
one=0;
}
if(two!=1)
{
two=0;
}
if(three!=5)
{
three=0;
}
}

 if((store_0[0]=='+')||(SBUF==10))
          {
           store_0[a]=SBUF;
          if(store_0[a]==compare[0][a])//+cmti:
         {
                    if(zero==a)
```

```
                {
                zero++;///6
                }
        }
                if(store_0[a]==compare[1][a])//+91
        {
        if(one==a)
                {
                one++;//3
                }
        }
                if(store_0[a]==compare[2][a])//10
        {
                if(two==a)
                {
                two++;//1
                }
        }
                if(store_0[a]==compare[3][a])//+cmgr
        {
                if(three==a)
                {
                three++;///5
                }
        }
            a++;
            }
if(zero==6)//for +cmti
{
zero=0;
flag.cmti=1;
}
if(one==3)//for +91
{
one=0;
flag.mobile=1;
aabb=0;
}
if(two==1)// for 10
{
two=0;
flag.next_line=1;
}
if(three==5)// for +cmgr
{
three=0;
```

```
flag.cmgr=1;
}
}
}
void main()
{
/*
char     arrqay[10]={"a123456709"},llll=0;
char     arrsay[10]={"a123456709"};
llll=compare_string(&arrqay[0],&arrsay[0],10);
         *////////////////
 flag.cmti=0;
 flag.cmgr=0;
 flag.mobile=0;
 flag.next_line=0;
 flag.a=0;
 flag.message_read=0;
 flag.message_record=0;
 flag.mobile_no_record=0;
 flag.mobile_no_display=0;
 abz=0;
 keymatrix=0xf0;
P3=0XFF;
rs=0;
rw=1;
do
{
en=0;
delay(1);
en=1;
}
while(busy==1);
cmd(0x38);
cmd(0x0c);
cmd(0x01);
cmd(0x80);
dataa('W');dataa('a');dataa('i');dataa('t');
delay(15100);
cmd(0x80);
dataa('R');dataa('e');dataa('a');dataa('d');dataa('y');
 TI=0;
 RI=0;
TMOD=0X20;
SCON=0X50;
IE=0X90;
TH1=0XFD;
```

```
TR1=1;
_at();
_at();
_formet();
messege_delete('1');
messege_delete('2');
while(1)
{
if(flag.message_read==1)
{
flag.message_read=0;
_read_();
}
if(flag.mobile_no_display==1)
{
flag.mobile_no_display=0;
cmd(0x80);
display_string(&mobile_no_rec[0],10);
}
if(flag.message_record==1)
{
flag.mobile_no_record=0;
flag.message_record=0;
cmd(0xc0);
display_string(&message_clear[0],16);

        /////////////
//      specific_mobile_1
if(compare_string(&specific_mobile_1,&mobile_no_rec[0],10))
        {
cmd(0xc0);
display_string(&mobile_message[0],abz-1);
        }
else if(compare_string(&specific_mobile_2,&mobile_no_rec[0],10))
        {
cmd(0xc0);
display_string(&mobile_message[0],abz-1);
        }
else if(compare_string(&specific_mobile_3,&mobile_no_rec[0],10))
        {
cmd(0xc0);
display_string(&mobile_message[0],abz-1);
        }
delete_string(&mobile_no_rec[0],10);
delete_string(&mobile_message[0],16);
messege_delete(message_location);
```

```
}
//////{{{{{{{{{message 1 replace
if(switch1==0)//p3.4
{
keymatrix=0xf0;
mobile_count=0;
cmd(0xc0);
display_string(&message_clear[0],16);
cmd(0x80);
display_string(&message_clear[0],16);
cmd(0x80);
dataa('E');dataa('n');dataa('t');dataa('e');dataa('r');
cmd(0xc0);
while(mobile_count<10)///p2
{
 if(keymatrix!=0xf0)
        {
specific_mobile_1[mobile_count]=key_matrix();
dataa(specific_mobile_1[mobile_count]);
        mobile_count++;
        while(keymatrix!=0xf0);
        delay(500);
        }
}
cmd(0xc0);
display_string(&message_clear[0],16);
cmd(0xc0);
dataa('S');dataa('a');dataa('v');dataa('e');dataa('d');
}
//////}}}}}}}}}message 1 replace
//////{{{{{{{{{message 2 replace
if(switch2==0)//p3.4
{
keymatrix=0xf0;
mobile_count=0;
cmd(0xc0);
display_string(&message_clear[0],16);
cmd(0x80);
display_string(&message_clear[0],16);
cmd(0x80);
dataa('E');dataa('n');dataa('t');dataa('e');dataa('r');
cmd(0xc0);
while(mobile_count<10)///p2
{
        if(keymatrix!=0xf0)
        {
```

```
specific_mobile_2[mobile_count]=key_matrix();
dataa(specific_mobile_2[mobile_count]);
                    mobile_count++;
          while(keymatrix!=0xf0);
          delay(500);
          }

}
cmd(0xc0);
display_string(&message_clear[0],16);
cmd(0xc0);
dataa('S');dataa('a');dataa('v');dataa('e');dataa('d');
}
//////}}}}}}}}}mob 2 replace
//////{{{{{{{{{mob 3 replace
if(switch3==0)//p3.4
{
keymatrix=0xf0;
mobile_count=0;
cmd(0xc0);
display_string(&message_clear[0],16);
cmd(0x80);
display_string(&message_clear[0],16);
cmd(0x80);
dataa('E');dataa('n');dataa('t');dataa('e');dataa('r');
cmd(0xc0);
while(mobile_count<10)///p2
{
          if(keymatrix!=0xf0)
          {
specific_mobile_3[mobile_count]=key_matrix();
dataa(specific_mobile_3[mobile_count]);
                    mobile_count++;
          while(keymatrix!=0xf0);
          delay(500);
          }
}
cmd(0xc0);
display_string(&message_clear[0],16);
cmd(0xc0);
dataa('S');dataa('a');dataa('v');dataa('e');dataa('d');
}
//////}}}}}}}}}mob 3 replace
}
}
void _read_()
```

```
{
  _sbuf('A');
  _sbuf('T');
            _sbuf('+');
            _sbuf('C');
            _sbuf('M');
            _sbuf('G');
            _sbuf('R');
            _sbuf('=');
            _sbuf(message_location);
            _sbuf(13);
            _sbuf(10);
            delay(50);
}
void _formet(void)
{
  _sbuf('A');
  _sbuf('T');
            _sbuf('+');
            _sbuf('C');
            _sbuf('M');
            _sbuf('G');
            _sbuf('F');
            _sbuf('=');
            _sbuf('1');
            _sbuf(13);
            _sbuf(10);
            delay(50);
}
void cmd(unsigned char value)
{
rs=0;
rw=0;
lcd=value;
en=1;
delay(1);
en=0;

}
void dataa(unsigned char value)
{
rs=1;
rw=0;
lcd=value;
en=1;
delay(1);
```

```
en=0;
}
void delay(unsigned int val)
{
unsigned int n,k;
for(n=0;n<val;n++)
{
for(k=0;k<150;k++);
}
}
void _sbuf(unsigned char Sbuf)
{
 SBUF= Sbuf;
 ABCD=1;
 while(ABCD==1);
}


void _at(void)
{
_sbuf('A');
_sbuf('T');
_sbuf(13);
_sbuf(10);
delay(50);
}
void display_string(const char *bb,unsigned char lenth)
{
unsigned char kk;
for(kk=0;kk<lenth;kk++)
        {
                dataa(bb[kk]);
        }
}
void delete_string(char *bb,unsigned char lenth)
{
unsigned char kk;
for(kk=0;kk<lenth;kk++)
        {
                bb[kk]=0;
        }
}

void messege_delete(unsigned char loc)
{
_sbuf('A');
```

```
_sbuf('T');
_sbuf('+');
_sbuf('C');
_sbuf('M');
_sbuf('G');
_sbuf('D');
_sbuf('=');
_sbuf(loc);
_sbuf(13);
_sbuf(10);
delay(50);
}
unsigned char compare_string(char *pc,char *gc,unsigned char lenth)
{
        unsigned char kk;
for(kk=0;((pc[kk]==gc[kk])&&(kk<lenth));kk++)
        {
                //bb[kk]=0;
        }
        if(kk==lenth)
        {
        return(1);
        }
        else
        {
        return(0);
        }
}
unsigned char key_matrix(void)
{
        unsigned char a='0';
keymatrix=0xfe;
if(keymatrix==0xee)
{
a='1';
}
if(keymatrix==0xde)
{
a='2';
}
if(keymatrix==0xbe)
{
a='3';
}
keymatrix=0xfd;
if(keymatrix==0xed)
```

```
{
a='4';
}
if(keymatrix==0xdd)
{
a='5';
}
if(keymatrix==0xbd)
{
a='6';
}
keymatrix=0xfb;
if(keymatrix==0xeb)
{
a='7';
}
if(keymatrix==0xdb)
{
a='8';
}
if(keymatrix==0xbb)
{
a='9';
}
keymatrix=0xf7;
if(keymatrix==0xd7)
{
a='0';
}
keymatrix=0xf0;
return(a);
}
```

CHAPTER 6: APPLICATIONS OF THE PROJECT

This project solves our following purposes.

- ✓ Sending message from any of the remote area to the distant located e-notice board using GSM mobile.
- ✓ To design a SMS driven automatic display board which can replace the currently used programmable electronic display

Apart from this, there are various applications of this project. Some of the most prominent applications are mentioned below:

- ✓ Can prove to be beneficial in sending message from any of the remote area to the distant located e-notice board
- ✓ Can prove to be very beneficial for developing a centralized system
- ✓ Educational Institutions and Organizations.
- ✓ Managing Traffic.
- ✓ Advertisement.
- ✓ Railway Station.

CONCLUSION

The use of microcontroller in place of a general purpose computer allows us to theorize on many further improvements on this project prototype. Temperature display during periods wherein no message buffers are empty is one such theoretical improvement that is very possible. The ideal state of the microcontroller is when the indices or storage space in the SIM memory are empty and no new message is there to display. With proper use of interrupt routines the incoming message acts as an interrupt, the temperature display is halted and the control flow jumps over to the specific interrupt service routine which first validates the sender's number and then displays the information field. Another very interesting and significant improvement would be to accommodate multiple receiver MODEMS at the different positions in a geographical area carrying duplicate SIM cards. With the help of principles of TDMA technique, we can choose to simulcast and /or broadcast important notifications. After a display board receives the valid message through the MODEM and displays it, it withdraws its identification from the network &synchronously another nearby MODEM signs itself into the network and starts to receive the message. The message is broadcast by the mobile switching center for a continuous time period during which as many possible display board MODEMS "catch" the message and display it as per the constraint of validation. Multilingual display can be another added variation of the project. The display boards are one of the single most important media for information transfer to the maximum number of end users. This feature can be added by programming the microcontroller to use different encoding decoding schemes in different areas as per the local language. This will ensure the increase in the number of informed users. Graphical display can also be considered as a long term but achievable and target able output. MMS technology along with relatively high end microcontrollers to carry on the tasks of graphics encoding and decoding along with a more expansive bank of usable memory can make this task a walk in the park.

REFERENCES

1. Muhammad Ali Mazidi, "The 8051 Microcontroller and Embedded Systems", 2008

2. ATMEL, Microcontroller data book, 1997.

3. James W. Stewart, The 8051 Microcontroller, Hardware, Software and Interfacing, Prentice Hall, 1993

4. www.8051projects.net/

5. www.alldatasheet.com/