A MAJOR PROJECT ON

CELL PHONE BASED ELECTRONIC VOTING MACHINE

SUBMITTED IN THE PARTIAL FULFILLMENT OF

REQUIRMENT

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

BY

PHALGUNI SINGH (1120434042)

RENUKA MANI RAJPUT (1120434050)

SNEHAL GUPTA (1120434075)

SHUBHI SRIVASTAVA(1120434071)

UNDER THE SUPERVISION OF

MR. PARITOSH KUMAR AWADH

SCHOOL OF ENGINEERING

BABU BANARASI DAS UNIVERSITY

LUCKNOW

MAY, 2016

CERTIFICATE

It is certified that the work contained in this Project entitled **"CELL PHONE BASED ELECTRONIC VOTING MACHINE"** by **PHALGUNI SINGH (1120434042) , RENUKA MANI RAJPUT (1120434050) , SNEHAL GUPTA (1120434075)** and **SHUBHI SRIVASTAVA (1120434071)** for the award of **Bachelor of Technology** from **Babu Banarasi Das University** has been carried out under my supervision.

Signature
Mrs. Poonam Pathak,
Associate Professor & H.O.D.,
Department of Electronics &
Communication Engineering,
Lucknow (U.P) .

Signature
Mr. Paritosh Kumar Awadh,
Associate Professor,
Department of Electronics &
Communication Engineering,
Lucknow (U.P).

Date: 02/05/2016

ACKNOWLEDGEMENT

Whenever a module of work is completed successfully, a source of inspiration and guidance is always there for the student. I, hereby take the opportunity to thank those entire people who helped me in many different ways.

First and foremost, I am grateful to my thesis guide **Mr. Paritosh Kumar Awadh,** B.B.D. University, for showing faith in my capability and providing able guidance and his generosity and advice extended to me throughout my thesis.

Last, but not least I would like to thank all my faculty and my friends for helping me in all measure of life and for their kind cooperation and moral support.

PHALGUNI SINGH
RENUKA MANI RAJPUT
SNEHAL GUPTA
SHUBHI SRIVASTAVA

ABSTRACT

India is world's largest democracy. Fundamental right to vote or simply voting in elections forms the basis of Indian democracy.

In India all earlier elections a voter used to cast his vote by using ballot paper. This is a long, time-consuming process and very much prone to errors.

This situation continued till election scene was completely changed by electronic voting machine. No more ballot paper, ballot boxes, stamping, etc. all this condensed into a simple box called ballot unit of the electronic voting machine.

Cell phone based voting machine is capable of saving considerable printing stationery and transport of large volumes of electoral material. It is easy to transport, store and maintain. It completely rules out the chance of invalid votes. Its use results in reduction of polling time, resulting in fewer problems in electoral preparations, law and order, candidates' expenditure, etc. and easy and accurate counting without any mischief at the counting center.

Our cell phone based voting machine consists of microcontroller ATMEL ATMEGA16, GSM module. The voter can cast his vote by sending SMS in appropriate format to the EVM using his registered SIM number. Correspondingly a confirmation message will be sent to the voter informing the status of his vote.

This project is based on assembly language programming. The software platforms used in this project are BASCOMAVR using C language and PCB layout using DIPTRACE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1.1 INTRODUCTION

Electronic Voting Machine (EVM) retains all the characteristics of voting by ballot papers, while making polling a lot more expedient. Being fast and absolutely reliable, the EVM saves considerable time, money and manpower. And, of course, helps maintain total voting secrecy without the use of ballot papers. The EVM is 100 per cent tamper proof. And, at the end of the polling, just press a button and there you have the results.

Electronic voting machine has now days become an effective tool for voting. It ensures flawless voting and thus has become more widespread. It ensures people about their vote being secured. It avoids any kind of malpractice and invalid votes. Also such kind of system becomes more economical as consequent expenditure incurred on manpower is saved. It is also convenient on the part of voter, as he has to just press one key whichever belongs to his candidates.

The aim of our project is to design & develop a mobile based voting machine. In this project user can send SMS through his registered mobile number to cast his vote. The user is desired to send the SMS in a predefined format to cast his vote. The SMS will be received by the GSM module of the EVM. If he has entered a valid choice, his vote will be caste. For invalid password/choice appropriate message will be sent to the users mobile phone. A reset button is provided for resetting the system. A master pattern from the authorized SIM number is used to get the results instantaneously. The results can be deleted by a pattern from the authorized number.

1.2 COMPONENTS

LIST OF COMPONENTS USED

Sr. no	Equipment	Quantity
1	IC ATMEGA16 MC	1
2	GSM SIM 300 Module	1
3	Voltage Regulator 7805	1
4	16*2 LCD display(LM016L)	1
5	Crystal Oscillator(8 MHz)	1
6	Berg Strip	4
7	Push Button Switch	3
8	Adapter (12V,1.5 A)	1
9	LED	2
10	Resistors(10KΩ,47kΩ,330kΩ)	5
11	Capacitors(22pf,10µf)	4
12	A103J Resistance Network	1
13	DC Pin	1

Table No. 1.1: List of components

1.3 COMPONENT DESCRIPTION

1.3.1) MICRO-CONTROLLER

Microcontroller can be termed as a single on chip computer which includes number of peripherals like RAM, EEPROM, Timers etc., required to perform some predefined task.

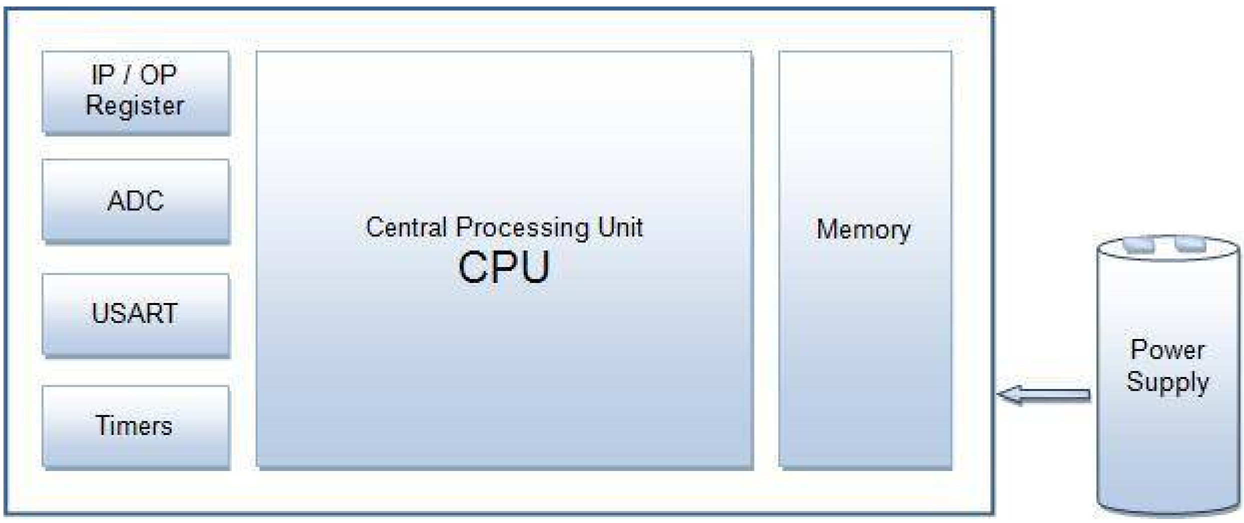


Fig:1.1:Microcontroller

The computer on one hand is designed to perform all the general purpose tasks on a single machine like you can use a computer to run a software to perform calculations or you can use a computer to store some multimedia file or to access internet through the browser, whereas the microcontrollers are meant to perform only the specific tasks, for e.g., switching the AC off automatically when room temperature drops to a certain defined limit and again turning it ON when temperature rises above the defined limit.

There are number of popular families of microcontrollers which are used in different applications as per their capability and feasibility to perform the desired task, most common of these are 8051, **AVR** and PIC microcontrollers.

1.3.1.1 AVR

The **AVR** is a modified Harvard architecture 8-bit RISC single-chip microcontroller, which was developed by Atmel in 1996. The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM,EPROM, or EEPROM used by other microcontrollers at the time. The AVR 8-bit microcontroller architecture was introduced in 1997.

1.3.1.2 History of AVR

AVR was developed in the year 1996 by Atmel Corporation. The architecture of **AVR** was developed by Alf-Egil Bogen and Vegard Wollan. AVR derives its name from its developers and stands for **Alf-EgilBogen VegardWollan RISC microcontroller**, also known as **A**dvanced **V**irtual **R**ISC. The AT90S8515 was the first microcontroller which was based on **AVR architecture** however the first microcontroller to hit the commercial market was AT90S1200 in the year 1997.

AVR microcontrollers are available in three categories:

1. **Tiny AVR –** Less memory, small size, suitable only for simpler applications
2. **Mega AVR –** These are the most popular ones having good amount of memory (upto 256 KB), higher number of inbuilt peripherals and suitable for moderate to complex applications.
3. **Xmega AVR –** Used commercially for complex applications, which require large program memory and high speed.

The following table compares the above mentioned AVR series of microcontrollers:

Series Name	Pins	Flash Memory	Special Feature
TinyAVR	6-32	0.5-8 KB	Small in size
MegaAVR	28-100	4-256KB	Extended peripherals
XmegaAVR	44-100	16-384KB	DMA , Event System included

Table: 1.2 : AVR Series

1.3.1.3 Special about AVR-

They are fast: **AVR microcontroller** executes most of the instructions in single execution cycle. AVRs are about 4 times faster than PICs, they consume less power and can be operated in different power saving modes. Let's do the comparison between the three most commonly used families of microcontrollers.

	8051	**PIC**	**AVR**
SPEED	Slow	Moderate	Fast
MEMORY	Small	Large	Large
ARCHITECTURE	CISC	RISC	RISC
ADC	Not Present	Inbuilt	Inbuilt
Timers	Inbuilt	Inbuilt	Inbuilt
PWM Channels	Not Present	Inbuilt	Inbuilt

Table: 1.3:Comparison of AVR

AVR is an 8-bit microcontroller belonging to the family of Reduced Instruction Set Computer (**RISC**). In RISC architecture the instruction set of the computer are not only fewer in number but also simpler and faster in operation. The other type of categorization is CISC (Complex Instruction Set Computers).

8-bit microcontroller is capable of transmitting and receiving 8-bit data. The input/output registers available are of 8-bits. The AVR family controllers have register based architecture which means that both the operands for an operation are stored in a register and the result of the operation is also stored in a register. Following figure shows a simple example performing OR operation between two input registers and storing the value in Output Register.

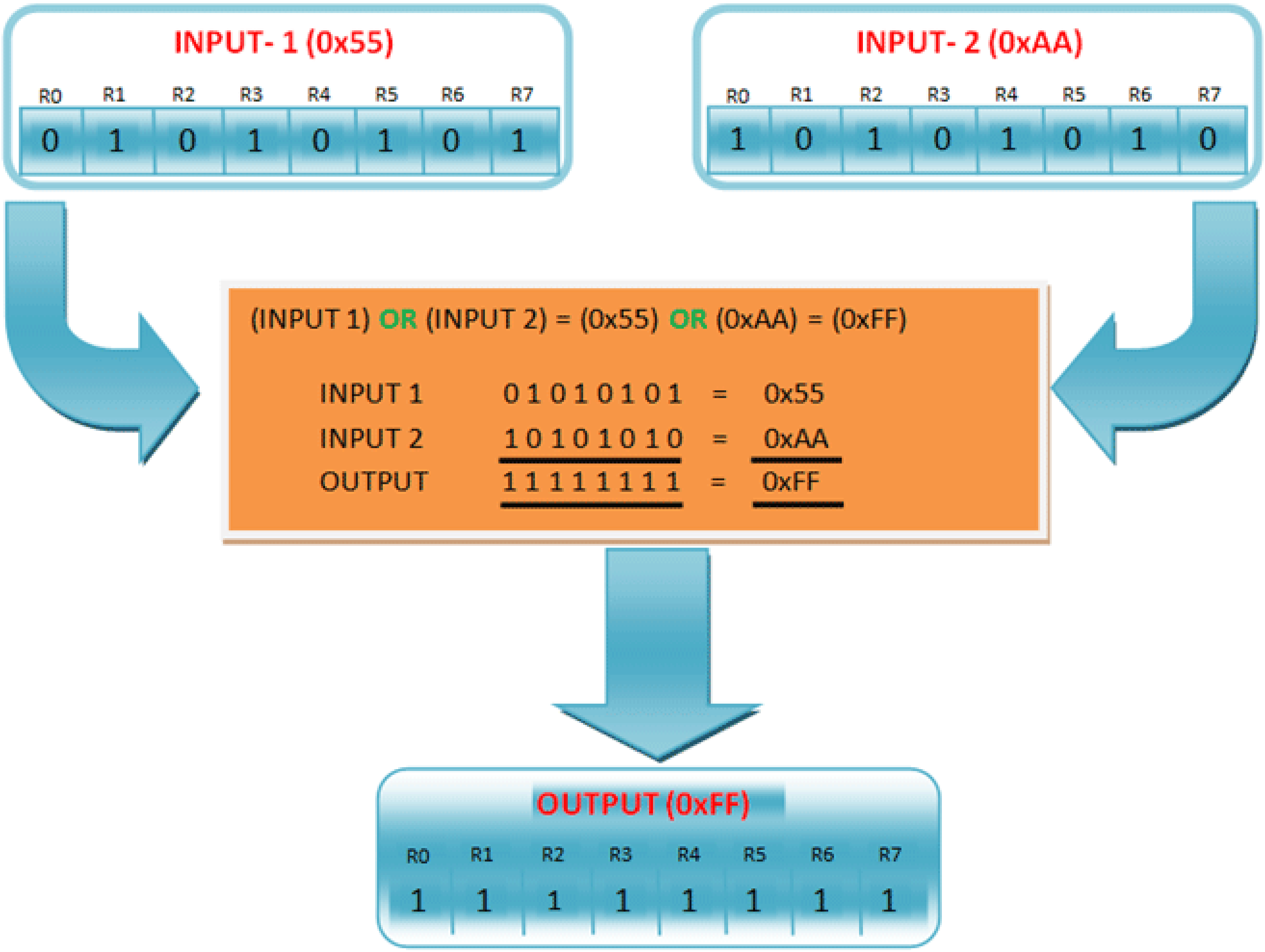


Fig:1.2: CPU Operation

The CPU takes values from two input registers INPUT-1 and INPUT-2, performs the logical operation and stores the value into the OUTPUT register. All this happens in 1 execution cycle.

1.3.1.4 Architecture of AVR-

The AVR microcontrollers are based on the advanced RISC architecture and consist of 32 x 8-bit general purpose working registers. Within one single clock cycle, AVR can take inputs from two general purpose registers and put them to ALU for carrying out the requested operation, and transfer back the result to an arbitrary register. The ALU can perform arithmetic as well as logical operations over the inputs from the register or between the register and a constant. Single register operations like taking a complement can also be executed in ALU. We can see that AVR does not have any register like accumulator as in 8051 family of microcontrollers; the operations can be performed between any of the registers and can be stored in either of them.

AVR follows Harvard Architecture format in which the processor is equipped with separate memories and buses for Program and the Data information. Here while an instruction is being executed, the next instruction is pre-fetched from the program memory.

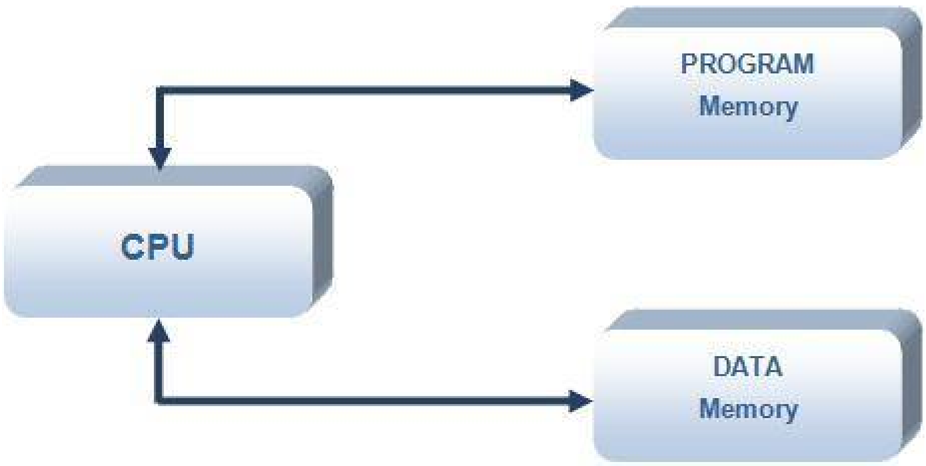


Fig1.3: CPU architecture

Since AVR can perform single cycle execution, it means that AVR can execute 1 million instructions per second if cycle frequency is 1MHz. The higher is the operating frequency of the controller, the higher will be its processing speed. We need to optimize the power consumption with processing speed and hence need to select the operating frequency accordingly.

1.3.1.5 AT-MEGA 16

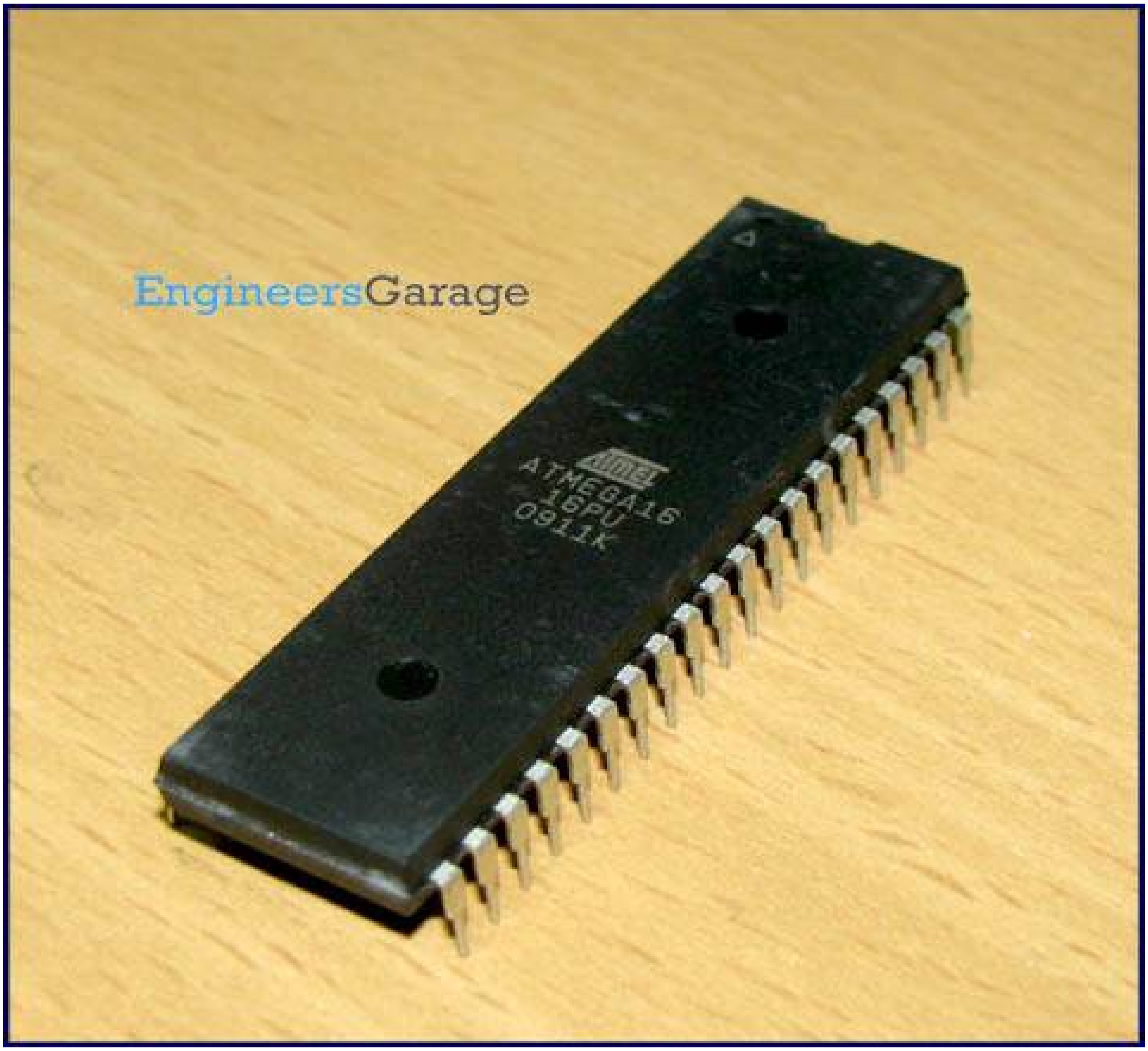


Fig: 1.4: ATMEGA16

ATmega16 is an 8-bit high performance microcontroller of Atmel's Mega AVR family with low power consumption. Atmega16 is based on enhanced RISC (Reduced Instruction Set Computing)

ATmega16 has 16 KB programmable flash memory, static RAM of 1 KB and EEPROM of 512 Bytes. The endurance cycle of flash memory and EEPROM is 10,000 and 100,000, respectively.

ATmega16 is a 40 pin microcontroller. There are 32 I/O (input/output) lines which are divided into four 8-bit ports designated as PORTA, PORTB, PORTC and PORTD.

ATmega16 has various in-built peripherals like USART, ADC, Analog Comparator, SPI, JTAG etc. Each I/O pin has an alternative task related to in-built peripherals. The following table shows the pin description of ATmega16.

There are two flavors for Atmega16 microcontroller:

1. **Atmega16**:- Operating frequency range is 0 – 16 MHz
2. **Atmega16L**:- Operating frequency range is 0 – 8 MHz

If we are using a crystal of 8 MHz = 8 x 10^6 Hertz = 8 Million cycles, then AVR can execute 8 million instructions.

1.3.1.6 Naming Convention.!

The **AT** refers to Atmel the manufacturer, **Mega** means that the microcontroller belong to MegaAVR category, **16** signifies the memory of the controller, which is 16KB.

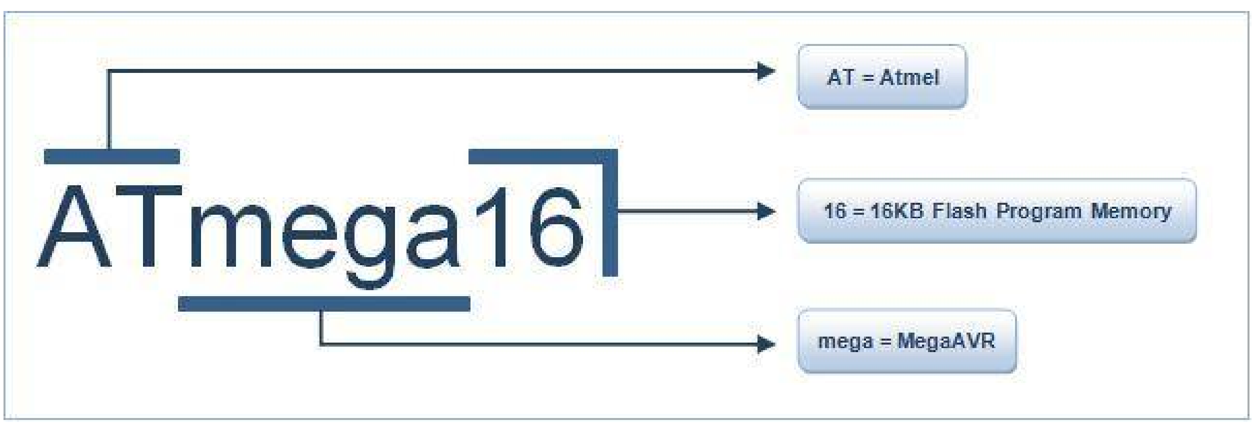


Fig:1.5:Naming Convention

1.3.1.7 Architecture Diagram: Atmega16

Following points explain the building blocks of **Atmega16 architecture** :

·

I/O Ports: Atmega16 has four (PORTA, PORTB, PORTC and PORTD) **8-bit** input-output ports.

· **Internal Calibrated Oscillator:** Atmega16 is equipped with an internal oscillator for driving its clock. By default Atmega16 is set to operate at internal calibrated oscillator of 1 MHz The maximum frequency of internal oscillator is 8Mhz. Alternatively, ATmega16 can be operated using an external crystal oscillator with a maximum frequency of 16MHz.

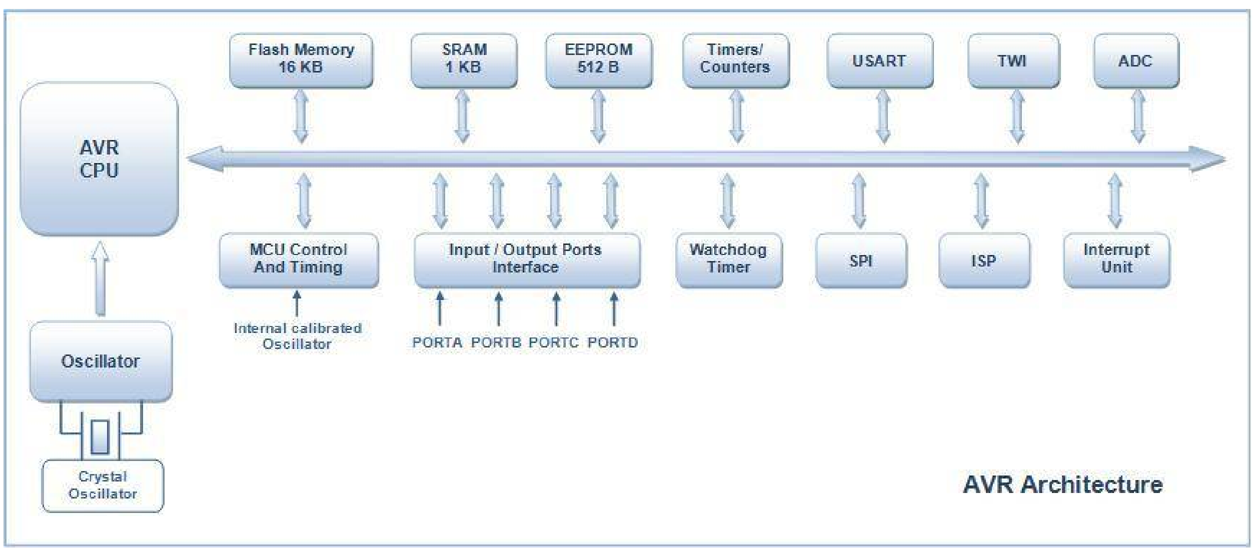


Fig 1.6: Architecture of AVR

1.3.1.8 ADC Interface:

- Atmega16 is equipped with an 8 channel ADC (**Analog to Digital Converter**) with a resolution of 10-bits. ADC reads the analog input for e.g., a sensor input and converts it into digital information which is understandable by the microcontroller.

    - **Timers/Counters:** Atmega16 consists of two 8-bit and one 16-bit timer/counter. Timers are useful for generating precision actions for e.g., creating time delays between two operations.

    - **Watchdog Timer:** It is present with internal oscillator. Watchdog timer continuously monitors and resets the controller if the code gets stuck at any execution action for more than a defined time interval.

    - **Interrupts:** Atmega16 consists of 21 interrupt sources out of which four are external. The remaining are internal interrupts which support the peripherals like USART, ADC, Timers etc.

- **USART: Universal Synchronous and Asynchronous Receiver and Transmitter** interface is available for interfacing with external device capable of communicating serially (data transmission bit by bit).

General Purpose Registers: Atmega16 is equipped with 32 general purpose registers which are coupled directly with the Arithmetic Logical Unit (ALU) of CPU.

1.3.1.9 Memory:

Atmega16 consist of three different memory sections:

1. **Flash EEPROM**: Flash EEPROM or simple flash memory is used to store the program dumped or burnt by the user on to the microcontroller. It can be easily erased electrically as a single unit. Flash memory is non-volatile i.e., it retains the program even if the power is cut-off. Atmega16 is available with 16KB of in system programmable Flash EEPROM.

2. **Byte Addressable EEPROM**: This is also a nonvolatile memory used to store data like values of certain variables. Atmega16 has 512 bytes of EEPROM, this memory can be useful for storing the lock code if we are designing an application like electronic door lock.

3. **SRAM**: Static Random Access Memory, this is the volatile memory of microcontroller i.e., data is lost as soon as power is turned off. Atmega16 is equipped with 1KB of internal SRAM. A small portion of SRAM is set aside for general purpose registers used by CPU and some for the peripheral subsystems of the microcontroller.

- **ISP:** AVR family of controllers have **In System Programmable** Flash Memory which can be programmed without removing the IC from the circuit, ISP allows to reprogram the controller while it is in the application circuit.

- **SPI: Serial Peripheral Interface**, SPI port is used for serial communication between two devices on a common clock source. The data transmission rate of SPI is more than that of USART.

- **TWI: Two Wire Interface** (TWI) can be used to set up a network of devices, many devices can be connected over TWI interface forming a network, the devices can simultaneously transmit and receive and have their own unique address.

- **DAC:** Atmega16 is also equipped with a **Digital to Analog Converter** (DAC) interface which can be used for reverse action performed by ADC. DAC can be used when there is a need of converting a digital signal to analog signal.

1.3.1.10 INSTRUCTION SETS-

There are two prevalent instruction set architectures:

Complex Instruction Set Architecture (**CISC**) : The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction.

Reduced Instruction Set Architecture (**RISC**): RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program
Subsequent sections will discuss **RISC, CISC** designs and their characteristics.

SEMANTIC GAP

Both **RISC and CISC architectures** have been developed as an attempt to cover the semantic gap.

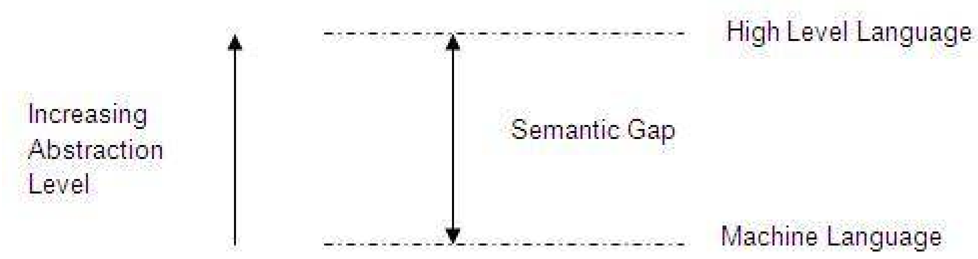


Fig:1.7: Semantic Gap

With an objective of improving efficiency of software development, several powerful programming languages have come up, viz., Ada, C, C++, Java, etc. They provide high level of abstraction, conciseness and power. By this evolution the semantic gap grows. To enable efficient compilation of high level language programs, **CISC and RISC designs** are the two options.
CISC designs involve very complex architectures including a large number of instructions and addressing modes, whereas RISC designs involve simplified instruction set and adapt it to the real requirements of user programs.

PIN DIAGRAM

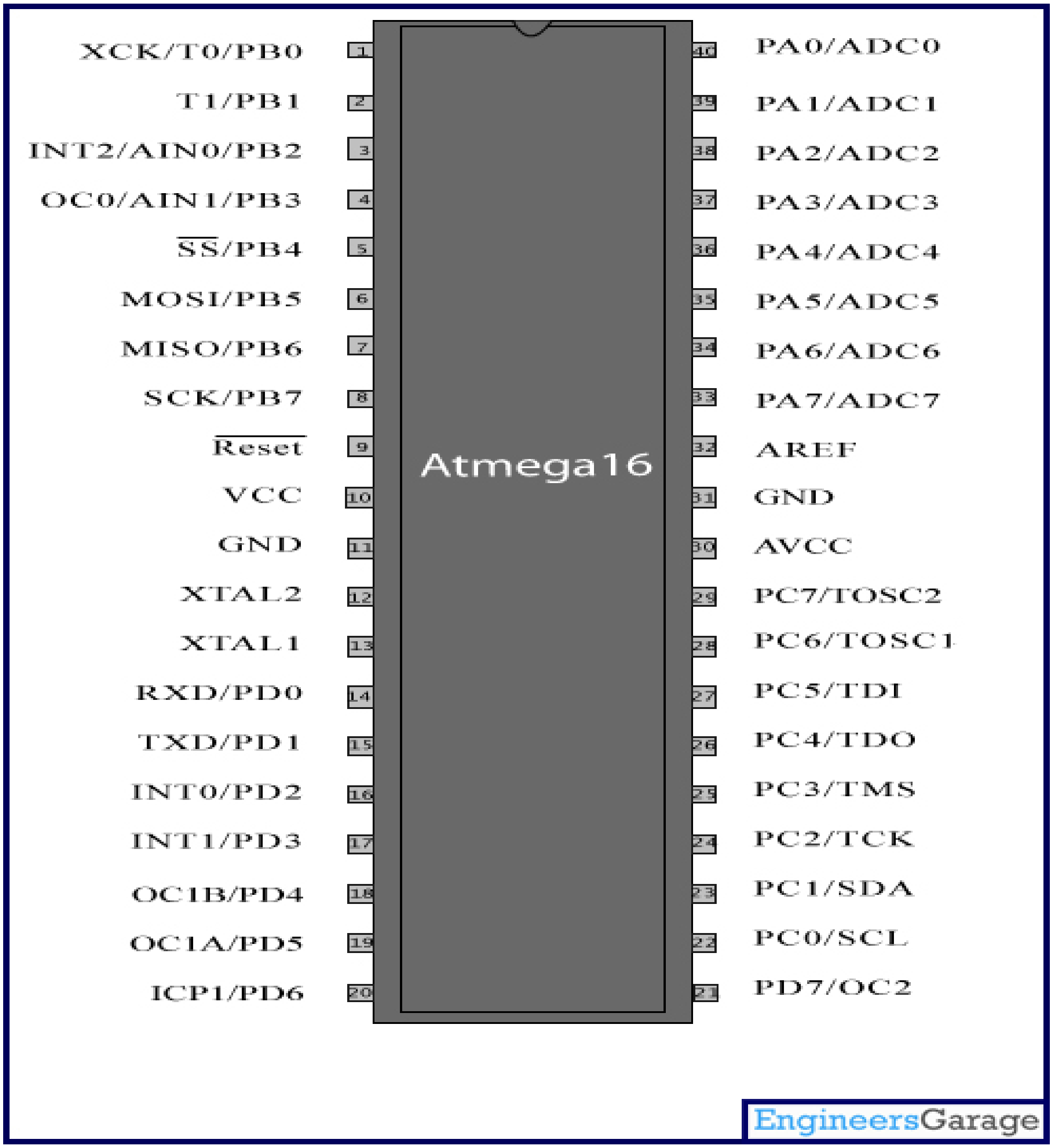


Fig:1.8 : PIN Diagram of ATMEGA16

1.3.1.11 PIN DESCRIPTION

Pin No.	Pin name	Description	Alternate Function
1	(XCK/T0) PB0	I/O PORTB, Pin 0	T0: Timer0 External Counter Input. XCK : USART External Clock I/O
2	(T1) PB1	I/O PORTB, Pin 1	T1:Timer1 External Counter Input
3	(INT2/AIN0) PB2	I/O PORTB, Pin 2	AIN0: Analog Comparator Positive I/P INT2: External Interrupt 2 Input
4	(OC0/AIN1) PB3	I/O PORTB, Pin 3	AIN1: Analog Comparator Negative I/P OC0 : Timer0 Output Compare Match Output
5	(SS) PB4	I/O PORTB, Pin 4	In System Programmer (ISP) Serial Peripheral Interface (SPI)
6	(MOSI) PB5	I/O PORTB, Pin 5	
7	(MISO) PB6	I/O PORTB, Pin 6	
8	(SCK) PB7	I/O PORTB, Pin 7	
9	RESET	Reset Pin, Active Low Reset	
10	Vcc	Vcc = +5V	
11	GND	GROUND	
12	XTAL2	Output to Inverting Oscillator Amplifier	
13	XTAL1	Input to Inverting Oscillator Amplifier	
14	(RXD) PD0	I/O PORTD, Pin 0	USART Serial Communication Interface
15	(TXD) PD1	I/O PORTD, Pin 1	
16	(INT0) PD2	I/O PORTD, Pin 2	External Interrupt INT0
17	(INT1) PD3	I/O PORTD, Pin 3	External Interrupt INT1
18	(OC1B) PD4	I/O PORTD, Pin 4	PWM Channel Outputs
19	(OC1A) PD5	I/O PORTD, Pin 5	
20	(ICP) PD6	I/O PORTD, Pin 6	Timer/Counter1 Input Capture Pin
21	PD7 (OC2)	I/O PORTD, Pin 7	Timer/Counter2 Output Compare Match Output
22	PC0 (SCL)	I/O PORTC, Pin 0	TWI Interface
23	PC1 (SDA)	I/O PORTC, Pin 1	

24	PC2 (TCK)	I/O PORTC, Pin 2	JTAG Interface
25	PC3 (TMS)	I/O PORTC, Pin 3	
26	PC4 (TDO)	I/O PORTC, Pin 4	
27	PC5 (TDI)	I/O PORTC, Pin 5	
28	PC6 (TOSC1)	I/O PORTC, Pin 6	Timer Oscillator Pin 1
29	PC7 (TOSC2)	I/O PORTC, Pin 7	Timer Oscillator Pin 2
30	AVcc	Voltage Supply = Vcc for ADC	
31	GND	GROUND	
32	AREF	Analog Reference Pin for ADC	
33	PA7 (ADC7)	I/O PORTA, Pin 7	ADC Channel 7
34	PA6 (ADC6)	I/O PORTA, Pin 6	ADC Channel 6
35	PA5 (ADC5)	I/O PORTA, Pin 5	ADC Channel 5
36	PA4 (ADC4)	I/O PORTA, Pin 4	ADC Channel 4
37	PA3 (ADC3)	I/O PORTA, Pin 3	ADC Channel 3
38	PA2 (ADC2)	I/O PORTA, Pin 2	ADC Channel 2
39	PA1 (ADC1)	I/O PORTA, Pin 1	ADC Channel 1
40	PA0 (ADC0)	I/O PORTA, Pin 0	ADC Channel 0

Table:1.4: PIN Description

1.3.1.12 FEATURES OF AT-MEGA 16

High-performance, Low-power AVR 8-bit Microcontroller

Advanced RISC Architecture
- 131 Instructions - Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Up to 16 MIPS Throughput at 16MH
- Fully Static Operation
- On-chip 2-cycle Multiplier

Non-Volatile Program and Data Memories
- 16k Bytes of In-System Self-Programmable Flash
- Optional Boot Code Section with Independent Lock Bits
- 512 Bytes EEPROM
- Programming Lock for Software Security

JTAG Interface
- Boundary-scan Capabilities According to the JTAG Standard
- Extensive On-chip Debug Support
- Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

Peripheral Features
- On-chip Analog Comparator
- Programmable Watchdog Timer with Separate On-chip Oscillator
- Master/Slave SPI Serial Interface
- Two 8-bit Timer/Counters with Separate Prescalar, Compare
- One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- Programmable Serial USART
- 8-channel, 10-bit ADC
- Byte-oriented Two-wire Serial Interface

Special Microcontroller Features
- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

I/O and Packages
- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, and 44-pad MLF

Operating Voltages
- 4.5-5.5V for ATmega16

Speed Grades
- 0-16MHz for ATmega16

Power Consumption at 4 MHz, 3V, 35 °C
- Active: 1.1mA
- Idle Mode: 0.35mA
- Power-down Mode: < 1µA

1.3.1.13 IN-BUILT PERIPHERALS OF AT –MEGA 16

- Serial communication (Data receive) using AVR Microcontroller (ATmega16) USART

Communication between two entities is important for the information flow to take place. In general the information transport system can be parallel in which the complete byte of data is sent at a time, with each bit having a separate dedicated line or it can be serial where only one communication line is available which is shared by all the bits sequentially. The pros and cons of these two systems are equivalent and selection between the two depends on the application.

Data can be exchanged using parallel or serial techniques. Setup for parallel data transfer is not cost effective but is a very fast method of communication. Serial communication is cost effective because it requires only a single line of connection but on the other hand is a slow process in comparison to parallel communication. This article explains serial communication of AVR microcontroller (ATmega16) with PC. The data is transmitted from the controller using RS232 standard and displayed on the PC using Hyper Terminal.

- How to use inbuilt ADC of AVR microcontroller (ATmega16)

Microcontroller understands only digital language. However, the inputs available from the environment to the microcontroller are mostly analog in nature, i.e., they vary continuously with time. In order to understand the inputs by the digital processor, a device called Analog to Digital Converter (ADC) is used. As the name suggests this peripheral gathers the analog information supplied from the environment and converts it to the controller understandable digital format, microcontroller then processes the information and provides the desired result at the output end.

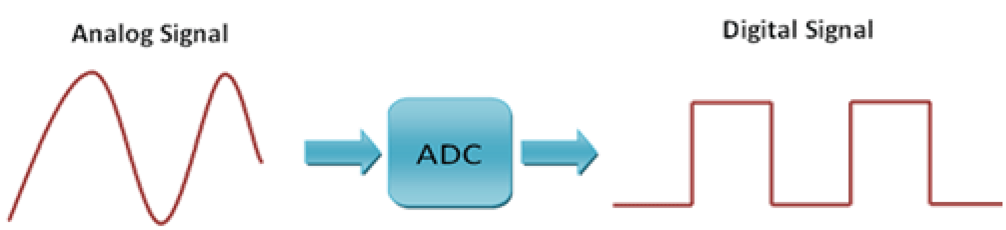


Fig1.9: ADC OF AVR

- How to use inbuilt analog comparator of AVR microcontroller

Analog comparator is a device which compares two input voltages and generates output accordingly. The article on IR sensor explains the use of comparator in sensor designing. Comparators form an integral part of circuit designing in majority of the applications. AVR microcontrollers have in-built analog comparator. Using the in-built analog comparator of AVR, the controller can be used to compare the signal and process the signal as well. This reduces the external comparator components on our circuits. In this article proximity sensor is designed using in-built analog comparator of ATmega16.

- SPI (serial peripheral interface) using AVR microcontroller (ATmega16)

There are different protocols for serial communication between two deceives like, USART, SPI, I2C etc. Before selecting any communication protocol, data transfer rate is an important parameter. SPI transfers data at high speed data. AVR microcontroller contains on chip SPI interface. This article will explore the hardware configuration and programming of SPI.

- How to disable JTAG of AVR microcontroller

JTAG stands for "Joint Test Action Group" which was standardized as the IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture in 1990. JTAG is generally used in IC debugging and device programming. Atmega16 consists of one JTAG port which shares four pins with PORTC. Until JTAG port is not disabled, these pins can't be used as normal I/O pins.

1.3.1.14 Various microcontroller of MegaAVR series:

ATmega8 and Atmega32 are other members of MegaAVR series controllers. They are quite similar to ATmega16 in architecture. Low power version MegaAVR controllers are also available in markets. The following table shows the comparison between different members of MegaAVR family:

Part Name	ROM	RAM	EEPROM	I/0 Pins	Timer	Interrupts	OperationVoltage	Operating frequency	Packaging
ATmega 8	8KB	1KB	512B	23	3	19	4.5-5.5 V	0-16 MHz	28
ATmega 8L	8KB	1KB	512B	23	3	19	2.7-5.5 V	0-8 MHz	28
ATmega 16	16KB	1KB	512B	32	3	21	4.5-5.5 V	0-16 MHz	40
ATmega 16L	16KB	1KB	512B	32	3	21	2.7-5.5 V	0-8 MHz	40
ATmega 32	32KB	2KB	1KB	32	3	21	4.5-5.5 V	0-16 MHz	40
ATmega 32L	32KB	2KB	1KB	32	3	21	2.7-5.5 V	0-8 MHz	40

Table:1.5:MegaAVR Series

1.3.1.15 PROCESSOR

A processor is an electronic device capable of manipulating data in a way specified by a sequence of instructions.

1.3.1.16 Program Memory

All AVR microcontrollers have some amount of 16 bit wide non-volatile flash memory for program storage, from 1 KB up to 256 KB (or, 512-128K typical program words). The program memory holds the executable program opcodes and static data tables. Program memory is linearly addressed, and so mechanisms like page banking or segment registers are not required to call any function, regardless of its location in program memory.

AVRs cannot use external program memory; the flash memory on the chip is the only program memory available to the AVR core.

The flash program memory can be reprogrammed using a programming tool, the most popular being those that program the chip *in situ* and are called in-system programmers (ISP). Atmel AVRs can also be reprogrammed with a high-voltage parallel or serial programmer, and via JTAG (again, *in situ*) on certain chips. The flash memory in an AVR can be re-programmed at least 10,000 times.

Many of the newer AVRs (MegaAVR series) have the capability to self-program the flash memory. This functionality is used mainly by bootloaders.

1.3.1.17 Data Memory

Data Memory includes the registers, the I/O registers, and internal SRAM.

The AVR has thirty-two general purpose eight-bit registers (R0 to R31), six of which can be used in pairs as sixteen-bit pointers (X, Y, and Z).

All AVR microcontrollers have some amount of RAM, from 32 bytes up to several KB. This memory is byte addressable. The register file (both general and special purpose) is mapped into the first addresses and thus accessible also as RAM. Some of the tiniest AVR microcontrollers have only the register file as their RAM.

The data address space consists of the register file, I/O registers, and SRAM. The working registers are mapped in as the first thirty-two memory spaces (0000_{16}-$001F_{16}$) followed by the reserved space for up to 64 I/O registers (0020_{16}-$005F_{16}$). The actual usable SRAM starts after both these sections (address 0060_{16}). (Note that the I/O register space may be larger on some more extensive devices, in which case the beginning address of SRAM will be higher.) Even though there are separate addressing schemes and optimized opcodes for register file and I/O register access, they can still be addressed and manipulated as if they were SRAM.

The I/O registers (and the program counter) are reset to their default starting values when a reset occurs. The registers and the rest of SRAM have initial random values, so typically one of the first things a program does is clear them to all zeros or load them with some other initial value.

The registers, I/O registers, and SRAM never wear out, no matter how many times they are written.

1.3.1.18 External Data Memory

Some of the higher pin-count AVR microcontrollers allow for external expansion of the data space, addressable up to 64 KB. When enabled, external SRAM is overlaid by internal SRAM; an access to address 0000_{16} in the data space will always resolve to on-chip memory. Depending on the amount of on-chip SRAM present in the particular AVR, anywhere from 512 bytes to several KB of external RAM will not be accessible. This usually does not cause a problem.

The support circuitry required is described in the datasheet for any device that supports external data memory, such as the Mega 162, in the "External Memory Interface" section. The support circuitry is minimal, consisting of a '573 or similar latch, and potentially some chip select logic. The SRAM chip select may be tied to a logic level that permanently enables the chip, or it may be driven by a pin from the AVR. For an SRAM of 32 KB or less, one option is to use a higher-order address line to drive the chip select line to the SRAM.

1.3.1.19 EEPROM Storage

Almost all AVR microcontrollers have internal EEPROM memory for non-volatile data storage. Only the Tiny11 and Tiny28 have no EEPROM.

EEPROM memory is not directly mapped in either the program or data space, but is instead accessed indirectly as a peripheral, using I/O registers. Many compilers available for the AVR hide some or all of the details of accessing EEPROM. IAR's C compiler for the AVR recognizes the compiler-specific keyword `eeprom` on a variable declaration. Thereafter, a person writes code to read and write that variable with the same standard C syntax as normal variables (in RAM), but the compiler generates code to access the EEPROM instead of regular data memory.

Atmel's datasheets indicate that the EEPROM can be re-written a minimum of 100,000 times. An application must implement a wear-leveling scheme if it writes to the EEPROM so frequently that it will reach the write limit before it reaches the expected lifetime of the device. AVRs ship from the factory with the EEPROM erased, i.e. the value in each byte of EEPROM is FF.

Many of the AVRs have errata about writing to EEPROM address 0 under certain power conditions (usually during brownout), and so Atmel recommends that programs not use that address in the EEPROM.

1.3.1.20 Fuse Settings

A fuse is an EEPROM bit that controls low level features and pin assignments. Fuses are not accessible by the program; they can only be changed by a chip programmer. Fuses control features which must be set before the chip can come out of reset and begin executing code.

The most frequently modified fuses include:

1. Oscillator/crystal characteristics, including drive strength and start-up time.
2. JTAG pins used for JTAG or GPIO
3. RESET pin used as a reset input, debugWire, or GPIO
4. Brown Out Detect (BOD) enable and BOD voltage trigger points

There is also a fuse to enable serial in-system programming, which is set by default. If it is set incorrectly, the only way to program the chip is by using a high-voltage programmer, such as the STK-500, AVR Dragon, or third-party programmer. A developer is therefore cautioned to be careful when manipulating fuses.

1.3.1.21 Reset

The AVR's RESET pin is an active-low input that forces a reset of the processor and its integrated peripherals. The line can be driven by an external power-on reset generator, a voltage supervisor (which asserts RESET when the power supply voltage drops below a predefined threshold), or another component in a larger system. For example, if the AVR is managing a few sensors and servos as part of a large integrated system, another controller might observe some condition that justifies resetting the AVR; it could do so by asserting the AVR's RESET line.

AVRs also include a watchdog timer, which can reset the processor when it times out. The watchdog timer must be reset periodically to prevent it from timing out. Failure to reset the watchdog timer is usually an indication that the program code has failed (locked up, entered an infinite loop, or otherwise gone astray), and the processor should be reset. On some AVRs the watchdog can be programmed to issue an interrupt instead of resetting the processor. This functionality can be used to wake up the AVR from a sleep mode.

The RESET pin is used for in-system serial programming, as a GPIO, or for debugWIRE low pin count debugging, depending on the chip and the programming of the fuse bits. If the reset functionality of that pin is disabled, it cannot be recovered by in-system serial programming, and another method such as high-voltage programming must be used.

1.3.1.22 Interrupts

AVRs support multiple interrupt sources, both internal and external. An interrupt could be from an internal peripheral reaching a certain state (i.e. character received on UART), or from an external event like a certain level on a pin. Each interrupt source causes a jump to a specific location in memory. That location is expected to contain either a RETI (Return from Interrupt) instruction to essentially ignore the interrupt, or a jump to the actual interrupt handler.

Most AVRs have at least one dedicated external interrupt pin (INT0). Older AVRs can trigger an interrupt on a high or low level, or on a falling edge. Newer AVRs add more options, such as triggering on the rising edge or either edge. Additionally, many of the newer AVRs implement pin-change interrupts for all pins in groups of eight, eliminating the need for polling the pins. The pin-change interrupt handler must examine the state of the pins that are associated with that interrupt vector, and determine what action to take.

Due to button bounce issues, it is considered poor design to connect a push button or other user input directly to an interrupt pin; some debouncing or other signal conditioning must be interposed so that the signal from the button does not violate the setup and hold times required on the interrupt pins.

1.3.1.23 General Purpose I/O Ports

General Purpose I/O, or GPIO, pins are the digital I/O for the AVR family. These pins are true push-pull outputs. The AVR can drive a high or low level, or configure the pin as an input with or without a pull-up. GPIOs are grouped into "ports" of up to 8 pins, though some AVRs do not have enough pins to provide all 8 pins in a particular port, e.g. the Mega48/88/168 does not have a PortC7 pin. Control registers are provided for setting the data direction, output value (or pull-up enabled), and for reading the value on the pin itself. An individual pin can be accessed using bitwise manipulation instructions.

Each port has 3 control registers associated with it, DDRx, PORTx, and PINx. Each bit in those registers controls one GPIO pin, i.e. bit 0 in DDRA controls the data direction for PortA0 (often abbreviated PA0), and bit 0 in PORTA will control the data (or pullup) for PA0.

The DDR (Data Direction Register) controls whether the pin is an input or an output. When the pin is configured as an output, the corresponding bit in the PORT register will control the drive level to the pin, high or low. When the pin is configured as an input, the bit in the PORT register controls whether a pull-up is enabled or disabled on that pin. The PIN (Port Input) register was read-only on earlier AVRs, and was used to read the value on the port pin, regardless of the data direction. Newer AVRs allow a write to the PIN register to toggle the corresponding PORT bit, which saves a few processor cycles when bit-banging an interface.

1.3.1.24 Timer/Counters

All AVRs have at least one 8-bit timer/counter. For brevity, a timer/counter is usually referred to as simply a timer.

Some of the Tiny series have only one 8-bit timer. At the high end of the Mega series, there are chips with as many as six timers (two 8-bit and four 16-bit).

A timer can be clocked directly by the system clock, by a divided-down system clock, or by an external input (rising or falling edge). Some AVRs also include an option to use an external crystal, asynchronous to the system clock, which can be used for maintaining a real-time clock with a 32.768 kHz crystal.

In normal mode, the timer counts up to the top FF_8 (or $FFFF_{16}$), roll over to zero, and set an overflow bit, which may cause an interrupt if enabled. Within the interrupt routine, the counter can be loaded with the desired value in addition to any other processing required.

The Clear Timer on Compare (CTC) mode allows for the timer to be cleared when it matches a value in the compare register, before the timer reaches the top and overflows. Clearing the timer prior to overflow manipulates the timer resolution, allowing for greater control of the output frequency of a compare match. It can also simplify the counting of an external event.

The value of a timer can be read back at any time, even while it is running. (There is a specific sequence documented in the datasheets to read back a 16-bit timer so that a consistent result is returned, since the AVR can only move 8 bits at a time.) A timer can be halted temporarily by changing its clock input to "no clock source," then resumed by re-selecting the previous clock input.

1.3.1.25 PWM (Pulse Width Modulation)

Many of the AVRs include a compare register for at least one of the timers. The compare register can be used to trigger an interrupt and/or toggle an output pin (i.e. OC1A for Timer 1) when the timer value matches the value in the compare register. This may be done separately from the overflow interrupt, enabling the use of pulse-width modulation (PWM).

Some AVRs also include options for phase-correct PWM, or phase- and frequency-correct PWM. Phase correction is required by many motors. The ATtiny26 is unique in its inclusion of a 64 MHz high-speed PWM mode. The 64 MHz clock is generated from a PLL, and is independent of, and asynchronous to, the processor clock.

Some AVRs also include complementary outputs suitable for controlling some motors. A dead-time generator (DTG) inserts a delay between one signal falling and the other signal rising so that both signals are never high at the same time. The high-end AT90PWM series allows the dead time to be programmed as a number of system clock cycles, while other AVRs with this feature simply use 1 clock cycle for the dead time.

1.3.1.26 Compare Modulator

An Output Compare Modulator (OCM), which allows generating a signal that is modulated with a carrier frequency. OCM requires two timers, one for the carrier frequency, and the second for the signal to be modulated. OCM is available on some of the Mega series.

1.3.1.27 Serial Communication

AVR microcontrollers are in general capable of supporting a plethora of serial communication protocols and serial bus standards. The exact types of serial communication support varies between the different members of the AVR microcontroller family.

On top of support in hardware there is also often the option to implement a particular serial communication mechanism entirely in software. Typically this is used in case a particular AVR controller does not support some serial communication mechanism in hardware, the particular hardware is already in use (e.g. when two RS-232 interfaces are needed, but only one is supported in hardware), or the chip's hardware can't be used, because it shares pins with other chip functions, and such a function is already in used for the particular hardware. The latter often happens with the low-pin count AVRs in DIP packages.

Finally, there is also the possibility to use additional logic to implement a serial communication function. For example, most AVRs don't support the USB bus (some later ones do so, however). When using an AVR which doesn't support USB directly, a circuit designer can add USB functionality with a fixed-function chip such as the FTDI232 USB to RS-232 converter chip, or a general-purpose USB interface such as the PDIUSB11. Adding additional electronics is in fact necessary for some supported communication protocols, e.g. standard-compliant RS-232 communication requires adding voltage level converters like the MAX232.

The number of serial communication possibilities supported by a particular AVR can be confusing at times, in particular if the pins are shared with other chip functions. An intensive study of the particular AVR's datasheet is highly recommended. The serial communication features most commonly to be found on AVRs are discussed in the following sections.

1.3.1.28 INSTRUCTIONS

Instructions in a computer are binary numbers just like data. Different numbers, when read and executed by a processor, cause different things to happen. The instructions are also called opcodes or machine codes. Different bit patterns activate or deactivate different parts of the processing core. Every processor has its own instruction set varying in number, bit pattern and functionality.

1.3.1.29 PROGRAM

The sequence of instructions is what constitutes a program. The sequence of instructions may be altered to suit the application.

1.3.1.30 ASSEMBLY LANGUAGE

Writing and understanding such programs in binary or hexadecimal form is very difficult ,so each instructions is given a symbolic notation in English language called as mnemonics. A program written in mnemonics Form is called an assembly language program. But it must be converted into machine language for execution by processor.

1.3.1.31 ASSEMBLER

An assembly language program should be converted to machine language for execution by processor. Special software called ASSEMBLER converts a program written in mnemonics to its equivalent machine opcodes.

1.3.1.32 HIGH LEVEL LANGUAGE

A high level language like C may be used to write programs for processors. Software called compiler converts this high level language program down to machine code. Ease of programming and portability.

1.3.2) GSM SIM300

GSM

GSM (**Global System for Mobile Communications**, originally ***Groupe Spécial Mobile***, is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2G) digital cellular networks used by mobile phones, first deployed in Finland in July 1991. As of 2014 it has become the default global standard for mobile communications - with over 90% market share, operating in over 219 countries and territories.

2G networks developed as a replacement for first generation (1G) analog cellular networks, and the GSM standard originally described a digital, circuit-switched network optimized for full duplex voice telephony. This expanded over time to include data communications, first by circuit-switched transport, then by packet data transport via GPRS (General Packet Radio Services) and EDGE (Enhanced Data rates for GSM Evolution or EGPRS).

Subsequently, the 3GPP developed third-generation (3G) UMTS standards followed by fourth-generation (4G) LTE Advanced standards, which do not form part of the ETSI GSM standard.

"GSM" is a trademark owned by the GSM Association. It may also refer to the (initially) most common voice codec used, Full Rate.

1.3.2.1 Network structure

The network is structured into a number of discrete sections:

- Base Station Subsystem – the base stations and their controllers explained
- Network and Switching Subsystem – the part of the network most similar to a fixed network, sometimes just called the "core network"
- GPRS Core Network – the optional part which allows packet-based Internet connections
- Operations support system (OSS) – network maintenance

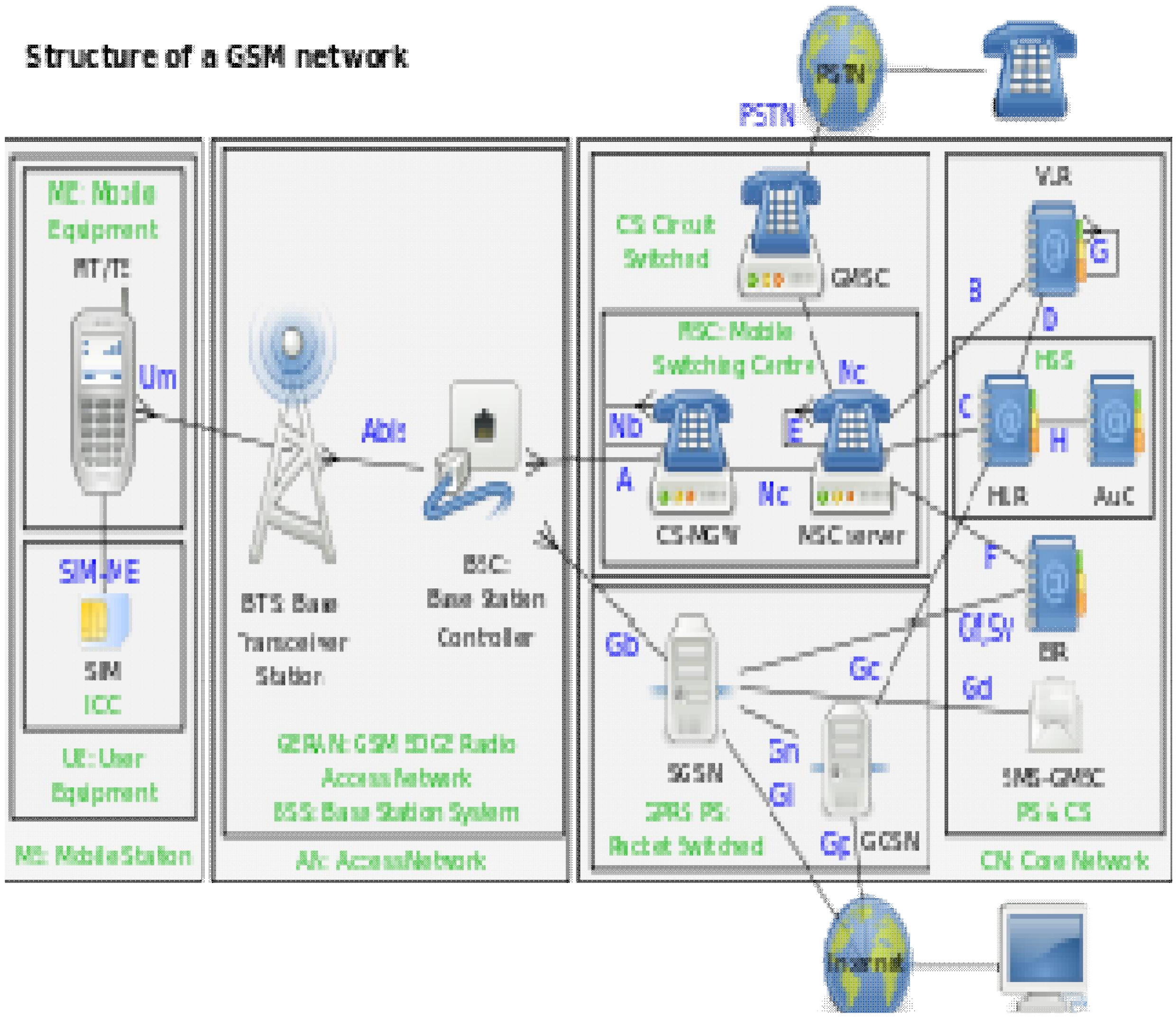


Fig:1.10 Structure of GSM network

- **Base station subsystem :**

Fig:1.11: Base Station Subsystem

GSM cell site antennas in the Deutsches Museum, Munich, Germany

GSM is a cellular network, which means that cell phones connect to it by searching for cells in the immediate vicinity. There are five different cell sizes in a GSM network—macro, micro, pico, femto, and umbrella cells. 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 rooftop level. Micro cells are cells whose antenna height is under average rooftop level; they are typically used in urban areas. Picocells are small cells whose coverage diameter is a few dozen metres; they are mainly used indoors. Femtocells are cells designed for use in residential or small business environments and connect to the service provider's network via a broadband internet connection. Umbrella cells are used to cover shadowed regions of smaller cells and fill in gaps in coverage between those cells.

Cell horizontal radius varies depending on antenna height, antenna gain, and propagation conditions from a couple of hundred meters to several tens of kilometres. The longest distance the GSM specification supports in practical use is 35 kilometres (22 mi). There are also several implementations of the concept of an extended cell,[14] 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 significant call capacity is needed indoors, like 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.

- **GSM carrier frequencies:**

GSM networks operate in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G), with most 2G GSM networks operating in the 900 MHz or 1800 MHz bands. Where these bands were already allocated, the 850 MHz and 1900 MHz bands were used instead (for example in Canada and the United States). In rare cases the 400 and 450 MHz frequency bands are assigned in some countries because they were previously used for first-generation systems.

Most 3G networks in Europe operate in the 2100 MHz frequency band. For more information on worldwide GSM frequency usage, see GSM frequency bands.

Regardless of the frequency selected by an operator, it is divided into timeslots for individual phones. This allows eight full-rate or sixteen half-rate speech channels per radio frequency. These eight radio timeslots (or burst periods) are grouped into a TDMA frame. Half-rate channels use alternate frames in the same timeslot. The channel data rate for all 8 channels is 270.833 kbit/s, and the frame duration is 4.615 ms.

The transmission power in the handset is limited to a maximum of 2 watts in GSM 850/900 and 1 watt in GSM 1800/1900.

- **Voice codecs:**

GSM has used a variety of voice codecs to squeeze 3.1 kHz audio into between 6.5 and 13 kbit/s. Originally, two codecs, named after the types of data channel they were allocated, were used, called Half Rate (6.5 kbit/s) and Full Rate (13 kbit/s). These used a system based on linear predictive coding (LPC). In addition to being efficient withbitrates, these codecs also made it easier to identify more important parts of the audio, allowing the air interface layer to prioritize and better protect these parts of the signal.

As GSM was further enhanced in 1997[15] with the Enhanced Full Rate (EFR) codec, a 12.2 kbit/s codec that uses a full-rate channel. Finally, with the development of UMTS, EFR was refactored into a variable-rate codec called AMR-Narrowband, which is high quality and robust against interference when used on full-rate channels, or less robust but still relatively high quality when used in good radio conditions on half-rate channel.

- **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.

- **Phone 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. A subscriber may usually contact the provider to remove the lock for a fee, utilize private services to remove the lock, or use software and websites to unlock the handset themselves.

In some countries (e.g., Bangladesh, Belgium, Brazil, Chile, Germany, Hong Kong, India, Iran, Lebanon, Malaysia, Nepal, Pakistan, Poland, Singapore, South Africa, Thailand) all phones are sold unlocked.

1.3.2.2 GSM SIM300:

This is a plug and play GSM Modem with a simple to interface serial interface. Use it to send SMS, make and receive calls, and do other GSM operations by controlling it through simple AT commands from micro controllers and computers. It uses the highly popular SIM300 module for all its operations. It comes with a standard RS232 interface which can be used to easily interface the modem to micro controllers and computers.

The modem consists of all the required external circuitry required to start experimenting with the SIM300 module like the power regulation, external antenna, SIM Holder, etc.

1.3.2.3 Features :

- Uses the extremely popular SIM300 GSM module

- Provides the industry standard serial RS232 interface for easy connection to computers and other devices

- Provides serial TTL interface for easy and direct interface to microcontrollers

- Power, RING and Network LEDs for easy debugging

- Onboard 3V Lithium Battery holder with appropriate circuitry for providing backup for the modules' internal RTC

- Can be used for GSM based Voice communications, Data/Fax, SMS,GPRS and TCP/IP stack

- Can be controlled through standard AT commands

- Comes with an onboard wire antenna for better reception.

- Board provides an option for adding an external antenna through an SMA connector

- The SIM300 allows an adjustable serial baud rate from 1200 to 115200 bps (9600 default)

- Modem a low power consumption of 0.25 A during normal operations and around 1 A during transmission

- Operating Voltage: 7 – 15V AC or DC (board has onboard rectifier)

With a tiny configuration of 40mm x 33mm x 2.85 mm , SIM300 can fit almost all the space requirement in your application, such as Smart phone, PDA phone and other mobile device.

The physical interface to the mobile application is made through a 60 pins board-to-board connector, which provides all hardware interfaces between the module and customers' boards except the RF antenna interface.

- The keypad and SPI LCD interface will give you the flexibility to develop customized applications.

- Two serial ports can help you easily develop your applications.

- Two audio channels include two microphones inputs and two speaker outputs. This can be easily configured by AT command.

SIM300 provide RF antenna interface with two alternatives: antenna connector and antenna pad. The antenna connector is MURATA MM9329-2700. And customer's antenna can be soldered to the antenna pad.

The SIM300 is designed with power saving technique, the current consumption to as low as 2.5mA in SLEEP mode.

The SIM300 is integrated with the TCP/IP protocol Extended TCP/IP AT commands are developed for customers to use the TCP/IP protocol easily, which is very useful for those data transfer applications.

Fig : 1.12:GSM SIM300 MODEM

1.3.2.4 Power supply :

The power supply of SIM300 is from a single voltage source of VBAT= 3.4V...4.5V. In some case, the ripple in a transmit burst may cause voltage drops when current consumption rises to typical peaks of 2A, So the power supply must be able to provide sufficient current up to 2A. For the VBAT input, a local bypass capacitor is recommended. A capacitor (about 100µF, low ESR) is recommended. Multi-layer ceramic chip (MLCC) capacitors can provide the best combination of low ESR and small size but may not be cost effective. A lower cost choice may be a 100 µF tantalum capacitor (low ESR) with a small (1 µF to 10µF) ceramic in parallel, which is illustrated as following figure.

And the capacitors should put as closer as possible to the SIM300 VBAT pins.

1.3.2.5 Serial interfaces :

SIM300 provides two unbalanced asynchronous serial ports. The GSM module is designed as a DCE (Data Communication Equipment), following the traditional DCE-DTE (Data Terminal Equipment) connection, the module and the client (DTE) are connected through the following signal (as following figure shows). Autobauding supports baud rate from 1200 bps to 115200bps.

Serial port 1

- Port/TXD @ Client sends data to the RXD signal line of module
- Port/RXD @ Client receives data from the TXD signal line of module

Serial port 2

- Port/TXD @ Client sends data to the DGBRXD signal line of module
- Port/RXD @ Client receives data from the DGBTXD signal line of module

All pins of two serial ports have 8mA driver, the logic levels are described in following table

PARAMETER	MIN	MAX	UNIT
Logic low input	0	0.3*VDD_EXT	V
Logic high input	0.7 *VDD_EXT	VDD_EXT +0.3	V
Logic low output	GND	0.2	V
Logic high output	VDD_EXT -0.2	VDD_EXT	V

Table:1.6:Logic levels of Serial Port Pins

1.3.2.6 Function of Serial Port 1 & 2 supporting

Serial port 1

- Seven lines on Serial Port Interface
- Contains Data lines /TXD and /RXD, State lines /RTS and /CTS, Control lines /DTR, /DCD and RING
- Serial Port 1 can be used for CSD FAX, GPRS service and send AT command of controlling module. Serial Port 1 can use multiplexing function, but you cannot use the Serial Port 2 at the same time;
- Serial Port 1 supports the communication rate as following: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 Default as 115200bps.
- Autobauding supports the communication rate as following: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200bps.

Serial port 2

- Two lines on Serial Port Interface
- Only contains Data lines /TXD and /RXD
- Serial Port 2 only used for transmitting AT command. It cannot be used for CSD call, FAX call. And the Serial port 2 can not use multiplexing function;
- Serial port 2 supports the communication rate as following: 9600, 19200, 38400, 57600, 115200

1.3.3) VOLTAGE REGULATOR

1.3.3.1 FEATURES

- Output current in Excess of 1.0 A.
- No external component required.
- Internal thermal overload protection.
- Internal short circuit current limiting.
- Output transistor safe-area compensation.
- Output voltage offered in 2% and 4% tolerance.
- Available I n surface mount D2PAK and standard 3-lead transistor packages.
- Previous commercial temperature range has been extended to a junction temperature range of -40 degree C to +125 degree C.

1.3.3.2 DESCRIPTION

Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. The maximum current they can pass also rates them. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current and overheating (thermal protection). Many of fixed voltage regulator ICs has 3 leads. They include a hole for attaching a heat sink if necessary.

Figure No. 1.13: 7805 Voltage Regulator

1.3.4) LCD DISPLAY

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if not all Parallel Ports. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required to run them is on board. A liquid-crystal display (LCD) is a flat panel display or electronic visual display.

LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock.

LCDs are used in a wide range of applications including computer monitors, televisions, instrument panels, aircraft cockpit displays, and signage. They are common in consumer devices such as DVD players, gaming devices, clocks, watches, calculators, and telephones, and have replaced cathode ray tube (CRT) displays in most applications.

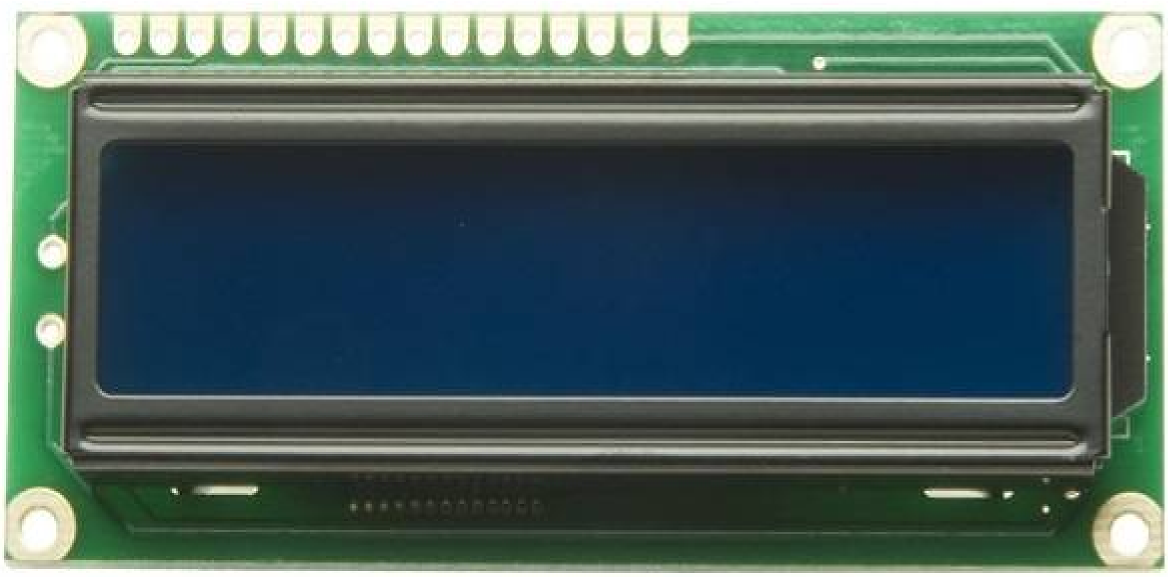


Figure No. 1.14: LCD Display

1.3.4.1 CIRCUIT DESCRIPTION

The LCD panel's Enable and Register Select is connected to the Control Port. The Control Port is an open collector / open drain output. While most Parallel Ports have internal pull-up resistors, there is a few which don't. Therefore by incorporating the two 10K external pull up resistors, the circuit is more portable for a wider range of computers, some of which may have no internal pull up resistors.

We make no effort to place the Data bus into reverse direction. Therefore we hard wire the R/W line of the LCD panel, into write mode. This will cause no bus conflicts on the data lines. As a result we cannot read back the LCD's internal Busy Flag which tells us if the LCD has accepted and finished processing the last instruction. This problem is overcome by inserting known delays into our program.

The 10k Potentiometer controls the contrast of the LCD panel. Nothing fancy here. As with all the examples, I've left the power supply out. You can use a bench power supply set to 5v or use an onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

1.3.4.2 LCD PIN DIAGRAM

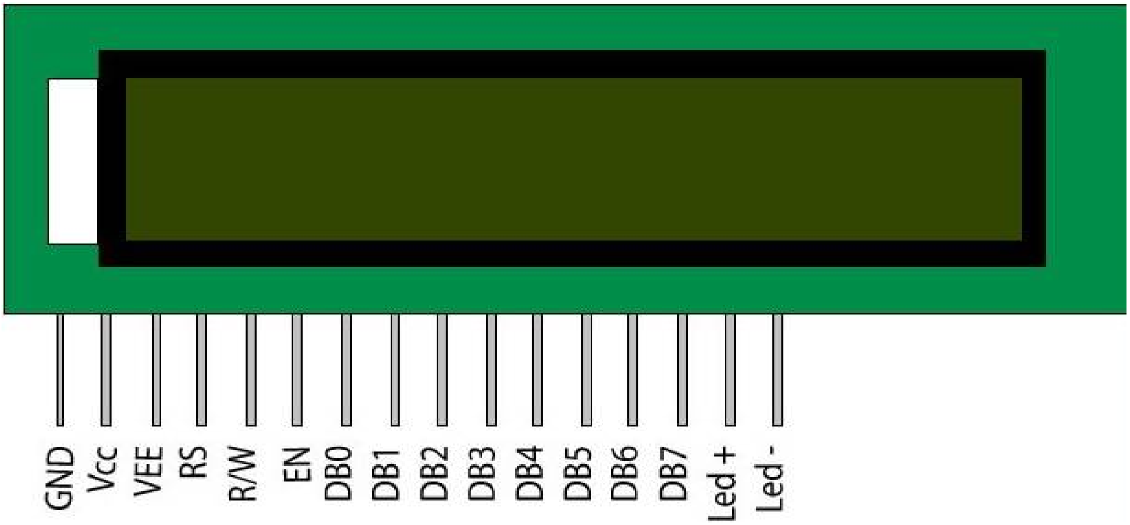


Figure No. 1.15: Pin Diagram of LCD Display

1.3.5) CRYSTAL OSCILLATORS

A crystal oscillator is an electronic oscillator circuit that uses the mechanical resonance of a vibrating crystal of piezoelectric material to create an electrical signal with a very precise frequency. Crystal oscillators are oscillators where the primary frequency determining element is a quartz crystal. Because of the inherent characteristics of the quartz crystal the crystal oscillator may be held to extreme accuracy of frequency stability. Temperature compensation may be applied to crystal oscillators to improve thermal stability of the crystal oscillator. Crystal oscillators are usually, fixed frequency oscillators where stability and accuracy are the primary considerations. For example it is almost impossible to design a stable and accurate LC oscillator for the upper HF and higher frequencies without resorting to some sort of crystal control. The frequency of older FT-243 crystals can be moved upward by crystal grinding.

Figure No. 1.16: Crystal Oscillator

1.3.6) PUSH BUTTON SWITCH

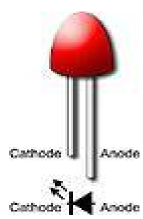
A push-button (also spelled pushbutton) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, though even many un-biased buttons (due to their physical nature) require a spring to return to their un-pushed state. Different people use different terms for the "pushing" of the button, such as press, depress, mash, and punch.

The "push-button" has been utilized in calculators, push-button telephones, kitchen appliances, and various other mechanical and electronic devices, home and commercial. Pushbuttons are often color-coded to associate them with their function so that the operator will not push the wrong button in error. Commonly used colors are red for stopping the machine or process and green for starting the machine or process.

Figure No. 1.17: Push Button Switch

1.3.7) LED

LED falls within the family of P-N junction devices. The light emitting diode (LED) is a diode that will give off visible light when it is energized. In any forward biased P-N junction there is, with in the structure and primarily close to the junction, a recombination of hole and electrons. This recombination requires that the energy possessed by the unbound free electron be transferred to another state. The process of giving off light by applying an electrical source is called electroluminescence.



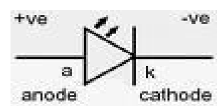


Figure No. 1.18: LED & LED Symbol

LED is a component used for indication. All the functions being carried out are displayed by led. The LED is diode which glows when the current is being flown through it in forward bias condition. The LEDs are available in the round shell and also in the flat shells. The positive leg is longer than negative leg. Early LEDs were often used as indicator lamps for electronic devices, replacing small incandescent bulbs. They were soon packaged into numeric readouts in the form of seven-segment displays, and were commonly seen in digital clocks.

LEDs have many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are now used in applications as diverse as aviation lighting ,automotive headlamps, advertising, general lighting, traffic signals, and camera flashes. However, LEDs powerful enough for room lighting are still relatively expensive, and require more precise current and heat management than compact fluorescent lamp sources of comparable output.

1.3.8) RESISTOR

The flow of charge through any material encounters an opposing force similar in many respects to mechanical friction .this opposing force is called resistance of the material .in some electric circuit resistance is deliberately introduced in form of resistor. Resistor used fall in three categories , only two of which are color coded which are metal film and carbon film resistor .the third category is the wire wound type ,where value are generally printed on the vitreous paint finish of the component. Resistors are in ohms and are represented in Greek letter omega, looks as an upturned horseshoe. Most electronic circuit require resistors to make them work properly and it is obliviously important to find out something about the different types of resistors available. Resistance is measured in ohms, the symbol for ohm is an omega ohm. 1 ohm is quite small for electronics so resistances are often given in k ohm and M ohm.

Resistors used in electronics can have resistances as low as 0.1 ohm or as high as 10 M ohm.

Figure No. 1.19: Symbol of Resistance

1.3.8.1 TESTING

Resistors are checked with an ohm meter/millimeter. For a defective resistor the ohm-meter shows infinite high reading.

1.3.8.2 COLOR CODING

Components and wires are coded are with colors to identify their value and function.

Color	Digit	Multiplier	Tolerance (%)
Black	0	10^0 (1)	
Brown	1	10^1	1
Red	2	10^2	2
Orange	3	10^3	
Yellow	4	10^4	
Green	5	10^5	0.5
Blue	6	10^6	0.25
Violet	7	10^7	0.1
Grey	8	10^8	
White	9	10^9	
Gold		10^{-1}	5
Silver		10^{-2}	10
(none)			20

Table No. 1.7: Color Coding of Resistor

All coded components will have at least two value bands and a multiplier; other bands are optional. Zero ohm resistors are made as lengths of wire wrapped in a resistor-shaped body which can be substituted for another resistor value in automatic insertion equipment. They are marked with a single black band. The 'body-end-dot' or 'body-tip-spot' system was used for radial-lead (and other cylindrical) composition resistors sometimes still found in very old equipment; the first band was given by the body color, the second band by the color of the end of the resistor, and the multiplier by a dot or band around the middle of the resistor. The other end of the resistor was colored gold or silver to give the tolerance, otherwise it was 20%.

The colors brown, red, green, blue, and violet are used as tolerance codes on 5-band resistors only. All 5-band resistors use a colored tolerance band. The blank (20%) "band" is only used with the "4-band" code (3 colored bands + a blank "band").

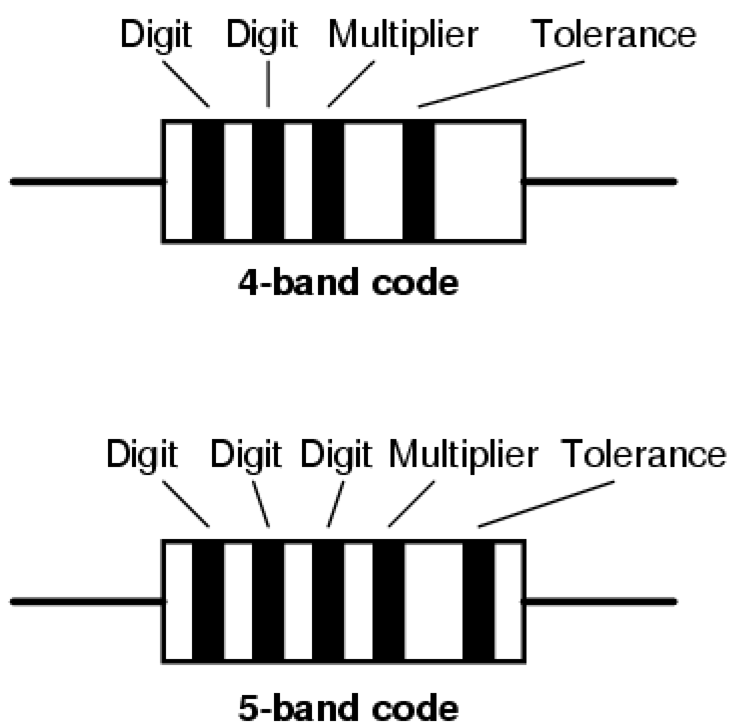


Figure No. 1.20: Color Coding of Resistor

1.3.9) CAPACITOR

In a way, a capacitor is a little like a battery. Although they work in completely different ways, capacitors and batteries both store electrical energy. If you have read How Batteries Work, then you know that a battery has two terminals. Inside the battery, chemical reactions produce electrons on one terminal and absorb electrons at the other terminal.

1.3.9.1 BASIC

Like a battery, a capacitor has two terminals. Inside the capacitor, the terminals connect to two metal plates separated by a dielectric. The dielectric can be air, paper, plastic or anything else that does not conduct electricity and keeps the plates from touching each other. You can easily make a capacitor from two pieces of aluminum foil and a piece of paper. It won't be a particularly good capacitor in terms of its storage capacity, but it will work.
In an electronic circuit, a capacitor is shown like this:

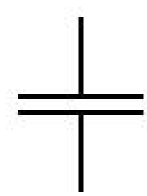


Figure No. 1.21: Symbol of Capacitor

When you connect a capacitor to a battery, here's what happens:

- The plate on the capacitor that attaches to the negative terminal of the battery accepts electrons that the battery is producing.
- The plate on the capacitor that attaches to the positive terminal of the battery loses electrons to the battery.

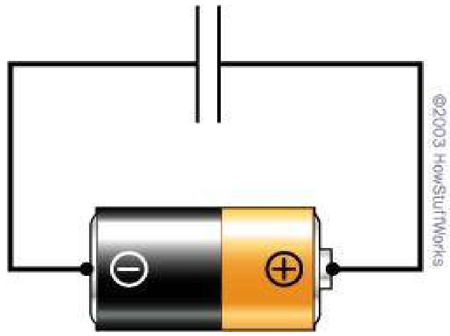


Figure No. 1.22: Capacitor & Battery Connection

1.3.9.2 TESTING

To test the capacitors, either analog meters or special digital meters with the specified function are used. The non-electrolyte capacitor can be tested by using the digital meter.

1.3.10) ADAPTER

An AC adapter, AC/DC adapter, or AC/DC converter is a type of external power supply, often enclosed in a case similar to an AC plug. Other common names include plug pack, plug-in adapter block, domestic mains adapter, ladapteine power adapter, wall wartpower brick, and power adapter. Adapters for battery-powered equipment may be described as chargers or rechargers (see also battery charger). AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from mains power. The internal circuitry of an external power supply is very similar to the design that would be used for a built-in or internal supply.

External power supplies are used both with equipment with no other source of power and with battery-powered equipment, where the supply, when plugged in, can sometimes charge the battery in addition to powering the equipment.

Use of an external power supply allows portability of equipment powered either by mains or battery without the added bulk of internal power components, and makes it unnecessary to produce equipment for use only with a specified power source; the same device can be powered from 120 VAC or 230 VAC mains, vehicle or aircraft battery by using a different adapter.

Fig:1.23: Internal circuit of adapter

Originally, most AC/DC adapters were linear power supplies, containing a transformer to convert the mains electricity voltage to a lower voltage, a rectifier to convert it to pulsating DC, and a filter to smooth the pulsating waveform to DC, with residual ripple variations small enough to leave the powered device unaffected. Size and weight of the device was largely determined by the transformer, which in turn was determined by the power output and mains frequency. Ratings over a few watts made the devices too large and heavy to be physically supported by a wall outlet. The output voltage of these adapters varied with load; for equipment requiring a more stable voltage, linear voltage regulator circuitry was added. Losses in the transformer and the linear regulator were considerable; efficiency was relatively low, and significant power dissipated as heat even when not driving a load.

In the early twenty-first century, switched-mode power supplies (SMPSs) became almost ubiquitous for this purpose. Mains voltage is rectified to a high direct voltage driving a switching circuit, which contains a transformer operating at a high frequency and outputs direct current at the desired voltage. The high-frequency ripple is more easily filtered out than mains-frequency. The high frequency allows the transformer to be small, which reduces its losses; and the switching regulator can be much more efficient than a linear regulator. The result is a much more efficient, smaller, and lighter device. Safety is ensured, as in the older linear circuit, because there is still a transformer which electrically isolates the output from the mains.

A linear circuit must be designed for a specific, narrow range of input voltages (e.g., 220–240 VAC) and must use a transformer appropriate for the frequency (usually 50 or 60 Hz), but a switched-mode supply can work efficiently over a very wide range of voltages and frequencies; a single 100–240 VAC unit will handle almost any mains supply in the world.

External AC adapters are widely used to power small or portable electronic devices. The advantages include:

- Safety – External power adapters can free product designers from worrying about some safety issues. Much of this style of equipment uses only voltages low enough not to be a safety hazard internally, although the power supply must out of necessity use dangerous mains voltage. If an external power supply is used (usually via a power connector, often of coaxial type), the equipment need not be designed with concern for hazardous voltages inside the enclosure. This is particularly relevant for equipment with lightweight cases which may break and expose internal electrical parts.

- Heat reduction – Heat reduces reliability and longevity of electronic components, and can cause sensitive circuits to become inaccurate or malfunction. A separate power supply removes a source of heat from the apparatus.
- Electrical noise reduction – Because radiated electrical noise falls off with the square of the distance, it is to the manufacturer's advantage to convert potentially noisy AC line power or automotive power to "clean", filtered DC in an external adapter, at a safe distance from noise-sensitive circuitry.

- Weight and size reduction – Removing power components and the mains connection plug from equipment powered by rechargeable batteries reduces the weight and size which must be carried.

- Ease of replacement – Power supplies are more prone to failure than other circuitry due to their exposure to power spikes and their internal generation of waste heat. External power supplies can be replaced quickly by a user without the need to have the powered device repaired.

- Configuration versatility – Externally powered electronic products can be used with different power sources as needed (e.g. 120VAC, 240VAC, 12VDC, or external battery pack), for convenient use in the field, or when traveling.

- Simplified product inventory, distribution, and certification – An electronic product that is sold and used internationally must be powered from a wide range of power sources, and must meet product safety regulations in many jurisdictions, usually requiring expensive certification by national or regional safety agencies such as Underwriters Laboratories or TechnischerÜberwachungsverein. A single version of a device may be used in many markets, with the different power requirements met by different external power supplies, so that only one version of the device need be manufactured, stocked, and tested. If the design of the device is modified over time (a frequent occurrence), the power supply design itself need not be retested (and vice versa).

Fig:1.24: **Adapter**

1.3.11) NETWORK RESISTANCE (A103J)

Network resistor is a passive circuit element that is a combination of multiple resistances. It forms a convenient solution when the user needs multiple resistances while constructing a circuit. The manner in which the constituent resistances are combined can vary according to the circuit requirement.

Hence, this type of resistor can be used to condense the circuit with additional benefits of low costs and improved resistance tolerance matching.

Network resistor also helps to reduce the number of components on the circuit and the mounting is also easy.

A pull-up resistor is used to "pull-up" a signal to prevent it from floating or creating an undefined state. Typically a digital input might be looking for a ground or low state to indicate a state. The pull-up resistor is attached to the signal line and to a level which will "pull" the line to a high level in the absence of a low signal. The effect is to provide a reliable hi or lo to an input. The pull up may be located close to an input or directly at an output to effect the results. Some electronic devices have the pull-up devices internally installed.

Fig:1.25: Pull up resistance network

Here we are using pull up resistance network to replace number of resistors with single network of resistance.

It has good resistive material which helps in providing required resistance to the circuit.

CHAPTER 2

2.1) P.C.B. DESIGNING

2.1.1 P.C.B. LAYOUT

The entire circuit can be easily assembled on a general purpose P.C.B. board respectively. Layout of desired diagram and preparation is first and most important operation in any printed circuit board manufacturing process. First of all layout of component side is to be made in accordance with available components dimensions. The following points are to be observed while forming the layout of P.C.B.

1. Between two components, sufficient space should be maintained.
2. High voltage/max dissipated components should be mounted at sufficient distance from semiconductor and electrolytic capacitors.
3. The most important points are that the components layout is making proper compromise with copper side circuit layout.

Printed circuit board (P.C.B.s) is used to avoid most of all the disadvantages of conventional breadboard. These also avoid the use of thin wires for connecting the components; they are small in size and efficient in performance.

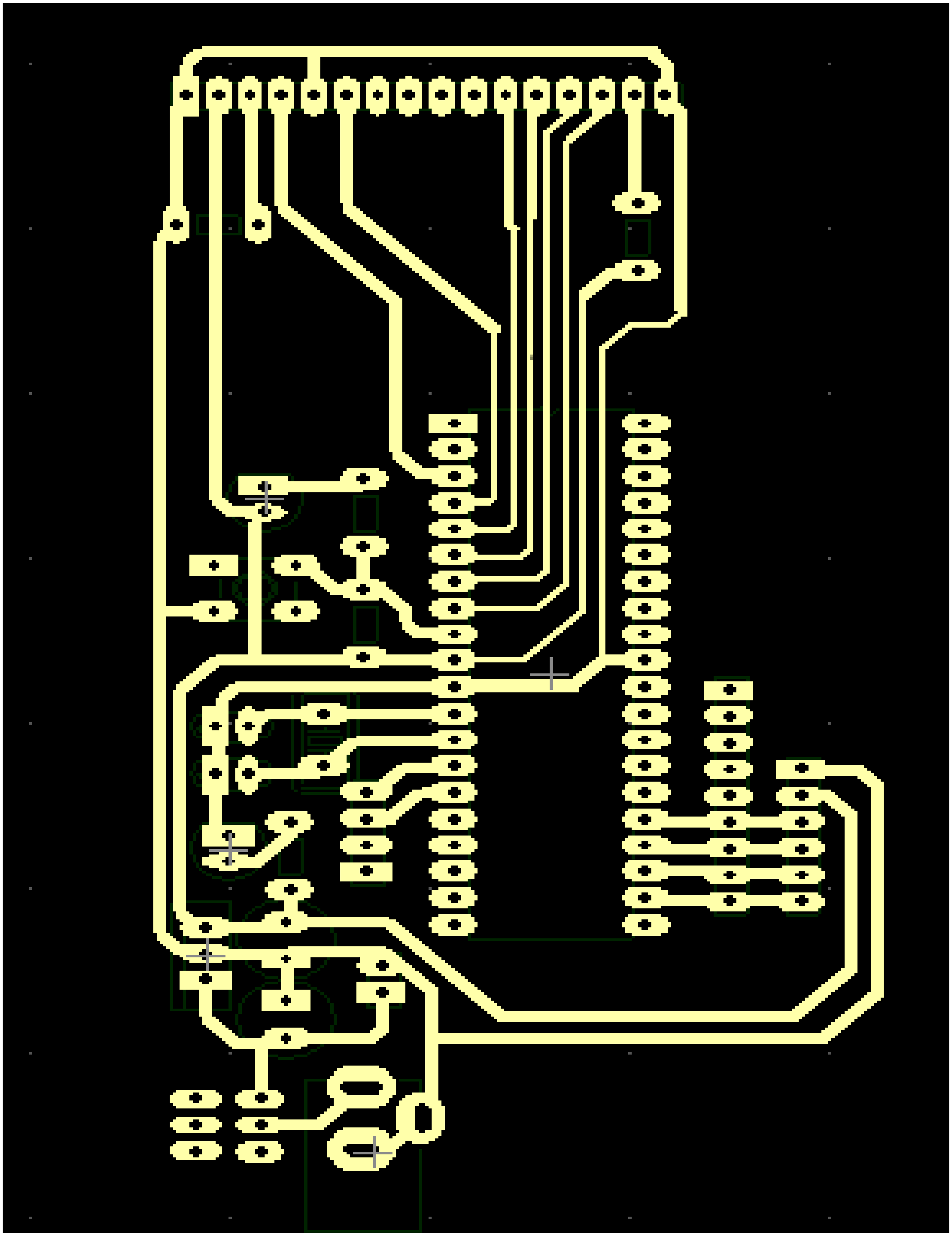


Figure No. 2.1: PCB Layout

2.1.2 PREPARING CIRCUIT LAYOUT

First of all the actual size circuit layout is to be drawn on the copper side of the copper clad board. Then enamel paint is applied on the tracks of connection with the help of a shade brush. We have to apply the paints surrounding the point at which the connection is to be made. It avoids the disconnection between the leg of the component and circuit track. After completion of painting work, it is allowed to dry.

2.1.3 DRILLING

After completion of painting work, holes 1/23inch(1mm) diameter are drilled at desired points where we have to fix the components.

2.1.4 ETCHING

The removal of excess of copper on the plate apart from the printed circuit is known as etching. From this process the copper clad board with printed circuit is placed in the solution of FeCl with 3-4 drops of HCL in it and is kept so for about 10 to 15 minutes and is taken out when all the excess copper is removed from the P.C.B.

After etching, the P.C.B. is kept in clean water for about half an hour in order to get P.C.B. away from acidic, field, which may cause poor performance of the circuit. After the P.C.B. has been thoroughly washed, paint is removed by soft piece of cloth dipped I thinner or turbine. Then P.C.B. is checked as per the layout, now the P.C.B. is ready for use.

2.1.5 SOLDERING

Soldering is the process of joining two metallic conductor the joint where two metal conductors are to be join or fused is heated with a device called soldering iron and then as allow of tin and lead called solder is applied which melts and converse the joint. The solder cools and solidifies quickly to ensure is good and durable connection between the jointed metal converting the joint solder also present oxidation.

2.1.6 SOLDERING AND DESOLDERING TECHIQUES:

These are basically two soldering techniques.

- Manual soldering with iron.
- Mass soldering.

2.2) WORKING OF PROJECT

The working of this project is controlled by a microcontroller ATMEGA16 and a GSM module is used to send and receive SMS between user and EVM. The project works in the following ways:

1. Switch on power supply.
2. Message "WELCOME TO GSM BASED VOTING" will appear on LCD.
3. Message "U CAN VOTE FOR A B C D" will be displayed on LCD after a delay.
4. SMS your vote from registered number according to the prescribed format -- *VOTE A/B/C/D/X#
5. A,B,C,D denote the parties and X denotes "None of the above" option.
6. "1 MESSAGE RECEIVED " will be displayed on LCD as soon as voter's SMS is received by the GSM module.
7. The vote will be displayed on the LCD.
8. Message "THANK YOU FOR VOTE" will be sent on to the voter's cell phone number.
9. After that, the number of votes received by each candidate will be displayed on LCD.
10. If the user fails to message according to the prescribed format, the vote will not be cast and "SORRY U CANT VOTE" will be sent to the voter's cell phone.
11. Each registered number can be used to cast vote for only one time. If a voter tries to vote twice from the registered number then the vote will not be cast and message "SORRY U CANT VOTE " will be sent to his cell phone.
12. Master Pattern "*RESULT#" can be messaged by the authorized number to display the result.
13. Pattern "*DELETE#" can be messaged by the authorized number to delete the previous values and start counting votes from 0.
14. A reset key is present to reset the microcontroller.

2.3) BLOCK DIAGRAM

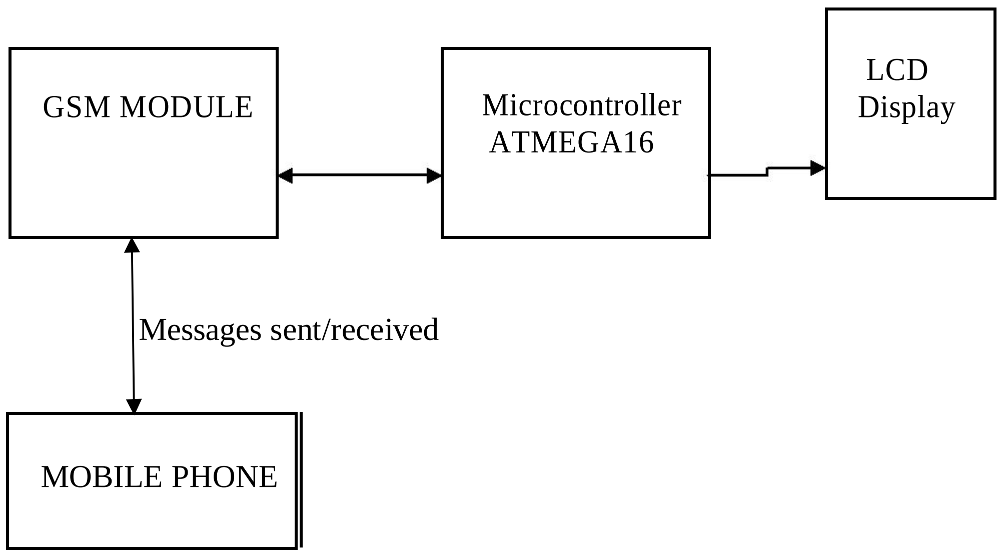


Figure No. 2.2: Block Diagram

2.4) CIRCUIT DIAGRAM

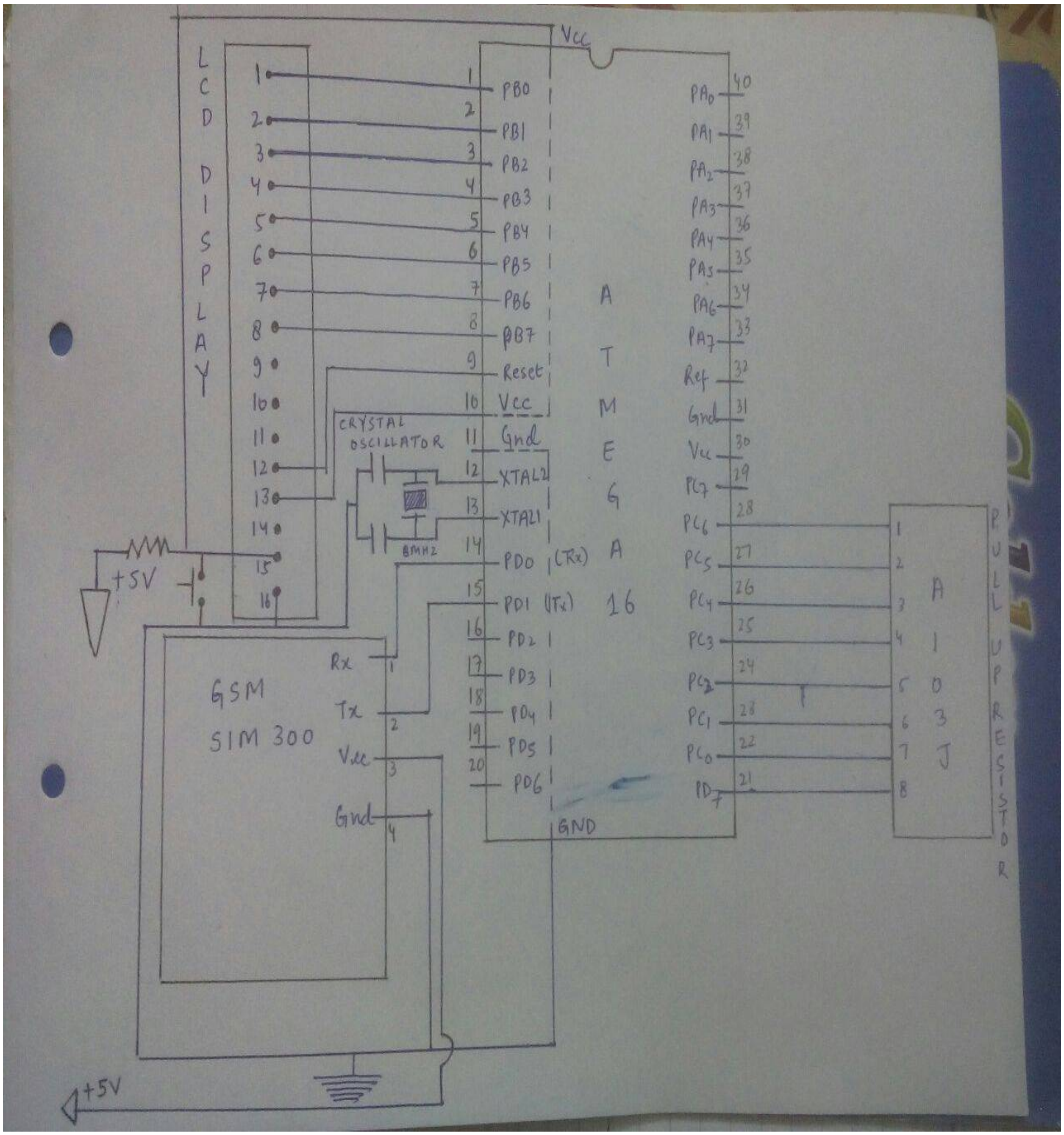


Figure No. 2.3: Circuit Diagram

CHAPTER 3

3.1) COST ANALYSIS OF COMPONENTS USED

Sr. no	Equipment	Quantity	Rate (in Rs.)
1	IC ATMEGA16 MC	1	250
2	GSM MODULE	1	1500
3	Adapter(12V)	1	130
4	Voltage Regulator 7805	1	20
5	2 line LCD display	1	180
6	Berg Strip	4	30
7	Crystal Oscillator(8 MHz)	1	50
8	Push Button Switch	3	60
9	LED	3	10
10	Resistors(1KΩ,10KΩ,47kΩ,100KΩ,330kΩ)	13	25
11	Capacitors(22pf,10µf)	4	20
12	Network Resistor A103J	1	10
13	DC Pin	1	25
	TOTAL		2510

Table no. 3.1: Cost Analysis

3.2 PROBLEM FACED

- First problem that was in making the circuit, it is difficult to solder the microcontroller.

- We have to take extra care while soldering 2 line LCD.

- During soldering, many of the connection become short circuit. So we desolder the connection and did soldering again.

- A leg of the crystal oscillator was broken during mounting. So it has to be replaced.

- LED`s get damaged when we switched ON the supply so we replace it by the new one.

3.3 TROUBLESHOOT

- Care should be taken while soldering. There should be no shorting of joints.
- Proper power supply should maintain.
- Project should be handled with care since IC are delicate.
- Component change and check again circuit.

CHAPTER 4

4.1 CONCULSION

In fact, we believe that an open process would result in more careful development, as more scientists, software engineers, political activists, and others who value their democracy would be paying attention to the quality of the software that is used for their elections. (Of course, open source would not solve all of the problems with electronic elections. It is still important to verify somehow that the binary program images running in the machine correspond to the source code and that the compilers used on the source code are non-malicious. However, open source is a good start.) Such open design processes have proven successful in projects ranging from very focused efforts, such as specifying the Advanced Encryption Standard (AES) [23], through very large and complex systems such as maintaining the Linux operating System. Australia is currently using an open source voting system10Alternatively, security models such as the voter-verified audit trail allow for electronic voting systems that produce a paper trail that can be seen and verified by a voter. In such a system, the correctness burden on the voting terminal's code is significantly less as voters can see and verify a physical object that describes their vote.

4.2 FUTURE SCOPE

- Number of candidates could be increased by using other microcontroller.
- It could be interfaced with printer to get the hard copy of the result almost instantly from the machine itself.
- It could also be interfaced with the personal computer and result could be stored in the central server and its backup could be taken on the other backend servers.
- Again, once the result is on the server it could be relayed on the network to various offices of the election conducting authority.
- Thus our project could make the result available any corner of the world in a matter of seconds

4.3 AREA OF APPLICATIONS

- Fast track voting which could be used in small scale elections, like resident welfare association, “panchayat” level election and other society level elections.
- It could also be used to conduct opinion polls during annual share holders meeting.
- It could also be used to conduct general assembly elections where number of candidates are less than or equal to eight in the current situation.
- It is used in various TV serials as for public opinion.

4.4 REFERENCES

- Steven F. Barrett , Daniel J. Pack (2008) , 'ATMEL AVR MICROCONTROLLER PRIMER' .

- Muhammad Ali Mazidi - ' AVR MICROCONTROLLER and EMBEDDED SYSTEMS: USING ASSEMBLY and C' .

- Richard H. Barnett, Sarah A. Cox, Larry D. O'Cull - 'Embedded C Programming and the Atmel AVR'.

- Chuck Kuhnel - 'AVR RISC Microcontroller Handbook'.

- Elliot Williams - 'AVR Programming: Learning to Write Software for Hardware'.

APPENDIX

CODING

SOFTWARE:-

```
'-----------------------------------------------------------
'               (c) 1997-2000 MCS Electronics
'-----------------------------------------------------------

'  Send and Receive SMS

$regfile = "m16def.dat"

$large

Config Lcd = 16 * 2

 'define xtal speed

$baud = 9600

$crystal = 8000000

'declare constants

Declare Sub Getline

Declare Sub Show

Declare Sub Send_sms1
```

```
Declare Sub Vote_erase

Declare Sub Choice

Dim Ct As Byte

Dim Bt As Byte

Dim N1 As Byte

Dim N2 As Byte

Dim N3 As Byte

 Ct = 0

Dim C1 As Byte

Dim C2 As Byte

Dim C3 As Byte

Dim C4 As Byte

Dim C5 As Byte

Dim Temp As Byte

Dim Gps As Byte

Dim Sty As String * 1
```

```
Dim Adres As Byte

Dim Value As Byte

Dim Num As String * 10

Dim Disp As String * 16

Dim Inmsg As String * 100

Enable Interrupts

Readeeprom C1 , 10

Readeeprom C2 , 11

Readeeprom C3 , 12

Readeeprom C4 , 13

Readeeprom C5 , 14

Readeeprom C1 , 10

Readeeprom C2 , 11

Readeeprom C3 , 12

Readeeprom C4 , 13

Readeeprom C5 , 14
```

```
Writeeeprom Ct , 10

Writeeeprom Ct , 11

Writeeeprom Ct , 12

Writeeeprom Ct , 13

Writeeeprom Ct , 14

Cursor Off
Cls

Lcd "WELCOME TO GSM"

Lowerline

Lcd "BASED VOTING"

Wait 2                                  ' give some time for gsm modem

Print "AT"

Wait 1

Print "AT+CREG?"

Wait 1

Print "ATE0"                            'Echo off
```

```
Wait 1

Print "AT+CNMI=1,1,0,0,0"                    'new message indication off

Wait 1

Print "AT+CMGF=1"                            'change to Text mode

Wait 1

Print "AT+CMGD=1,4"                          'Delete all sms

Wait 1


Cls

Lcd "YOU CAN VOTE FOR"

Lowerline

Lcd "A  B  C  D  X"

Waitms 200


Main:

Do
```

```
Print "AT+CMGR=1"                              'Read for message

Getline

If Inmsg <> "OK" Then                          'got new message

    Num = ","

    Gps = Instr(inmsg , Num)

    Gps = Gps + 5

    Num = Mid(inmsg , Gps , 10)                'extract the sender number

    Getline                           'get the message


    Cls

    Lcd "1 Msg Received"

    Lowerline

    Lcd "from:" ; Num

    Wait 1

    Cls

    Disp = Left(inmsg , 16)                    'Display first 16 characters
```

```
Lcd Disp

Lowerline

Disp = Mid(inmsg , 17 , 16)                    'Display Next 16 characters

Lcd Disp


If Num = "398876" Then

Gosub Choice

End If

If Num = "8948843901" And N1 = 0 Then

N1 = 1

Gosub Choice

Elseif Num = "8004629472" And N2 = 0 Then

 N2 = 1

Gosub Choice

Elseif Num = "9170408652" And N3 = 0 Then

 N3 = 1
```

```
      Gosub Choice

      Else

      Cls

      Lcd "INVALID VOTE"

      Waitms 400

      Inmsg = "SORRY YOU CANT VOTE"

       Gosub Send_sms1

      End If

      Print "AT+CMGD=1,4"                    'delete all sms

      Wait 1

      End If

  'End If

  Wait 1

Loop
```

```
End                                          'continious loop

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@

'       Read gsm modem

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@

Sub Getline

 Inmsg = ""

 Do

   Gps = Inkey()

   If Gps > 0 Then

      Select Case Gps

       Case 13 : If Inmsg <> "" Then Exit Do           ' if we have received something

 Case 10 : If Inmsg <> "" Then Exit Do           ' if we have received something

Case Else

  Inmsg = Inmsg + Chr(gps)                   ' build string
```

```
  End Select

   End If

 Loop

End Sub

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

'      choice

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Sub Choice:

If Inmsg = "*RESULT#" Then

Gosub Show

Waitms 300

Cls

Lcd "YOU CAN VOTE FOR"

Lowerline
```

```
Lcd "A  B  C  D  X"

Waitms 200

Goto Main

End If

If Inmsg = "*DEL#" Then

Gosub Vote_erase

End If

If Inmsg = "*VOTE A#" Then

C1 = C1 + 1

Writeeeprom C1 , 10

Waitms 200

Gosub Show

Waitms 100

Gosub Send_sms

Wait 1

End If
```

```
If Inmsg = "*VOTE B#" Then

C2 = C2 + 1

Writeeeprom C2 , 11

Waitms 200

Gosub Show

Waitms 200

Gosub Send_sms

Wait 1

End If

If Inmsg = "*VOTE C#" Then

C3 = C3 + 1

Writeeeprom C3 , 12

Waitms 200

Gosub Show

Waitms 200
```

```
Gosub Send_sms

Wait 1

Goto Main

End If

If Inmsg = "*VOTE D#" Then

C4 = C4 + 1

Writeeeprom C4 , 13

Waitms 200

Gosub Show

Waitms 200

Gosub Send_sms

Wait 1

End If

If Inmsg = "*VOTE D#" Then

C5 = C5 + 1

Writeeeprom C5 , 14
```

```
Waitms 200

Gosub Show

Waitms 200

Gosub Send_sms

Wait 1

End If

End Sub


'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@

'       SHOW

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@

Sub Show:

   Cls

 Lcd "A=" ; C1 ; " B=" ; C2 ; " C=" ; C3
```

```
 Lowerline

 Lcd "D= " ; C4 ; " X = " ; C5

 Wait 2


'Goto Main

End Sub

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@

'       SEND SMS

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@

Sub Send_sms

   If Num <> "" Then

        Wait 1

        Print "AT+CMGS=" ; Chr(34) ; "+91" ; Num ; Chr(34)       'send sms

        Wait 1
```

```
        Print "THANK YOU FOR YOUR VOTE" ; Chr(26)

        Wait 1

       Print "AT+CMGD=1,4"                    'delete all sms

       Wait 1

        Cls

        Lcd "YOU CAN VOTE FOR"

        Lowerline

        Lcd "A  B  C  D  X"

        Waitms 200

        Goto Main

  End If

End Sub

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@

'       SEND SMS

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@

Sub Send_sms1

   If Num <> "" Then

        Wait 1

        Print "AT+CMGS=" ; Chr(34) ; "+91" ; Num ; Chr(34)       'send sms

        Wait 1

        Print Inmsg ; Chr(26)

        Wait 1

  End If

  Print "AT+CMGD=1,4"                                    'delete all sms

       Wait 1

  Cls

       Lcd "YOU CAN VOTE FOR"

       Lowerline

       Lcd "A  B  C  D  X"
```

```
    Waitms 200

    Goto Main

End Sub

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@

'   DELETE RESULT

'@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@

Sub Vote_erase:


Bt = 0

Cls

Lcd "MEMMORY ERASE"

Waitms 200

Cls

Lcd "ERASING........."
```

```
Waitms 100

Cls

Lcd "ERASING........."

Waitms 100

Cls

Lcd "ERASING........."

Writeeeprom Bt , 10

Writeeeprom Bt , 11

Writeeeprom Bt , 12

Writeeeprom Bt , 13

Writeeeprom Bt , 14

Waitms 100

Cls

Lcd "SYSTEM ERASED"

Waitms 300

Gosub Show
```

```
Waitms 200

Cls

Lcd "YOU CAN VOTE FOR"

Lowerline

Lcd "A  B  C  D  X"

Waitms 200

Goto Main

End Sub
```