A

Major Project Stage-I Report

on

"WIRELESS SYSTEM FOR TRAFFIC SIGN IDENTIFICATION AND VIOLATION MONITORING"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

submitted by

NANCHARLA MANIKUMAR 18M61A04A7

Under the Guidance of

MS. P. NAGA LAXMI M. Tech

ASSISTANT PROFESSOR

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
SWARNA BHARATHI INSTITUTE OF SCIENCE & TECHNOLOGY, KHAMMAM
(Approved by AICTE, Govt. of TS& Affiliated to JNTUH, Hyderabad)

(2021-2022)

SWARNA BHARATHI INSTITUTE OF SCIENCE & TECHNOLOGY, KHAMMAM
(Approved by AICTE, Govt. of TS& Affiliated to JNTUH, Hyderabad)
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the major project stage-I entitled ***"WIRELESS SYSTEM FOR TRAFFIC SIGN IDENTIFICATION AND VIOLATION MONITORING"*** *is a bonafide record of work carried out by*

NANCHARLA MANIKUMAR 18M61A04A7

We hereby accord our approval of it as a major project carried out and presented in a manner required for its acceptance in partial fulfillment for the award of the degree of ***BACHELOR OF TECHNOLOGY*** *in* ***ELECTRONICS AND COMMUNICATION ENGINEERING*** of **Jawaharlal Nehru Technological University Hyderabad, Hyderabad** *during the academic year 2021-2022.*

UNDER THE GUIDANCE OF
MS. P. NAGA LAXMI M-Tech
ASSISTANT PROFESSOR

HEAD OF THE DEPARTMENT
Dr.K. AMIT BINDAJ M. Tech, Ph. D, MIEEE
PROFESSOR

External Examiner

Principal
Dr. G. RAJA KUMAR M. Tech, Ph.D.

ACKNOWLEDGEMENT

I take the opportunity to express my deep sense of gratitude to my internal guide, **MS. P. NAGA LAXMI, Professor,** Department of ECE, Swarna Bharathi Institute of Science & Technology, Khammam, for her excellent guidance, technical information, and support which helped us in successfully completing this major project.

I take the opportunity to express my heartfelt thanks to the Head of the Department **Dr. K. AMIT BINDAJ, Professor,** Department of Electronics & Communication Engineering, Swarna Bharathi Institute of Science & Technology, Khammam, for his excellent guidance, technical information, and support which helped us in successfully completing this major-Project.

I wish to express our profound thanks to **Dr. G. RAJA KUMAR, Principal**, Swarna Bharathi Institute of Science & Technology, Khammam, for providing the necessary facilities to make this project a success.

I wish to express my thanks to the **Management Members** of Swarna Bharathi Institute of Science & Technology, Khammam, for providing the necessary facilities to make this project a success

I thank all the members of the Teaching and Non-Teaching Staff, the Department of ECE, and all those who have helped us directly or indirectly in completing our project successfully.

DECLARATION

I hereby declare that the Major project stage-I entitled **"WIRELESS SYSTEM FOR TRAFFIC SIGN IDENTIFICATION AND VIOLATION MONITORING"** recorded in this project is based on our work carried out at the **"SWARNA BHARATHI INSTITUTE OF SCIENCE &TECHNOLOGY", Khammam** during the B. Tech course.

DATE:
PLACE: Khammam

Reported by,

NANCHARLA MANIKUMAR 18M61A04A7

ABSTRACT

In the field of Vehicular automation, the detection of traffic signs plays a vital role. Enormous growth has been faced in the field. Even though the technology warns the user about the detected traffic sign, there are chances that the user might violate them, and violating the traffic sign rules might lead to accidents where the life of people would be at stake. Hence, to avoid these problems this project has proposed a prototype in such a way that, it detects the traffic signs and acts accordingly denying the user's control in real-time if he tries to violate the rules in the traffic sign. The prototype contains a traffic system, which is controlled with an 8051 Microcontroller.

KEY WORD: LCD, SWITCH, RFID READER, RFID TAG, GSM, TRAFFIC LIGHT, PSU

INDEX

CONTENTS	PAGE-NO
1. INTRODUCTION	**1-10**
1.1 Introduction of the project	1-2
1.2 Block diagram of the project	3
1.3 The Microcontroller	4
1.3.1 Pin configuration	5-10
2. Embedded Systems	**11-16**
2.2 computer instruction set	15
2.2.1 Reduced Instruction Set computer (RSIC)	15
2.2.2 RSIC characteristics	16
2.2.3 Complex Instruction set computer (CSIC)	16
2.3 Memory Architecture	17
2.3.1 Harvard Architecture	18
2.3.2 Von-Neumann Architecture	19
3. Software	**20-36**
3.1 Keil software	20
3.1.1 Keil software tool steps	21
3.2 Proteus	29
3.2.1 Features	30
3.2.2 Starting new design	30-36

4.Hardware **37-47**

4.1. Liquid Cristal display 37

4.2. Description of 16x2 37-41

4.3.GSM 42

4.4. TDMA 43

4.5.GSM Modem 45-47

5. Result and Description **51-52**

7.Conclusion **53**

References 54

Source Code 55-62

LIST OF FIGURES PAGE-NO

Fig 1.1	Block Diagram of project	3
Fig 1.2	8051 PIN configuration	5
Fig 1.3	AT89C51 Block diagram	6
Fig 2.1	Block diagram of Microcontroller	12
Fig 2.2	Three basic elements of a microprocessor	14
Fig 2.3	Block diagram of microcontroller	14
Fig 2.4	Harvard Architecture	18
Fig 2.5	Schematic of the Von-Neumann Architecture	19
Fig 3.1	Uvision interface	22
Fig 3.2	New project	22
Fig 3.3	Saving file	23
Fig 3.4	Target1	23
Fig 3.5	Select AT89C52	24
Fig 3.6	Add file	24
Fig 3.7	Source Group1	25
Fig3.8	Text interface	25
Fig3.9	Compile window	26
Fig3.10	Proteus File menu	28
Fig3.11	Proteus default template select	29
Fig3.12	Proteus design sheet	29
Fig3.13	Component mode	30
Fig3.14	Pick from libraries	30
Fig3.15	Keywords Textbox	31
Fig3.16	Push Button Selection	31
Fig3.17	Component selection	32
Fig3.18	Component properties selection	32
Fig3.19	Component Properties Edit	33
Fig3.20	Simulation Run	34
Fig4.1	Schematic Diagram	36
Fig4.2	Pin diagram of 1*16 lines LCD	38
Fig4.3	GSM modem	47
Fig4.4	Establishing the connection between pc and GSM	48
Fig 5.1	program installation	51
Fig 5.2	compiling code	51
Fig 5.3	Before signal	52
Fig 5.4	After signal	52

LIST OF TABLES PAGE-NO

Table -1.1	Port pin	8
Table4.1	16*2 Alphanumeric LCD Module Specifications	37
Table4.2	Address location for a 1*16 LCD	38
Table4.3	Pin specifications	39
Table4.4	Commands	41

ACRONYMS

LCD	Liquid crystal display
GSM	Global system for mobile communication
RFID	Radiofrequency Identification
PSEN	Program Store Enable
EA	External Access enable
ASIC	Application-specific Integrated circuits
ALU	Arithmetic logic unit
CU	Control unit
CPU	Central process unit
RSIC	Reduced instruction set computer
CSIC	Complex instruction set computer
RAM	Random access memory
TDMA	Time division multiple access
BSC	Base station controller
BSS	Base station subsystem
NSS	Network station subsystem
HLR	Home location register
VLR	Visitor Location register
EIR	Equipment Identify register
AUC	Authentication Centre
PIN	Personal Identification code
PUK	Personal unlock code

CHAPTER 1

1. INTRODUCTION

Travel is an important part of today's fast-paced life as everyone has to move around for their day-to-day work. Road transport is the most commonly used mode of travel due to its ease, low cost, and availability to the common man. The ease of travel is affected by such factors as the quality of road, congestion, time is taken, accidents, speed, etc. The major threat is the increasing number of accidents on a daily basis.

An accident survey estimated that around 3,00,000 accidents occur on Indian roads every year. These accidents not only eat the lives of people but also adds to the economic loss of the country. It is reported that over 80,000 people are killed on Indian roads while the total economic loss owing to road accidents is estimated to be over Rs. 3,600 crores.

Lack of discipline and emotions of road users cause traffic congestions which might lead to traffic violations. Having a safe and free flow of traffic is crucial for the economic development of the country as we must ensure spending less on fuels and less time on the road. Traffic enforcement authorities can deal with the challenge of regulating the traffic and enforcing rules caused by the huge number of vehicles on the road and the indiscipline of the motorists by applying modern technology.

Flouting lane discipline is the single major factor in India that inhibits the safe and free flow of traffic. It is also a common sight in India to see one slow-moving vehicle blocking the way of hundreds of vehicles coming behind it. If keeping slow-moving vehicles to the left is enforced, one important cause of traffic jams is removed. The authorities are responsible for controlling the traffic violation and pollution and imposing fines on the violators. The single most important rule is to follow the traffic signals injunctions.

Violation of signals results in a number of accidents. The major problem is the manual tracking of every single vehicle that violates signals. This problem can be brought under control if the tracking can be automated along with calculating and updating the fines. In the existing systems, the tracking of signal violations is implemented using image processing techniques.

If a vehicle has violated the signal, the identified owner is imposed a suitable fine. However, there is a number of limitations to this system.

1. It requires a camera in every lane of the signal.
2. These cameras require high maintenance and are prone to damage in bad weather conditions.
3. Dirt on the number plate makes image processing difficult.
4. Various font types on the number plate cause lack of precision.
5. Objects far from the camera are captured with poor resolution.
6. There is a need for a larger number of technical persons.

On the other hand, RFID-based traffic violation detection systems use radiofrequency waves to identify vehicles that are endowed with unique identification numbers in the form of RFID tags. RFID (Radio Frequency Identification) is one of the upcoming technologies in the field of engineering and innovation. It has a number of applications in the market starting from vehicle identification at tolls to security systems at malls. These RFID-based systems consist of 3 main components, namely RFID Reader, RFID Tags, and RFID Database.

Tags can be classified into 2 types: passive tags and active tags. A passive tag contains no internal power source whereas an active tag contains its own power source which runs the microchip circuitry and also helps in broadcasting the signal to the readers. Similarly, there are two types of readers: λ Stationary Readers which are fixed at a specific location and able to read the tags within their range. λ Mobile Reader is a movable device. The frequency of low-frequency tags varies between 30 ~ 300 kHz that of high-frequency tags vary from 3 ~ 30 MHz and that of ultra-high frequency varies between 300 ~ 3000 MHz

In this work, an RFID reader uses radio waves to read tags and hence does not require to be in the line of sight. Every vehicle is endowed with an RFID tag, while RFID readers on tare the sides of the road. Road markings are done according to the required range. Any vehicle crossing the road marking when the signal is red will be detected from the RFID tag of that vehicle.

BLOCK DIAGRAM OF THE PROJECT

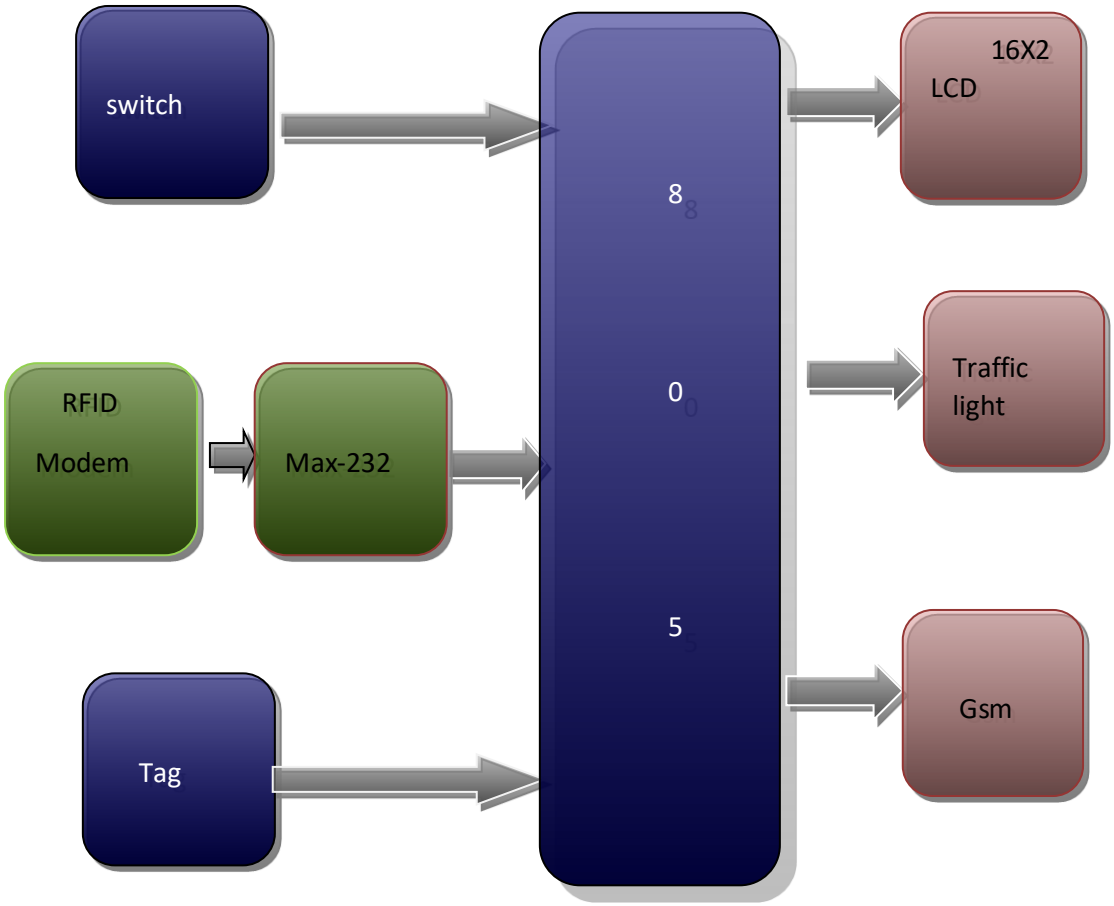

FIG-1.1 block diagram

1.3THE MICROCONTROLLER

A microcontroller is a general-purpose device, but that is meant to read data, perform limited calculations on that data and control its environment based on those calculations. The prime use of a microcontroller is to control the operation of a machine using a fixed program that is stored in ROM and that does not change over the lifetime of the system.

The microcontroller design uses a much more limited set of single- and double-byte instructions that are used to move data and code from internal memory to the ALU. The microcontroller is concerned with getting data from and to its own pins; the architecture and instruction set is optimized to handle data in a bit and byte size.

The AT89C51 is a low-power, high-performance CMOS 8-bit microcontroller with 4k bytes of Flash programmable and erasable read-only memory (EROM). The device is manufactured using Atmel's high-density non-volatile memory technology and is functionally compatible with the industry-standard 80C51 microcontroller instruction set and pinout. By combining a versatile 8-bit CPU with Flash on a monolithic chip, Atmel's AT89c51 is a powerful microcomputer, which provides high flexibility and a cost-effective solution to many embedded control applications.

AT89C51 MICROCONTROLLER

FEATURES

- 80C51 based architecture
- 4-Kbytes of on-chip Reprogrammable Flash Memory
  - Boolean processor
  - Four 8-bit I/O ports, 32 I/O lines
  - Memory addressing capability
  - 64K ROM and 64K RAM
  - Power save modes:
  - Idle and power-down
  - Six interrupt sources
  - Most instructions execute in 0.3 us
  - CMOS and TTL compatible

  - Maximum speed: 40 MHz @ Vcc = 5V
  - Industrial temperature available
  - Packages available:

– 40-pin DIP

– 44-pin PLCC

– 44-pin PQFP

PIN CONFIGURATION

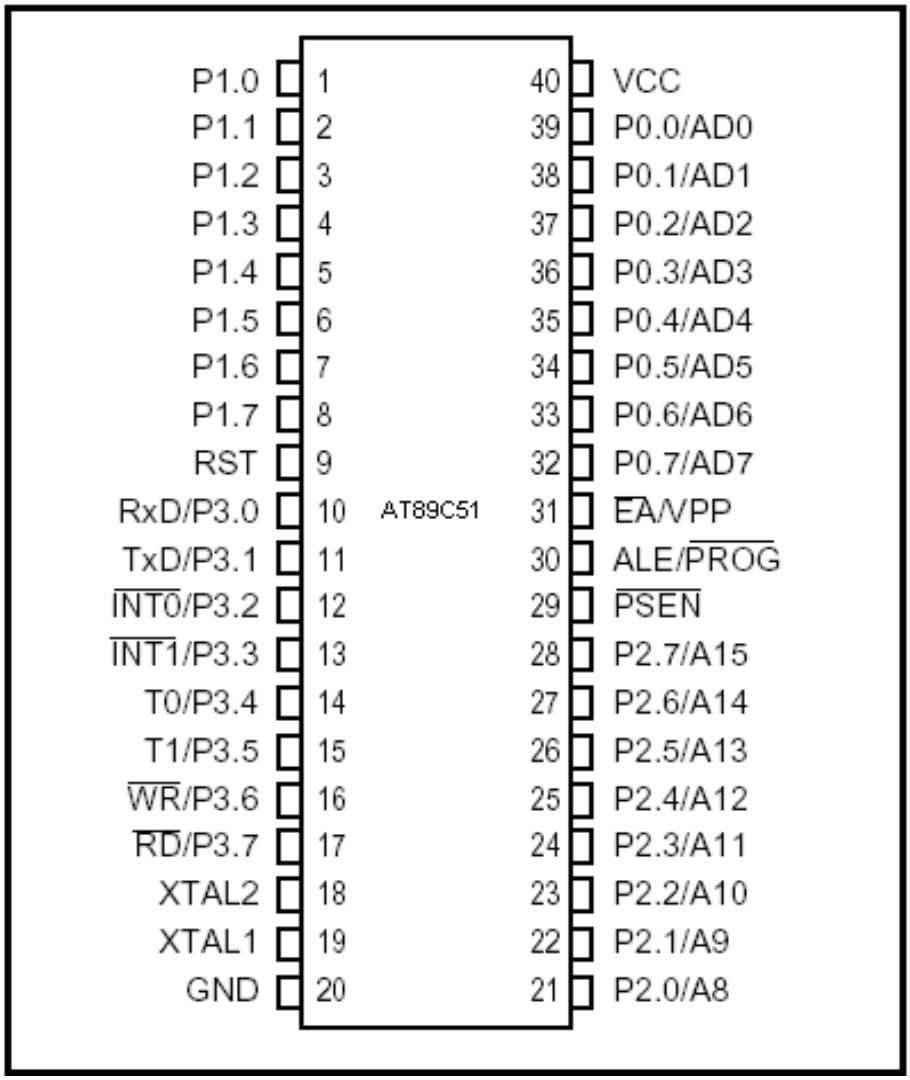

FIG-1.2 8051 pin configurations

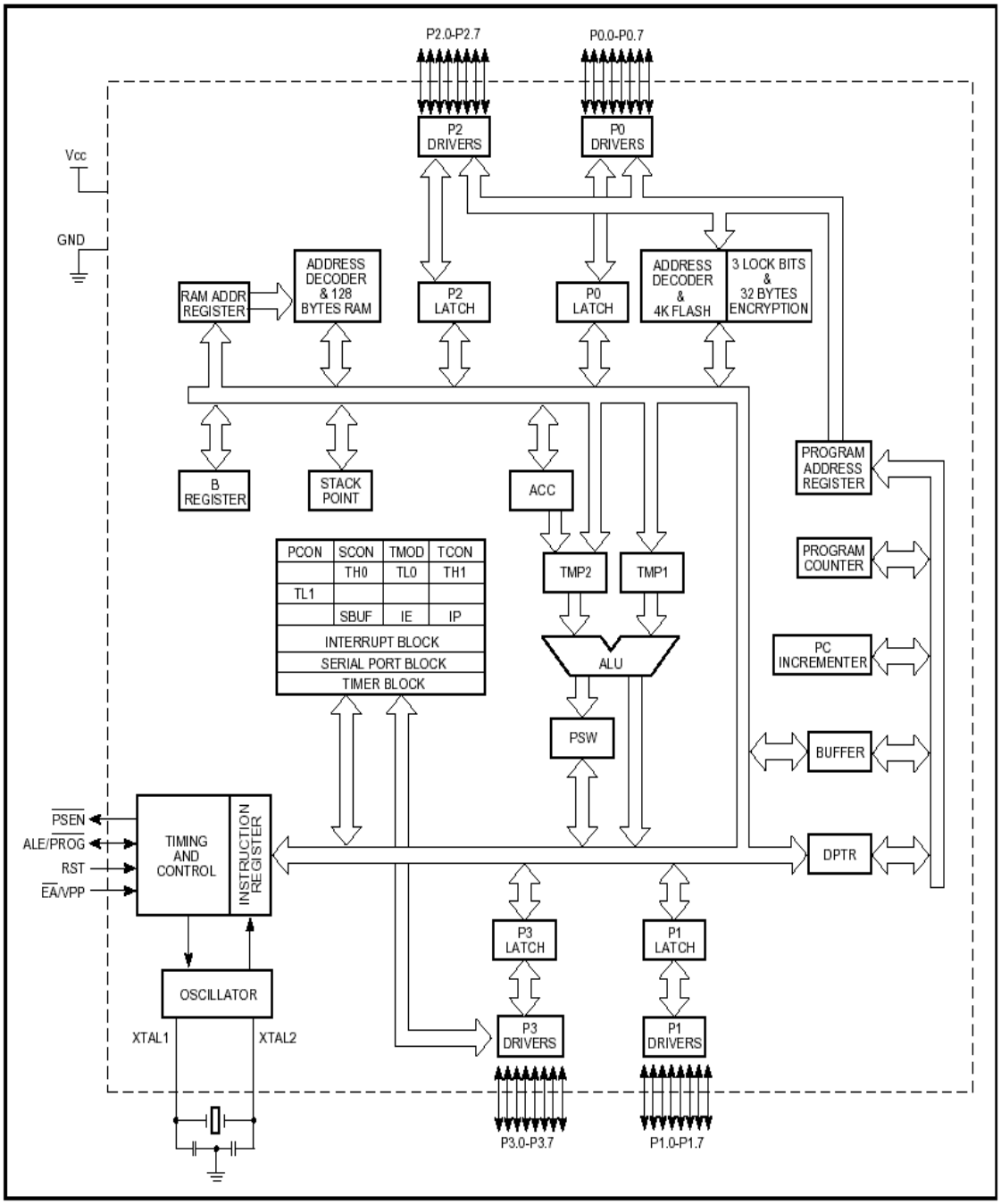

FIG-1.3 AT89C51 BLOCK DIAGRAM

PIN DESCRIPTION:

VCC

Supply voltage

GND

Ground

Port 0

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

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

Port 1

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

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The port 2 output buffers can sink/source four TTL inputs. When 1s are pulled high by the internal pull-ups can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during access to DPTR. In this application, Port 2 uses strong internal pull-ups when emitting 1s.

During accesses to external data memory that use 8-bit data address (MOVX@R1), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the

high-order address bits and some control signals during Flash programming and verification.

Port 3

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

Port 3 also receives some control signals for Flash Programming and verification.

Port pin	Alternate Functions
P3.0	RXD(serial input port)
P3.1	TXD(serial input port)
P3.2	INT0(external interrupt 0)
P3.3	INT1(external interrupt 1)
P3.4	T0(timer 0 external input)
P3.5	T1(timer 1 external input)
P3.6	WR(external data memory write strobe)
P3.7	RD(external data memory read strobe)

TABLE-1.1 PORT PIN

RST

Rest input A on this pin for two machine cycles while the oscillator is running reset the device.

ALE/PROG:

Address Latch Enable is an output pulse for latching the low byte of the address during an access to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE has emitted at a constant rate of 1/16 the oscillator frequency and may be used for external timing or clocking purpose. Note, however, that one ALE pulse is skipped during each access to external Data memory.

PSEN

Program Store Enable is the read strobe to external program memory when the AT89c51 is executing code from external program memory PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA /VPP

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

XTAL1

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

XTAL 2

Output from the inverting oscillator amplifier.

OPERATING DESCRIPTION

The detail description of the AT89C51 included in this description is:

• Memory Map and Registers

• Timer/Counters

MEMORY MAP AND REGISTERS

Memory

The AT89C51 has separate address spaces for program and data memory. The program and data memory can be up to 64K bytes long. The lower 4K program memory can reside on-chip. The AT89C51 has 128 bytes of on-chip RAM.

The lower 128 bytes can be accessed either by direct addressing or by indirect addressing. The lower 128 bytes of RAM can be divided into 3 segments as listed below

1. **Register Banks 0-3:** locations 00H through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers R0-R7. Reset initializes the stack point to location 07H, and is incremented once to start from 08H, which is the first register of the second register bank.

2. **Bit Addressable Area:** 16 bytes have been assigned for this segment 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH). Each of the 16 bytes in this segment can also be addressed as a byte.

3. **Scratch Pad Area:** 30H-7FH are available to the user as data RAM. However, if the data pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

CHAPTER 2
EMBEDDED SYSTEMS

2.1 Embedded System

An embedded system is a system which is going to do a predefined specified task is the embedded system and is even defined as combination of both software and hardware. A general-purpose definition of embedded systems is that they are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system. At the other extreme a general-purpose computer may be used to control the operation of a large complex processing plant, and its presence will be obvious.

All embedded systems are including computers or microprocessors. Some of these computers are however very simple systems as compared with a personal computer.

The very simplest embedded systems are capable of performing only a single function or set of functions to meet a single predetermined purpose. In more complex systems an application program that enables the embedded system to be used for a particular purpose in a specific application determines the functioning of the embedded system. The ability to have programs means that the same embedded system can be used for a variety of different purposes. In some cases a microprocessor may be designed in such a way that application software for a particular purpose can be added to the basic software in a second process, after which it is not possible to make further changes. The applications software on such processors is sometimes referred to as firmware

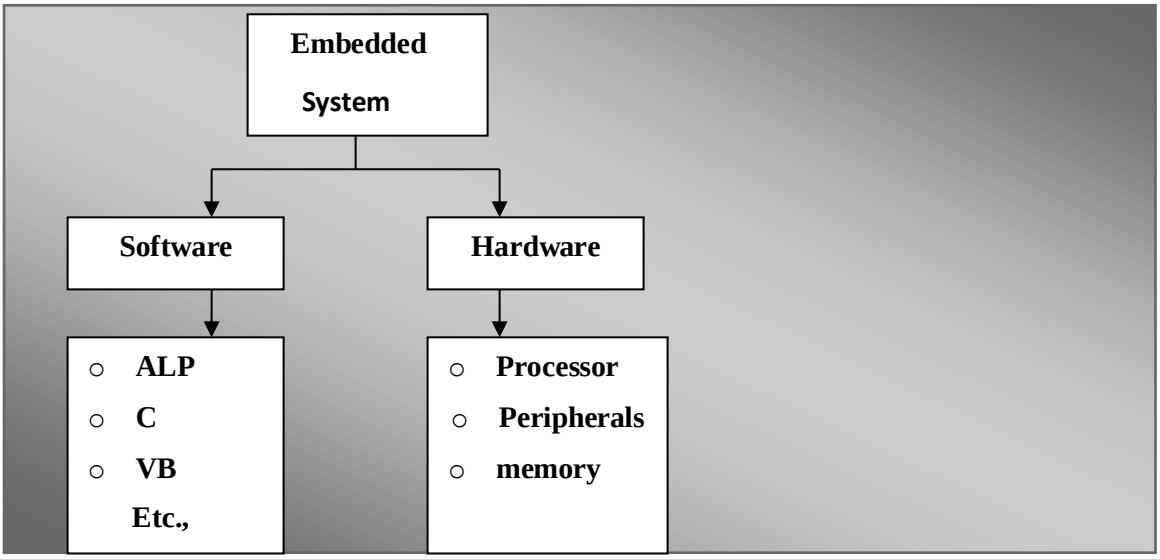

Figure 2.1 Block diagram of Embedded System

Software deals with the languages like ALP, C, VB etc., and Hardware deals with Processors, Peripherals, and Memory.

Memory: It is used to store data or addresses.

Peripherals: These are the external devices connected

Processor: It is an IC that is used to perform some task

Applications of embedded systems

- Manufacturing and process control
- Construction industry
- Transport
- Buildings and premises
- Domestic service
- Communications
- Office systems and mobile equipment
- Banking, finance, and commercial
- Medical diagnostics, monitoring, and life support
- Testing, monitoring, and diagnostic systems

Processors are classified into four types:

- Micro Processor (µp)
- Microcontroller (µc)
- Digital Signal Processor (DSP)
- Application-Specific Integrated Circuits (ASIC)

Micro Processor (µp):

A silicon chip that contains a CPU. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor.

Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Three basic characteristics differentiate microprocessors:

- **Instruction set**: The set of instructions that the microprocessor can execute.
- **Bandwidth:** The number of bits processed in a single instruction.
- **Clock speed:** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In both cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 50MHz is more powerful than a 16-bit microprocessor that runs at 25MHz. In addition to bandwidth and clock speed, microprocessors are classified as being either RISC (reduced instruction set computer) or CISC (complex instruction set computer).

A microprocessor has three basic elements, as shown above. The ALU performs all arithmetic computations, such as addition, subtraction and logic operations (AND, OR, etc). It is controlled by the Control Unit and receives its data from the Register Array.

The Register Array is a set of registers used for storing data. These registers can be accessed by the ALU very quickly. Some registers have specific functions - we will deal with these later. The Control Unit controls the entire process.

It provides the timing and a control signal for getting data into and out of the registers and the ALU and it synchronizes the execution of instructions (we will deal with instruction execution at a later date).

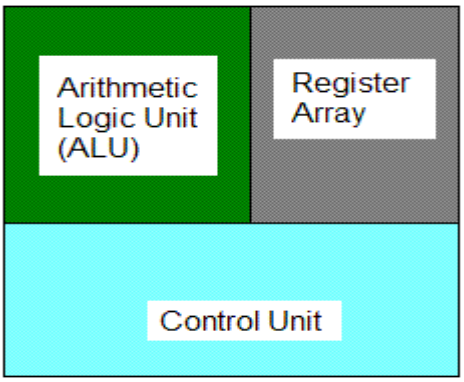

FIG-2.2 Three Basic Elements of a Microprocessor

Micro Controller (µc):

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

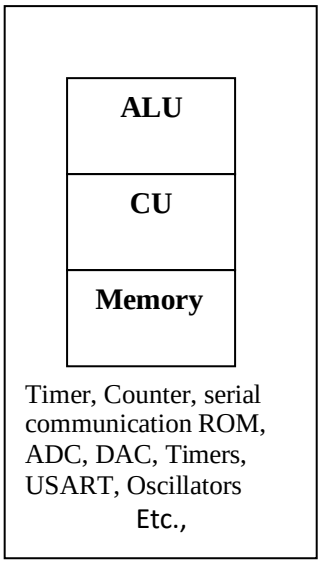

Figure 2.3 Block Diagram of Micro Controller (µc)

Digital Signal Processors (DSPs):

Digital Signal Processors is one which performs scientific and mathematical operation. Digital Signal Processor chips - specialized microprocessors with architectures designed specifically for the types of operations required in digital signal processing.

Like a general-purpose microprocessor, a DSP is a programmable device, with its own native instruction code. DSP chips are capable of carrying out millions of floating point operations per second, and like their better-known general-purpose cousins, faster and more powerful versions are continually being introduced. DSPs can also be embedded within complex "system-on-chip" devices, often containing both analog and digital circuitry.

Application Specific Integrated Circuit (ASIC)

ASIC is a combination of digital and analog circuits packed into an IC to achieve the desired control/computation function

ASIC typically contains

- CPU cores for computation and control
- Peripherals to control timing critical functions
- Memories to store data and program
-
- Analog circuits to provide clocks and interface to the real world which is analogy in nature
- I/Os to connect to external components like LEDs, memories, monitors etc.

2.2 Computer Instruction Set

There are two different types of computer instruction set there are:

1. RISC (Reduced Instruction Set Computer) and
2. CISC (Complex Instruction Set computer)

2.2.1 Reduced Instruction Set Computer (RISC)

A RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instruction so that it can operate at a higher speed (perform more million instructions per second, or millions of instructions

per second). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

Besides performance improvement, some advantages of RISC and related design improvements are:

- A new microprocessor can be developed and tested more quickly if one of its aims is to be less complicated.
- Operating system and application programmers who use the microprocessor's instructions will find it easier to develop code with a smaller instruction set.
- The simplicity of RISC allows more freedom to choose how to use the space on a microprocessor.

Higher-level language compilers produce more efficient code than formerly because they have always tended to use the smaller set of instructions to be found in a RISC computer.

2.2.2 RISC characteristics

- **Simple instruction set:**

In a RISC machine, the instruction set contains simple, basic instructions, from which more complex instructions can be composed.

- **Same length instructions.**

Each instruction is the same length, so that it may be fetched in a single operation.

- **1machine-cycleinstructions.**

Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. This pipelining is a key technique used to speed up RISC machines.

2.2.3 Complex Instruction Set Computer (CISC)

CISC, which stands for **Complex Instruction Set Computer,** is a philosophy for designing chips that are easy to program and which make efficient use of memory. Each instruction in a CISC instruction set might perform a series of operations inside the processor. This reduces the number of instructions required to implement a given program, and allows the programmer to learn a small but flexible set of instructions.

The advantages of CISC

At the time of their initial development, CISC machines used available technologies to optimize computer performance.

- Microprogramming is as easy as assembly language to implement, and much less expensive than hardwiring a control unit.
- The ease of micro-coding new instructions allowed designers to make CISC machines upwardly compatible: a new computer could run the same programs as earlier computers because the new computer would contain a superset of the instructions of the earlier computers.
- As each instruction became more capable, fewer instructions could be used to implement a given task. This made more efficient use of the relatively slow main memory.

The disadvantages of CISC

Still, designers soon realized that the CISC philosophy had its own problems, including:

- Earlier generations of a processor family generally were contained as a subset in every new version --- so instruction set & chip hardware become more complex with each generation of computers.
- So that as many instructions as possible could be stored in memory with the least possible wasted space, individual instructions could be of almost any length---this means that different instructions will take different amounts of clock time to execute, slowing down the overall performance of the machine.
- Many specialized instructions aren't used frequently enough to justify their existence --- approximately 20% of the available instructions are used in a typical program.
- CISC instructions typically set the condition codes as a side effect of the instruction. Not only does setting the condition codes take time, but programmers have to remember to examine the condition code bits before a subsequent instruction changes them.

2.3 Memory Architecture

There two different type's memory architectures there are:

- Harvard Architecture
- Von-Neumann Architecture

2.3.1 Harvard Architecture

Computers have separate memory areas for program instructions and data. There are two or more internal data buses, which allow simultaneous access to both instructions and data. The CPU fetches program instructions on the program memory bus.

The **Harvard architecture** is a computer architecture with physically separate storage and signal pathways for instructions and data. The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape (24 bits wide) and data in electro-mechanical counters. These early machines had limited data storage, entirely contained within the central processing unit, and provided no access to the instruction storage as data. Programs needed to be loaded by an operator, the processor could not boot itself.

Figure 2.4 Harvard Architecture

Modern uses of the Harvard architecture:

The principal advantage of the pure Harvard architecture - simultaneous access to more than one memory system - has been reduced by modified Harvard processors using modern CPU cache systems. Relatively pure Harvard architecture machines are used mostly in applications where tradeoffs, such as the cost and power savings from omitting caches, outweigh the programming penalties from having distinct code and data address spaces.

- concern to speed of execution. As a result, some DSPs have multiple data memories in distinct address spaces to facilitate SIMD and VLIW processing.
- Texas Instruments TMS320 C55x processors, as one example, have multiple parallel data busses (two write, three read) and one instruction bus.

- Microcontrollers are characterized by having small amounts of program (flash memory) and data (SRAM) memory, with no cache, and take advantage of the Harvard architecture to speed processing by concurrent instruction and data access.
- The separate storage means the program and data memories can have different bit depths, for example using 16-bit wide instructions and 8-bit wide data. They also mean that instruction pre-fetch can be performed in parallel with other activities. Examples include, the AVR by Atmel Corp, the PIC by Microchip Technology, Inc. and the ARM Cortex-M3 processor (not all ARM chips have Harvard architecture).

Even in these cases, it is common to have special instructions to access program memory as data for read-only tables, or for reprogramming.

2.3.2 Von-Neumann Architecture

A computer has a single, common memory space in which both program instructions and data are stored. There is a single internal data bus that fetches both instructions and data. The **von Neumann architecture** is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data. It is named after the mathematician and early computer scientist John von Neumann. Such computers implement a universal Turing machine and have a sequential architecture.

A **stored-program** digital computer is one that keeps its programmed instructions, as well as its data, in read-write, random-access memory (RAM). Stored-program computers were advancement over the program-controlled computers of the 1940s, such as the Colossus and the ENIAC, which were programmed by setting switches and inserting patch leads to route data and to control signals between various functional units. In the vast majority of modern computers, the same memory is used for both data and program instructions. The mechanisms for transferring the data and instructions between the CPU and memory are, however, considerably more complex than the original von Neumann architecture.

The terms "von Neumann architecture" and "stored-program computer" are generally used interchangeably, and that usage is followed in this article.

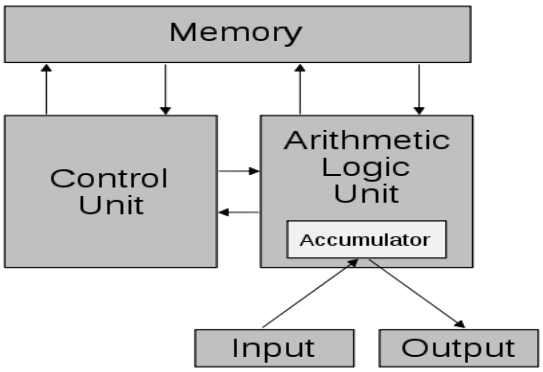

Figure 2.5 Schematic of the Von-Neumann Architecture.

Basic Difference between Harvard and Von-Neumann Architecture

- The primary difference between Harvard architecture and the Von Neumann architecture is in the Von Neumann architecture data and programs are stored in the same memory and managed by the same information handling system.
- Whereas the Harvard architecture stores data and programs in separate memory devices and they are handled by different subsystems.
- In a computer using the Von-Neumann architecture without cache; the central processing unit (CPU) can either be reading and instruction or writing/reading data to/from the memory. Both of these operations cannot occur simultaneously as the data and instructions use the same system bus.
- In a computer using the Harvard architecture the CPU can both read an instruction and access data memory at the same time without cache. This means that a computer with Harvard architecture can potentially be faster for a given circuit complexity because data access and instruction fetches do not contend for use of a single memory pathway.
- Today, the vast majority of computers are designed and built using the Von Neumann architecture template primarily because of the dynamic capabilities and efficiencies gained in designing, implementing, operating one memory system as opposed to two. Von Neumann architecture may be somewhat slower than the contrasting Harvard Architecture for certain specific tasks, but it is much more flexible and allows for many concepts unavailable to Harvard architecture such as self-programming, word processing and so on.

CHAPTER 3
SOFTWARE

3.1. Keil Software

Installing the Keil software on a Windows PC

- Insert the CD-ROM in your computer's CD drive
- On most computers, the CD will "auto run", and you will see the Keil installation menu. If the menu does not appear, manually double click on the Setup icon, in the root directory: you will then see the Keil menu.
- On the Keil menu, please select "Install Evaluation Software". (You will not require a license number to install this software).
- Follow the installation instructions as they appear.

Loading the Projects

The example projects for this book are NOT loaded automatically when you install the Keil compiler.
These files are stored on the CD in a directory "/Pont". The files are arranged by chapter: for example, the project discussed in Chapter 3 is in the directory "/Pont/Ch03_00-Hello".

Rather than using the projects on the CD (where changes cannot be saved), please copy the files from CD onto an appropriate directory on your hard disk.

Note: you will need to change the file properties after copying: file transferred from the CD will be 'read only'.

Configuring the Simulator

Open the Keil μVision2

go to Project – Open Project and browse for Hello in Ch03_00 in Pont and open it.

3.1.1. KEIL SOFTWARE TOOL STEPS

1. Click on the Keil uVision Icon on DeskTop
2. The following fig will appear

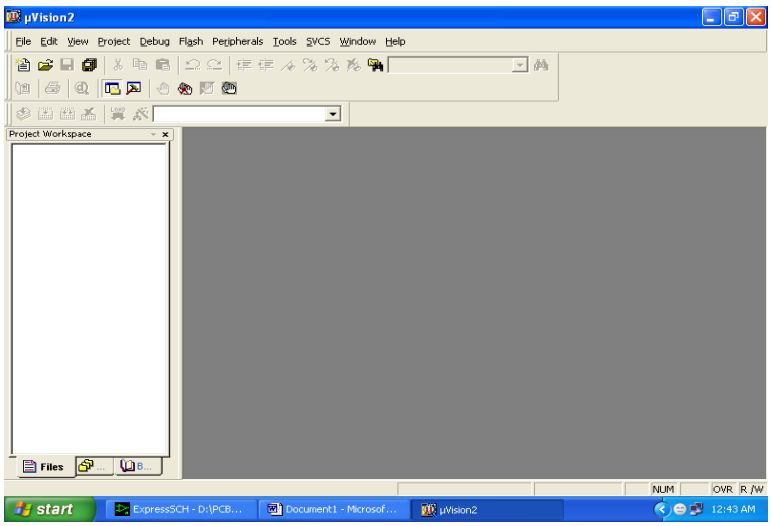

FIG-3.1 uVision interface

3. Click on the Project menu from the title bar
4. Then Click on New Project

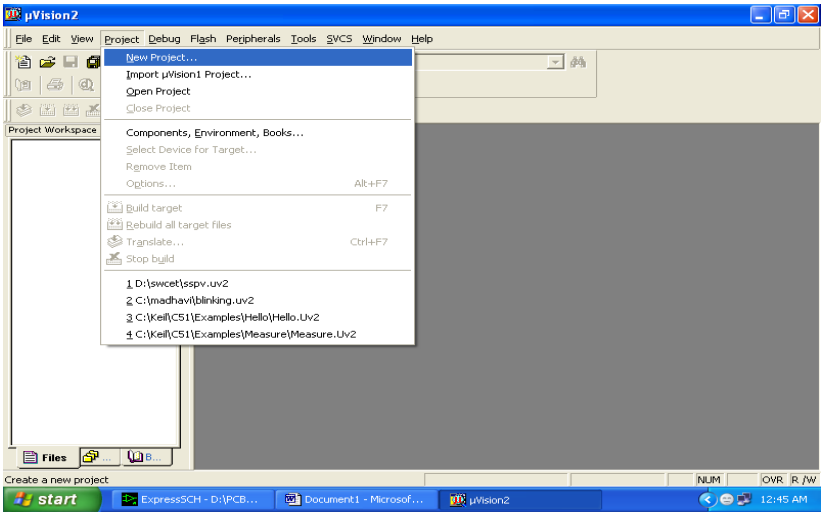

FIG-3.2 NEW PROJECT

5. Save the Project by typing suitable project name with no extension in u r own folder sited in either C:\ or D:\

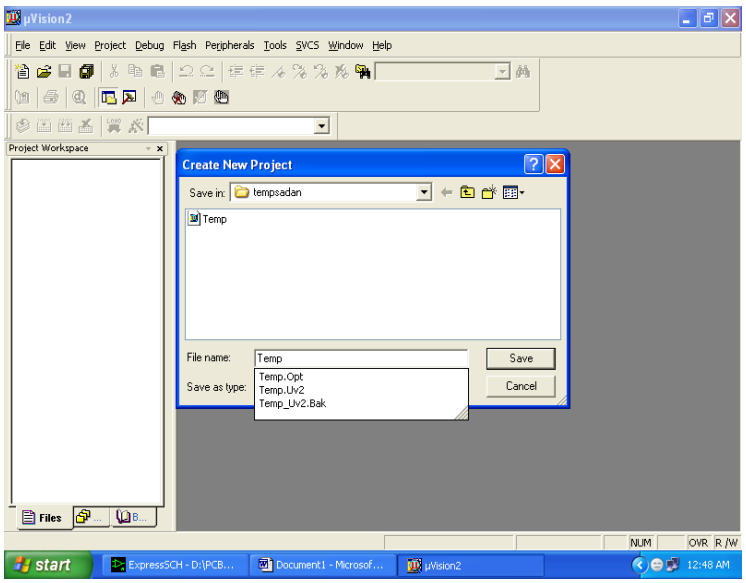

FIG-3.3 SAVING FILE

6. Then Click on Save button above.

7. Select the component for u r project. i.e. Atmel……

8. Click on the + Symbol beside of Atmel

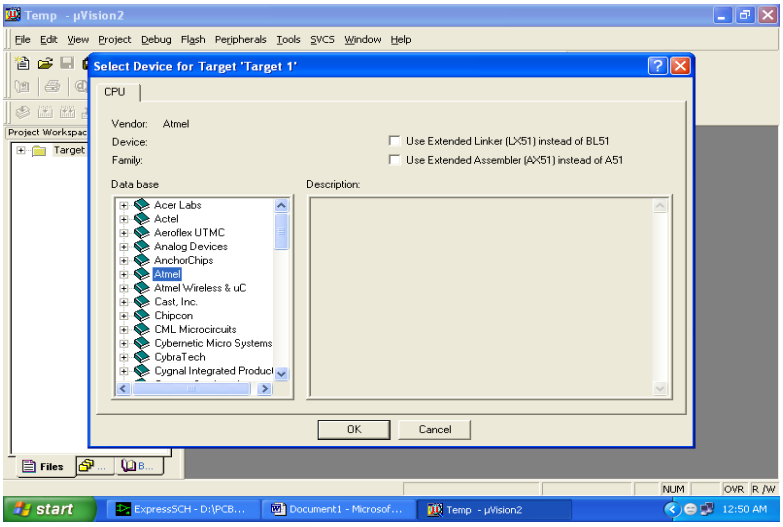

FIG-3.4 TARGET1

9. Select AT89C52 as shown below

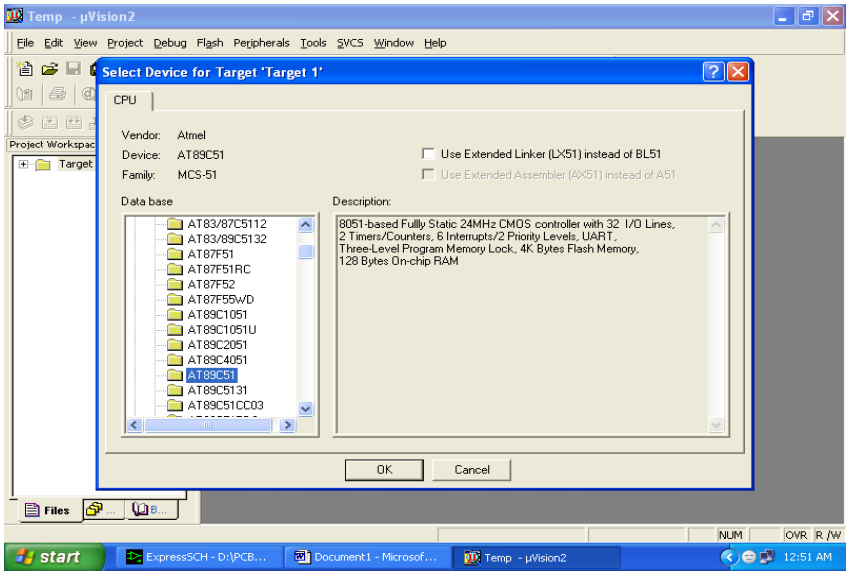

FIG-3.5 SELECT AT89C52

10. Then Click on “OK”

11. The Following fig will appear

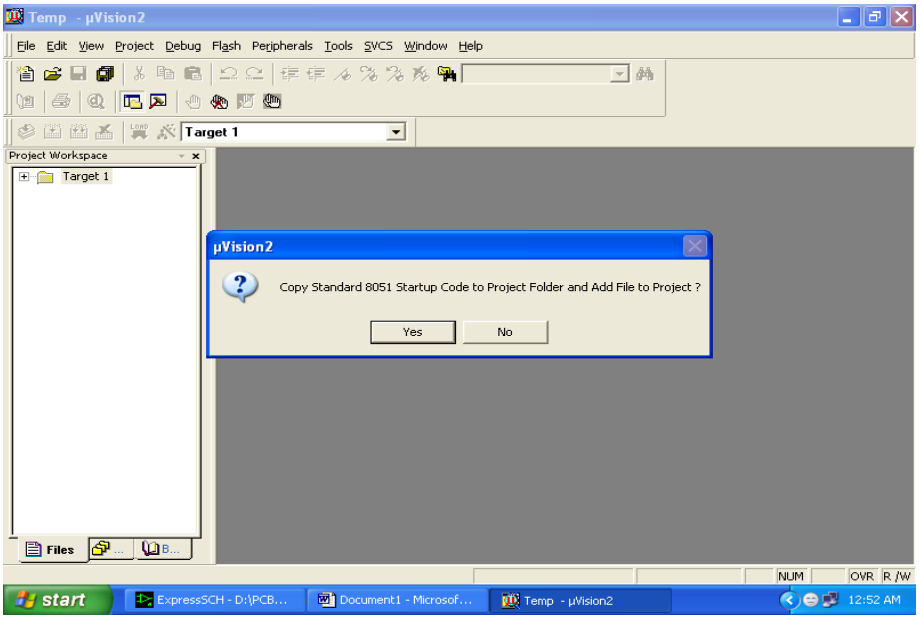

FIG-3.6-ADD FILE

12. Then Click either YES or NO……mostly “NO”

13. Now your project is ready to USE

14. Now double click on the Target1, you would get another option "Source group 1" as shown in next page.

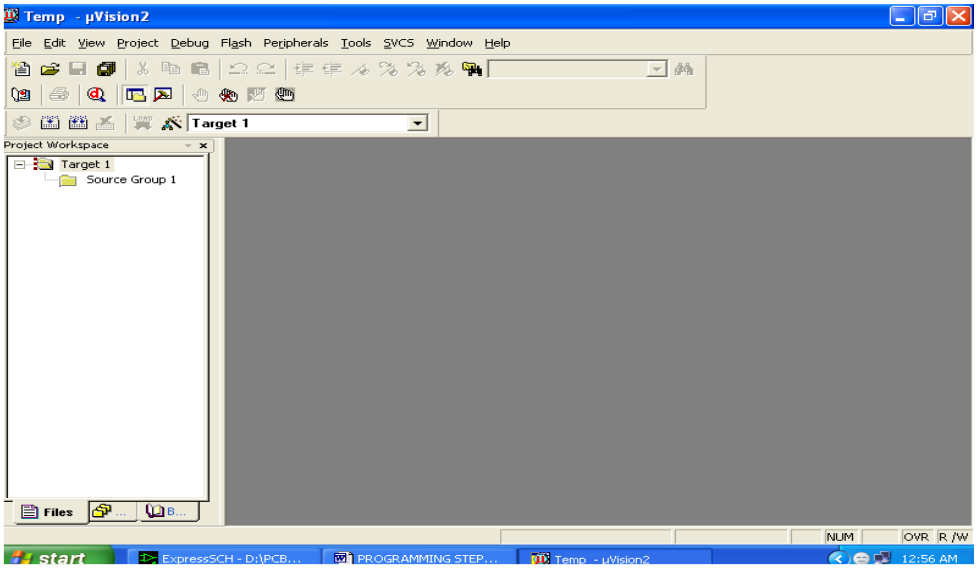

FIG-3.7-SOURCE GROUP1

15. Click on the file option from menu bar and select "new"

16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.

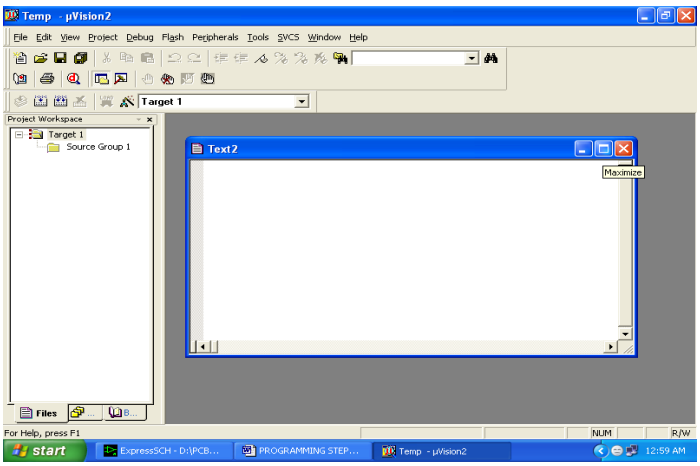

FIG3.8-TEXT INTERFACE

17. Now start writing program in either in "C" or "ASM"

18. For a program written in Assembly, then save it with extension ". asm" and for "C" based program save it with extension " .C"

19. Now right click on Source group 1 and click on "**Add files to Group Source**"

20. Now you will get another window, on which by default "C" files will appear.

21. Now select as per your file extension given while saving the file

22. Click only one time on option "**ADD**"

23. Now Press function key F7 to compile. Any error will appear if so happen.

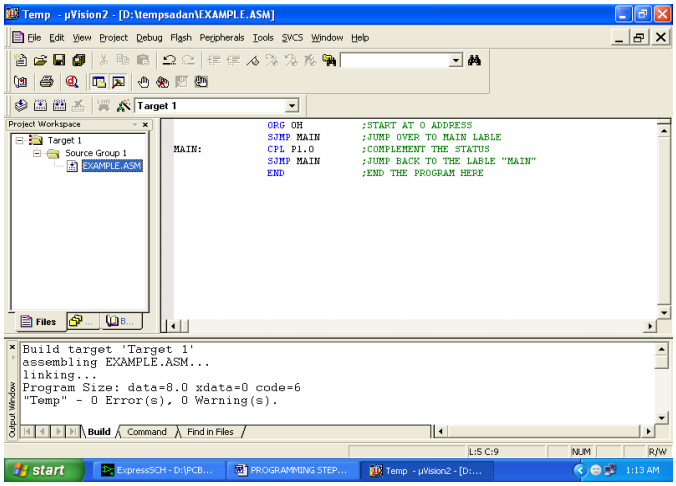

FIG-3.9 COMPILE WINDOW

24. If the file contains no error, then press Control+F5 simultaneously.

25. The new window is as follows

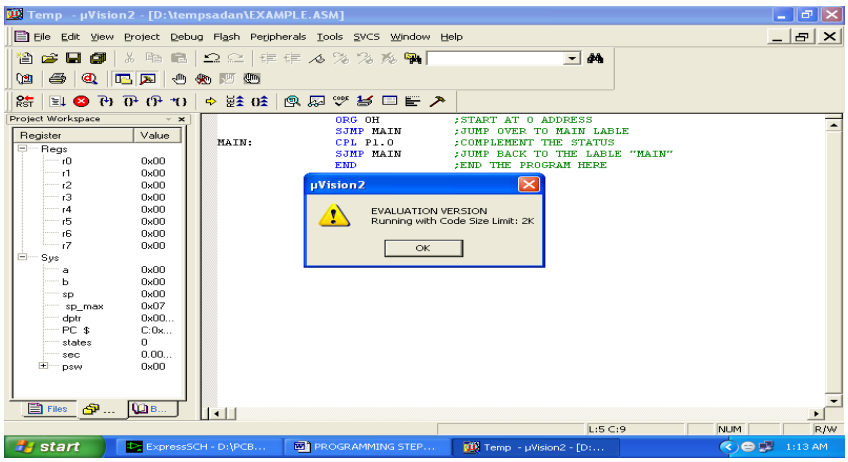

FIG-3.9.1.RUN WITH CODE

26. Then Click "OK"

27. Now Click on the Peripherals from menu bar, and check your required port as shown in fig below

28. Drag the port a side and click in the program file.

29. Now keep Pressing function key "F11" slowly and observe.

30. You are running your program successfully

3.2. Proteus

Proteus is a simulation and design software tool developed by Lab centre Electronics for Electrical and Electronic circuit design. It also possesses 2D CAD drawing feature. It deserves to bear the tagline "From concept to completion".

It is a software suite containing schematic, simulation as well as PCB designing.

ISIS is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.

ARES is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components

The designer can also develop 2D drawings for the product.

3.2.1 Features

ISIS has wide range of components in its library. It has sources, signal generators, measurement and analysis tools like oscilloscope, voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, switches, displays, loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

It is embedded with the foot prints of different category of components like ICs, transistors, headers, connectors and other discrete components. It offers Auto routing and manual routing options to the PCB Designer. The schematic drawn in the ISIS can be directly transferred ARES.

3.2.2 Starting New Design

Step 1: Open ISIS software and select New design in File menu

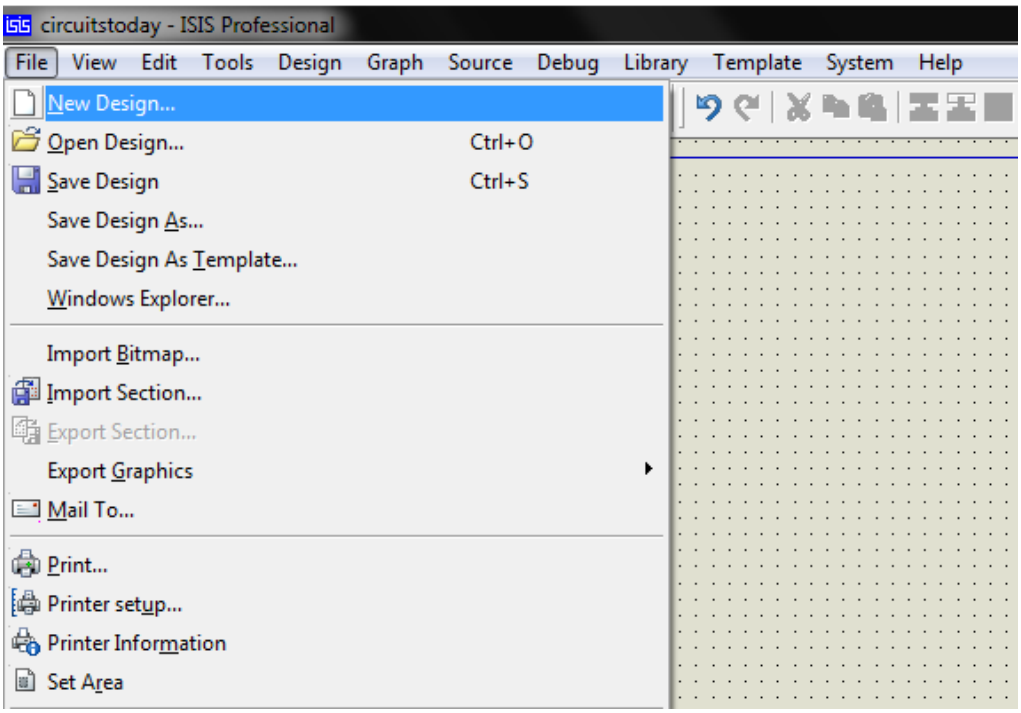

Figure 3.10 Proteus File Menu

Step 2: A dialogue box appears to save the current design. However, we are creating a new design file so you can click Yes or No depending on the content of the present file. Then a Pop-Up appears asking to select the template. It is similar to selecting the paper size while printing.

For now, select default or according to the layout size of the circuit.

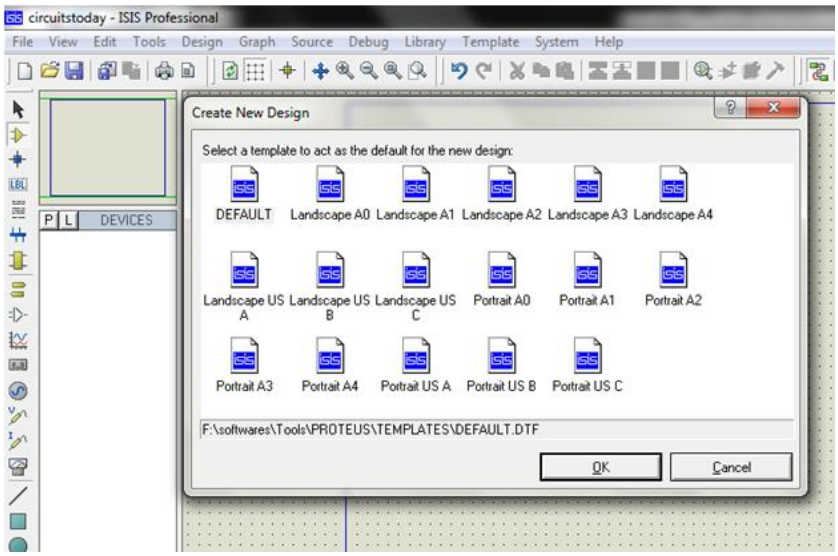

Figure 3.11 Proteus Default Template Select

Step 3: An untitled design sheet will be opened, save it according to your wish, it is better to create a new folder for every layout as it generates other files supporting your design. However, it is not mandatory.

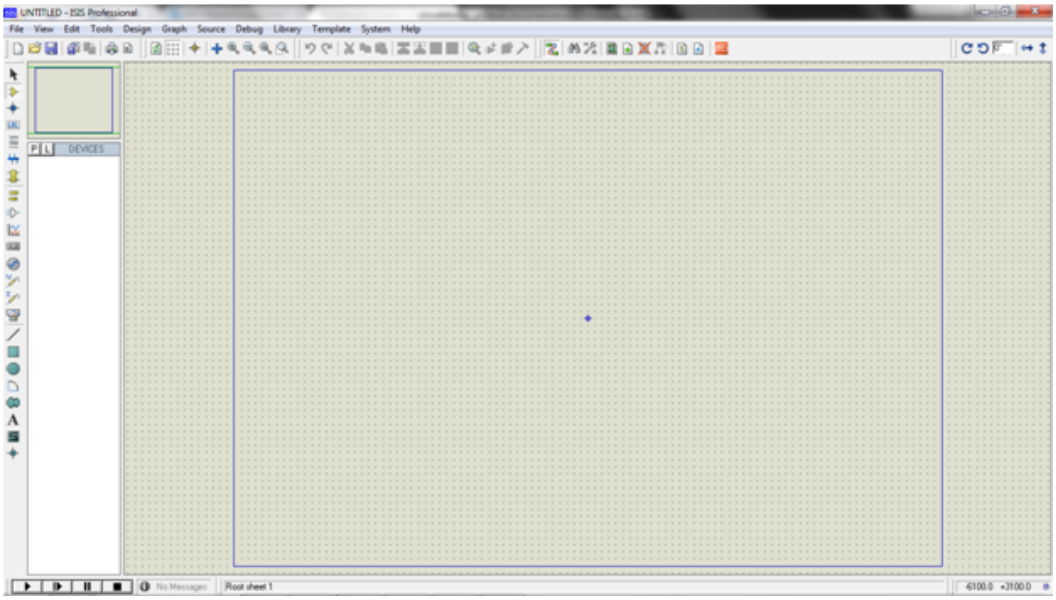

Figure 3.12 Proteus Design Sheet

Step 4: To Select components, Click on the component mode button.

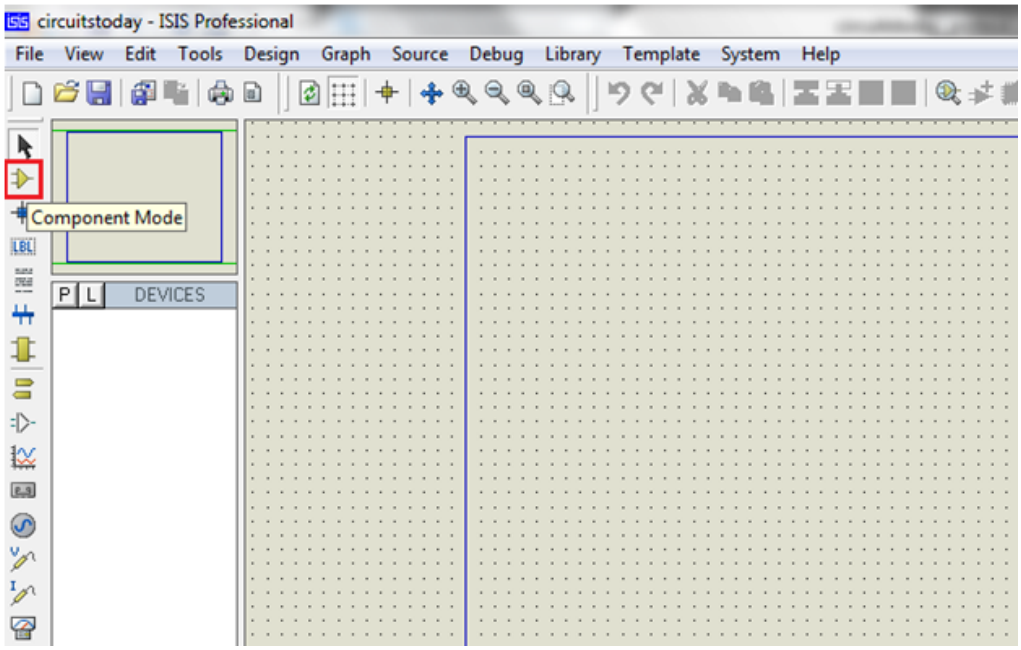

Figure 3.13 Component Mode

Step 5: Click on Pick from Libraries. It shows the categories of components available and a search option to enter the part name.

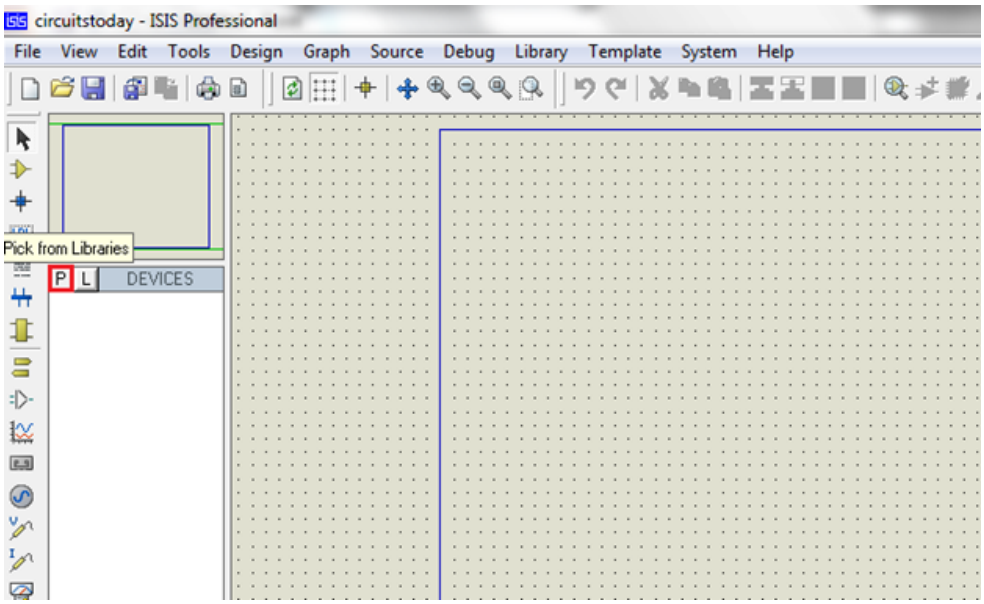

Figure 3.14 Pick from Libraries

Step 6: Select the components from categories or type the part name in Keywords text box.

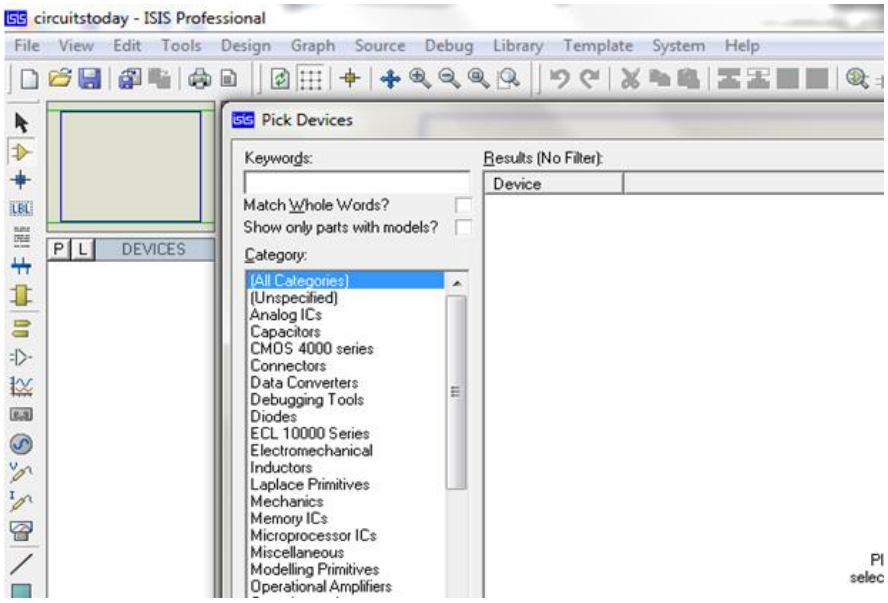

Figure 3.15 Keywords Textbox

Example shows selection of push button. Select the components accordingly.

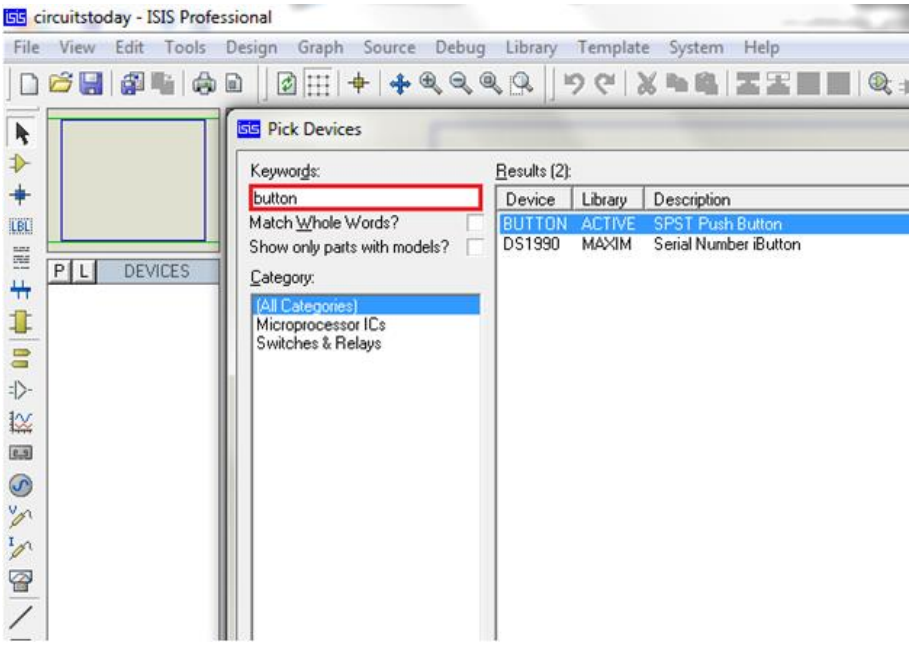

Figure 3.16 Push Button Selection

Step 7: The selected components will appear in the devices list. Select the component and place it in the design sheet by left-click.

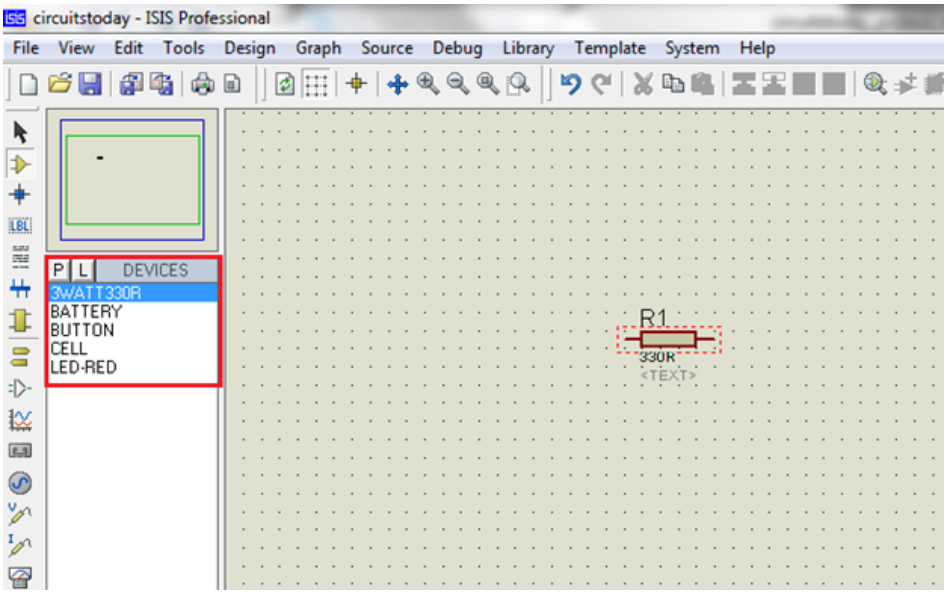

Figure 3.17 Component Selection

Place all the required components and route the wires i.e., make connections.

Either selection mode above the component mode or component mode allows to connect through wires. Left click from one terminal to other to make connection. Double right-click on the connected wire or the component to remove connection or the component respectively.

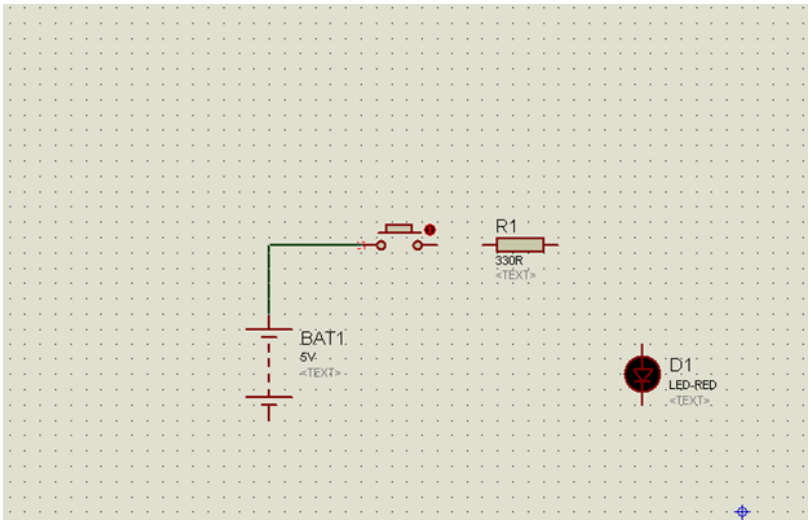

Figure 3.18 Component Properties Selection

Double click on the component to edit the properties of the components and click on Ok.

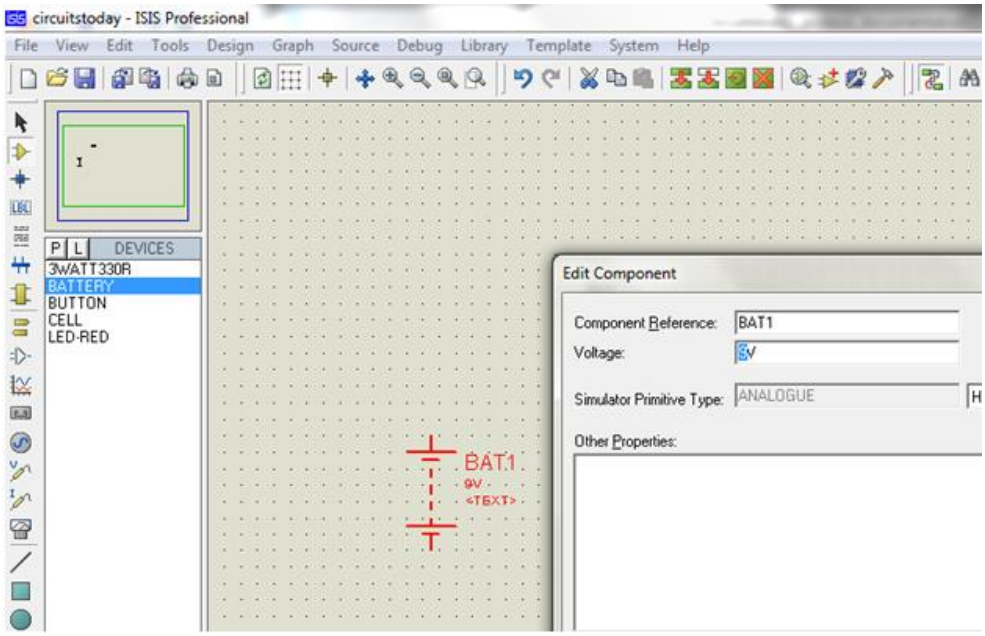

Figure 3.19 Component Properties Edit

Step 8: After connecting the circuit, click on the play button to run the simulation.

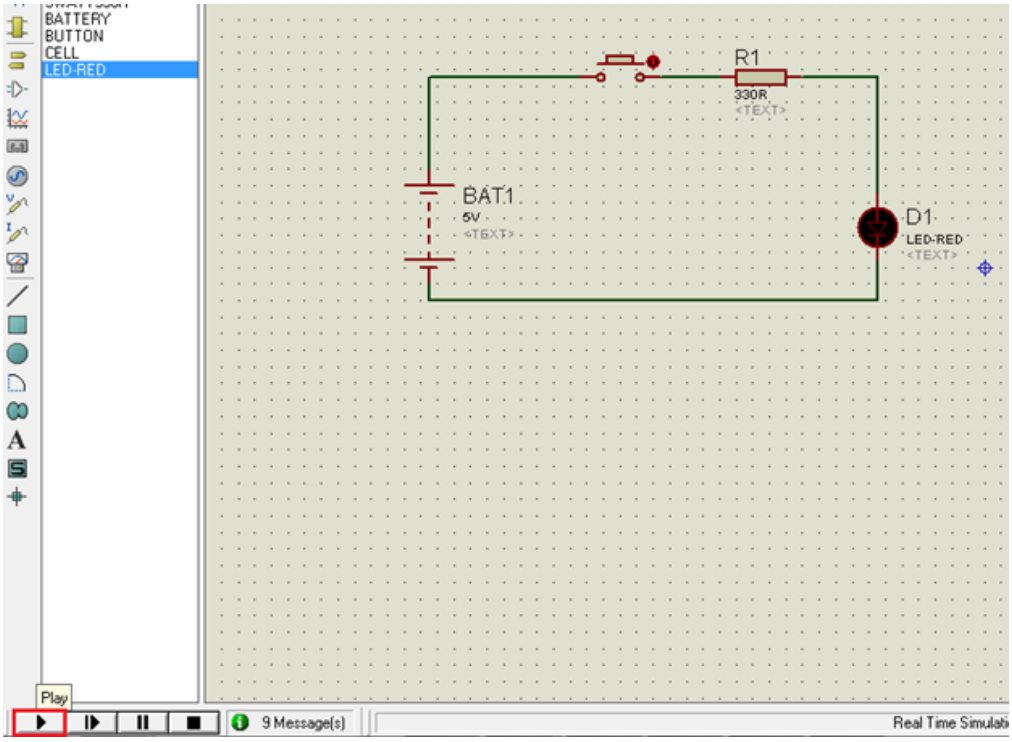

Figure 3.20Simulation Run

In this example simulation, the button is depressed during simulation by clicking on it to make LED glow.

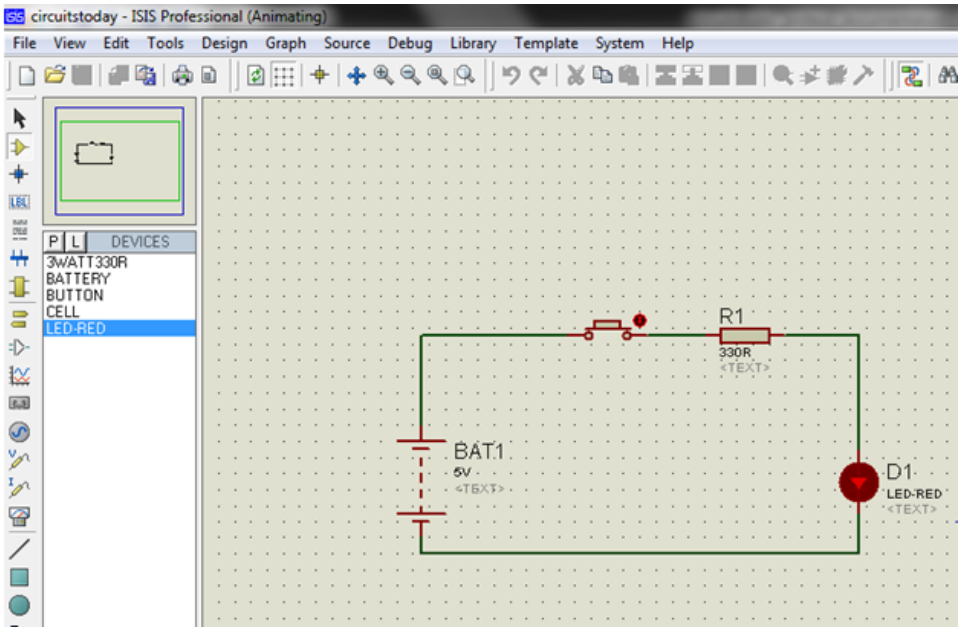

Figure 3.21 Simulation Animating

CHAPTER 4
HARDWARE

8051 MICROCONTROLLERS

4.1 Liquid Cristal Display

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

A program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an controller is an LCD display. Some of the most common LCDs connected to the contollers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Many microcontroller devices use 'smart LCD' displays to output visual information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display. They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS RW D7 D6 D5 D4 D3 D2 D1 D0). For a 4-bit data bus it only requires the supply lines plus 6 extra lines(RS RW D7 D6 D5 D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

4.2 Description Of 16x2:

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 no all-Parallel Ports. It however doesn't show the use of the Status Port as an input. So what are we interfacing? A 16 Character x 2 Line LCD Module to the Parallel Port.

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.

Schematic Diagram

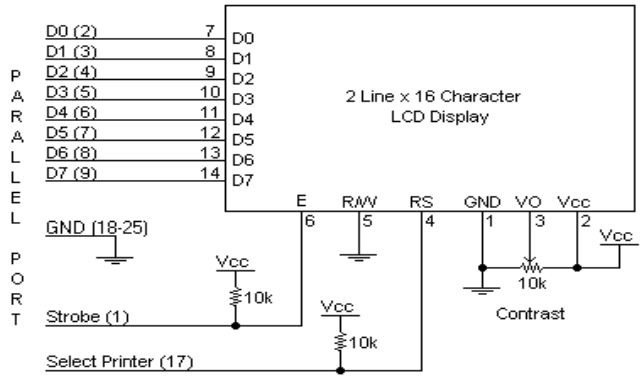

Figure 4.1 schematic diagram

- Above is the quite simple schematic. 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 are 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. 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 a onboard +5 regulator. Remember a few de-coupling capacitors, especially if you have trouble with the circuit working properly.

16 x 2 Alphanumeric LCD Module Features:

- Intelligent, with built-in Hitachi HD44780 compatible LCD controller and RAM providing simple interfacing
- 61 x 15.8 mm viewing area
- 5 x 7 dot matrix format for 2.96 x 5.56 mm characters, plus cursor line
- Can display 224 different symbols

- Low power consumption (1 mA typical)
- Powerful command set and user-produced characters
- TTL and CMOS compatible
- Connector for standard 0.1-pitch pin headers

16 x 2 Alphanumeric LCD Module Specifications:

Pin	Symbol	Level	Function
1	V_{SS}	-	Power, GND
2	V_{DD}	-	Power, 5V
3	Vo	-	Power, for LCD Drive
4	RS	H/L	Register Select Signal H: Data Input L: Instruction Input
5	R/W	H/L	H: Data Read (LCD->MPU) L: Data Write (MPU->LCD)
6	E	H,H->L	Enable
7-14	DB0-DB7	H/L	Data Bus; Software selectable 4- or 8-bit mode
15	NC	-	**NOT CONNECTED**
16	NC	-	**NOT CONNECTED**

Table 4.1Alphanumeric LCD module

FEATURES:

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are:

POSITION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ADDRESS	LINE1	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

Table 4.2 Address locations for a 1x16 line LCD

Even limited to character based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one-, two- and four-line versions.

Several different LC technologies exists. “supertwist” types, for example, offer Improved contrast and viewing angle over the older “twisted nematic” types. Some modules are available with back lighting, so that they can be viewed in dimly-lit conditions. The back lighting may be either “electro-luminescent”, requiring a high voltage inverter circuit, or simple LED illumination.

PIN DESCRIPTION:

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections).

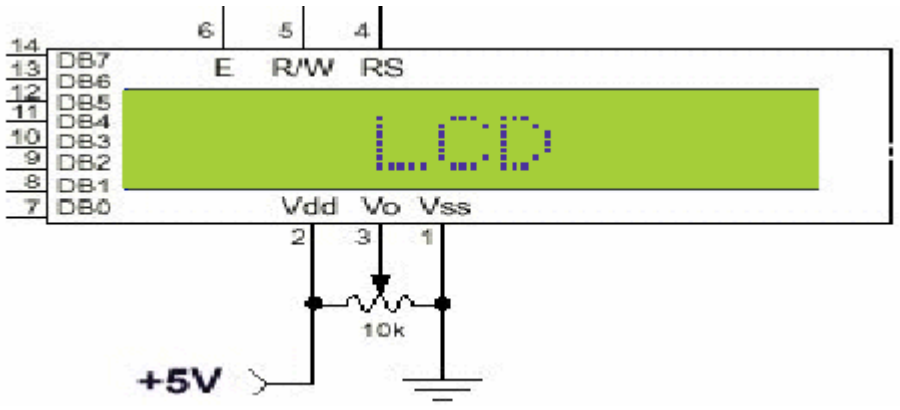

Figure 4.2 Pin diagram of 1x16 lines LCD

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 4.3 Pin specifications

CONTROL LINES:

EN: Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS: Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW: Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

- 1 Character

Writing data to the LCD:

1) Set R/W bit to low
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Read data from data lines (if it is reading) on LCD:

1) Set R/W bit to high
2) Set RS bit to logic 0 or 1 (instruction or character)
3) Set data to data lines (if it is writing)
4) Set E line to high
5) Set E line to low

Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the d.i.l. switch pack with a of re-wiring is necessary.

LCD Commands:

There are some present commands instructions in LCD, which we need to send to LCD through some microcontroller.

Some important command instructions are given below:

Hex Code	Command to LCD Instruction Register
0F	LCD ON, cursor ON
01	Clear display screen
02	Return home
04	Decrement cursor (shift cursor to left)
06	Increment cursor (shift cursor to right)
05	Shift display right
07	Shift display left
0E	Display ON, cursor blinking
80	Force cursor to beginning of first line
C0	Force cursor to beginning of second line
38	2 lines and 5×7 matrix
83	Cursor line 1 position 3
3C	Activate second line
08	Display OFF, cursor OFF
C1	Jump to second line, position 1
OC	Display ON, cursor OFF
C1	Jump to second line, position 1
C2	Jump to second line, position 2

Table 4.4 Commands

RFID (Radio Frequency Identification):

Introduction

Radio Frequency Identification (RFID) technology has been attracting considerable attention with the expectation of improved supply chain visibility for both suppliers and retailers.

It will also improve the consumer shopping experience by making it more likely that

the products they want to purchase are available.

Recent announcements from some key retailers have brought the interest in RFID to the forefront. This guide is an attempt to familiarize the reader with RFID technology so that they can be asking the right questions when considering the technology.

What is RFID?

RFID (Radio Frequency Identification) is a method of identifying unique items using radio waves. Typical RFID systems are made up of 2 major components: readers and tags. The reader, sometimes called the interrogator, sends and receives RF data to and from the tag via antennas. A reader may have multiple antennas that are responsible for sending and receiving the radio waves. The tag, or transponder, is made up of the microchip that stores the data, an antenna, and a carrier to which the chip and antenna are mounted.

RFID technology is used today in many applications, including security and access control, transportation and supply chain tracking. It is a technology that works well for collecting multiple pieces of data on items for tracking and counting purposes in a cooperative environment.

Is All RFID Created Equal?

There are many different versions of RFID that operate at different radio frequencies. The choice of frequency is dependent on the requirements of the application.

Three primary frequency bands have been allocated for RFID use.

Low Frequency (125/134 KHz):

Most commonly used for access control and asset tracking.

Mid-Frequency (13.56 MHz):

Used where medium data rate and read ranges are required.

Ultra-High-Frequency (850 MHz to 950 MHz and 2.4 GHz to 2.5 GHz): offer the longest read ranges and high reading speeds.Applications for RFID within the supply chain can be found at multiple frequencies and different RFID solutions may be required to meet the varying needs of the marketplace. Many of today's RFID technologies cannot reliably cover areas wider than 4 to 5 feet, making them unsuitable for wide openings that are the norm in manufacturing, distribution and store receiving dock environments.

Since UHF (Ultra High Frequency) can cover portals up to 9 feet wide it is gaining industry support as the choice bandwidth for inventory tracking

applications including pallets and cases.
Technology providers are developing readers that work with multiple system protocols and frequencies so that users will be able to choose the RFID products that work best for their market and products.

RFID tags are further broken down into two categories:

Active RFID Tags are battery powered .They broadcast a signal to the reader and can transmit over the greatest Distances (100+ feet).Typically they can cost $4.00 - $20.00 or more and are used to track high value goods like vehicles and large containers of goods. Shipboard containers are a good example of an active RFID tag application

Passive RFID Tags do not contain a battery. Instead, they draw their power from the reader. The reader transmits a low power radio signal through its antenna to the tag, which in turn receives it through its own antenna to power the integrated circuit (chip). The tag will briefly converse with the reader for verification and the exchange of data. As a result, passive tags can transmit information over shorter distances (typically 10 feet or less) than active tags. They have a smaller memory capacity and are considerably lower in cost ($1.00 or less) making them ideal for tracking lower cost items.

There are two basic types of chips available on RFID tags, Read-Only and Read-Write. Read only chips are programmed with unique information stored on them during the manufacturing process. The information on read-only chips can never be changed.
With Read-Write chips, the user can add information to the tag or write over existing information when the tag is within range of the reader. Read-Write chips are more expensive that Read Only chips. Another method used is something called a "WORM" chip (Write Once Read Many).

It can be written once and then becomes "Read only" afterwards. This is a desirable format since companies will be able to write an EPC (electronic product code) to the tag when the product is produced and packaged.

4.3. GSM:

Definition of GSM:

GSM (Global System for Mobile communications) is an open, digital cellular technology used for transmitting mobile voice and data services.

GSM (Global System for Mobile communication) is a digital mobile telephone system that is widely used in Europe and other parts of the world. GSM uses a variation of Time Division Multiple Access (TDMA) and is the most widely used of the three digital wireless telephone technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1,800 MHz frequency band. It supports voice calls and data transfer speeds of up to 9.6 kbit/s, together with the transmission of SMS (Short Message Service).

TDMA systems still rely on switch to determine when to perform a handoff. Handoff occurs when a call is switched from one cell site to another while travelling. The TDMA handset constantly monitors the signals coming from other sites and reports it to the switch without caller's awareness. The switch then uses this information for making better choices for handoff at appropriate times. TDMA handset performs hard handoff, i.e., whenever the user moves from one site to another, it breaks the connection and then provides a new connection with the new site.

Advantages of TDMA:

There are lots of advantages of TDMA in cellular technologies.

1. It can easily adapt to transmission of data as well as voice communication.
2. It has an ability to carry 64 kbps to 120 Mbps of data rates. This allows the operator to do services like fax, voice band data and SMS as well as bandwidth intensive application such as multimedia and video conferencing.
3. Since TDMA technology separates users according to time, it ensures that there will be no interference from simultaneous transmissions.
4. It provides users with an extended battery life, since it transmits only portion of the time during conversations. Since the cell size grows smaller, it proves to save base station equipment, space and maintenance.

TDMA is the most cost effective technology to convert an analogy system to digital.

Disadvantages of TDMA:

One major disadvantage using TDMA technology is that the users has a predefined time slot. When moving from one cell site to other, if all the time slots in this cell are full the user might be disconnected. Likewise, if all the time slots in the cell in which the user is currently in are already occupied, the user will not receive a dial tone.

The second problem in TDMA is that it is subjected to multipath distortion. To overcome this distortion, a time limit can be used on the system. Once the time limit is expired, the signal is ignored.

Architecture of the GSM Network

In a GSM network, the user terminal is called a **mobile station**. A mobile station is made up of a **SIM** (*Subscriber Identity Module*) card allowing the user to be uniquely identified and a mobile terminal. The terminals (devices) are identified by a unique 15-digit identification number called **IMEI** (*International Mobile Equipment Identity*). Each SIM card also has a unique (and secret) identification number called **IMSI** (*International Mobile Subscriber Identity*). This code can be protected using a 4-digit key called a *PIN code*.

The SIM card therefore allows each user to be identified independently of the terminal used during communication with a base station. Communications occur through a radio link (air interface) between a mobile station and a base stati

All the base stations of a cellular network are connected to a **base station controller** (**BSC**) which is responsible for managing distribution of the resources. The system consisting of the base station controller and its connected base stations is called the **Base Station Subsystem** (**BSS**).

Finally, the base station controllers are themselves physically connected to the **Mobile Switching Centre** (**MSC**), managed by the telephone network operator, which connects them to the public telephone network and the Internet.

The MSC belongs to a **Network Station Subsystem** (**NSS**), which is responsible for managing user identities, their location and establishment of communications with other subscribers. The MSC is generally connected to databases that provide additional functions:

1. The **Home Location Register** (**HLR**) is a database containing information (geographic position, administrative information etc.) of the subscribers registered in the area of the switch (MSC).
2. The **Visitor Location Register** (**VLR**) is a database containing information of users other than the local subscribers. The VLR retrieves the data of a new user from the HLR of the user's subscriber zone. The data is maintained as long as the user is in the zone and is deleted when the user leaves or after a long period of inactivity (terminal off).
3. The **Equipment Identify Register** (**EIR**) is a database listing the mobile terminals.
4. The **Authentication Centre** (**AUC**) is responsible for verifying user identities.
5. The cellular network formed in this way is designed to support mobility via management of *handovers* (movements from one cell to another).

Finally, GSM networks support the concept of **roaming** i.e., movement from one operator network to another.

4.5.GSM Modem:

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer.

FIGURE 4.3 GSM modem

It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.

A SIM card contains the following information:

- Subscriber telephone number (MSISDN)
- International subscriber number (**IMSI, International Mobile Subscriber Identity)**
- State of the SIM card
- Service code (operator)
- Authentication key
- PIN (***Personal Identification Code***)
- PUK (***Personal Unlock Code***)

Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands.

These extended AT commands are defined in the GSM standards. With the extended AT commands, the following operations can be performed:

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

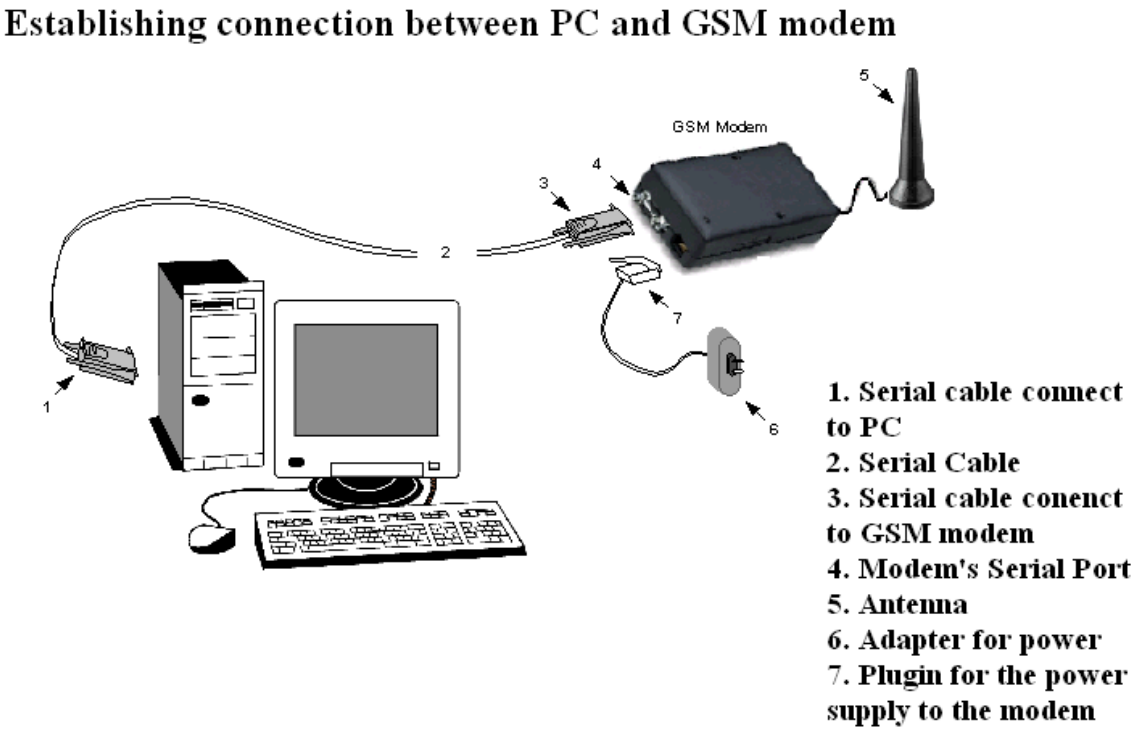

Figure 4.4 establishing connection b/w pc and GSM

The number of SMS messages that can be processed by a GSM modem per minute is very low i.e., about 6 to 10 SMS messages per minute.

Introduction to AT Commands

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's the reason, modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state) are also supported by GSM modems and mobile phones.

Besides this common AT command set, GSM modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

It should be noted that the starting "AT" is the prefix that informs the modem about the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS.

Some of the tasks that can be done using AT commands with a GSM modem or mobile phone are listed below:

- Get basic information about the mobile phone or GSM modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).
- Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).
- Get the current status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).
- Establish a data connection or voice connection to a remote modem (ATD, ATA, etc).
- Send and receive fax (ATD, ATA, AT+F*).
- Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain notifications of newly received SMS messages (AT+CNMI).
- Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.
- Perform security-related tasks, such as opening or closing facility locks (AT+CLCK), checking whether a facility is locked (AT+CLCK) and changing passwords(AT+CPWD).

  (Facility lock examples: SIM lock [a password must be given to the SIM card

every time the mobile phone is switched on] and PH-SIM lock [a certain SIM card is associated with the mobile phone. To use other SIM cards with the mobile phone, a password must be entered.])

- Control the presentation of result codes / error messages of AT commands. For example, the user can control whether to enable certain error messages (AT+CMEE) and whether error messages should be displayed in numeric format or verbose format (AT+CMEE=1 or AT+CMEE=2).
- Get or change the configurations of the mobile phone or GSM/GPRS modem. For example, change the GSM network (AT+COPS), bearer service type (AT+CBST), radio link protocol parameters (AT+CRLP), SMS center address (AT+CSCA) and storage of SMS messages (AT+CPMS).
- Save and restore configurations of the mobile phone or GSM/GPRS modem. For example, save (AT+CSAS) and restore (AT+CRES) settings related to SMS messaging such as the SMS center address.

It should be noted that the mobile phone manufacturers usually do not implement all AT commands, command parameters and parameter values in their mobile phones. Also, the behavior of the implemented AT commands may be different from that defined in the standard.

CHAPTER-5

RESULT AND DESCRIPTION

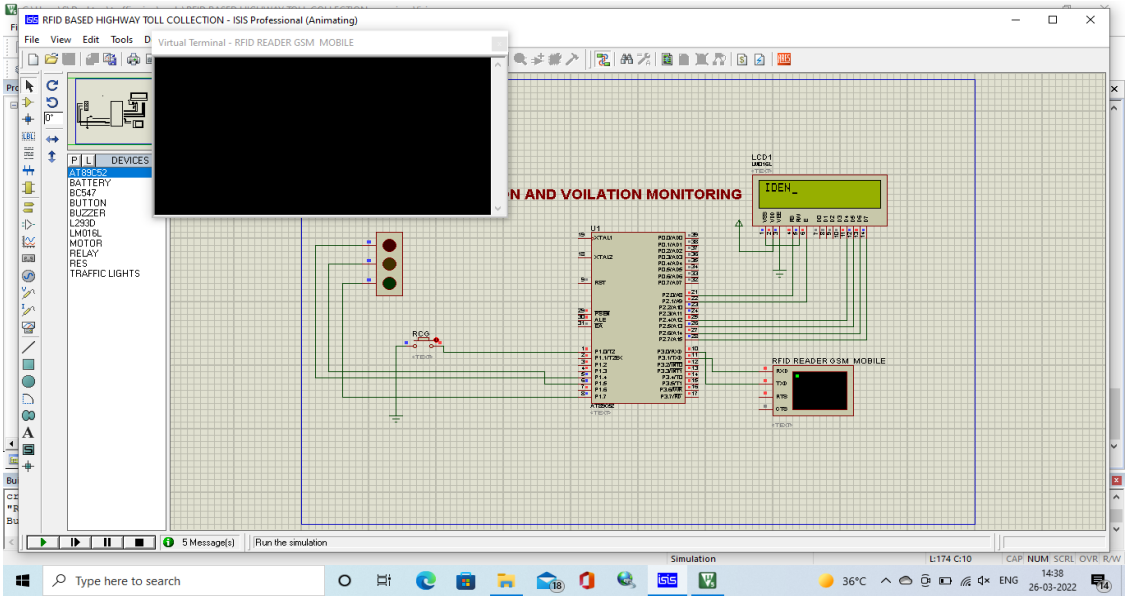

Fig 5.1 program installation

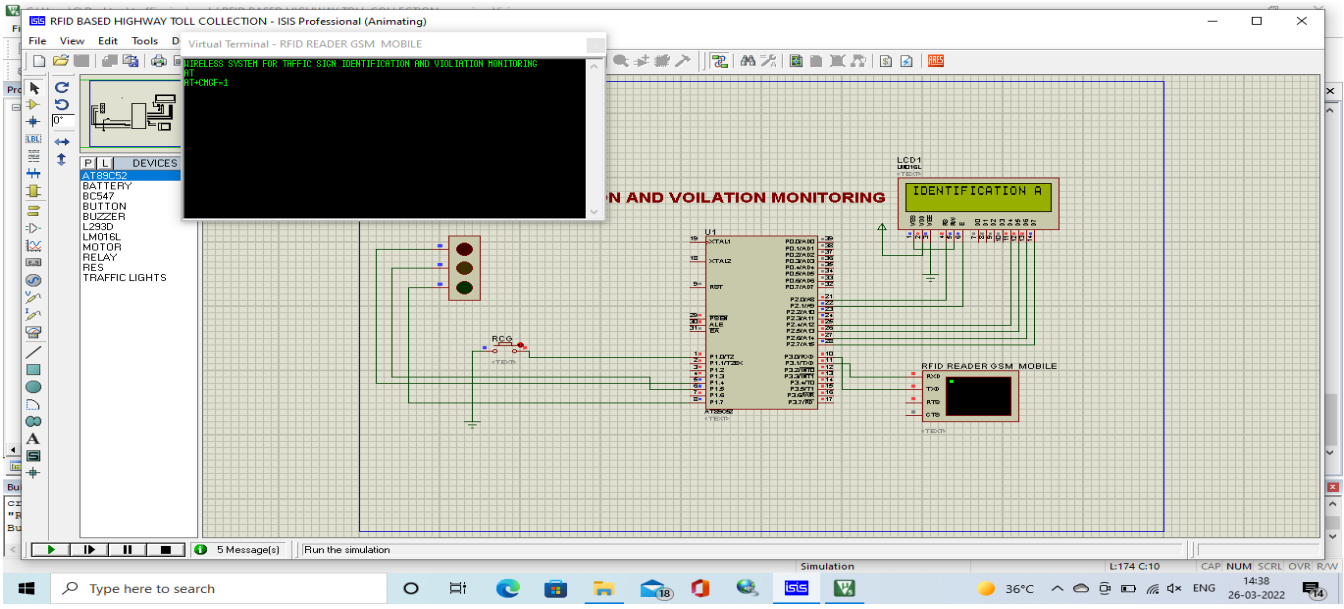

Figure 5.2 compiling code

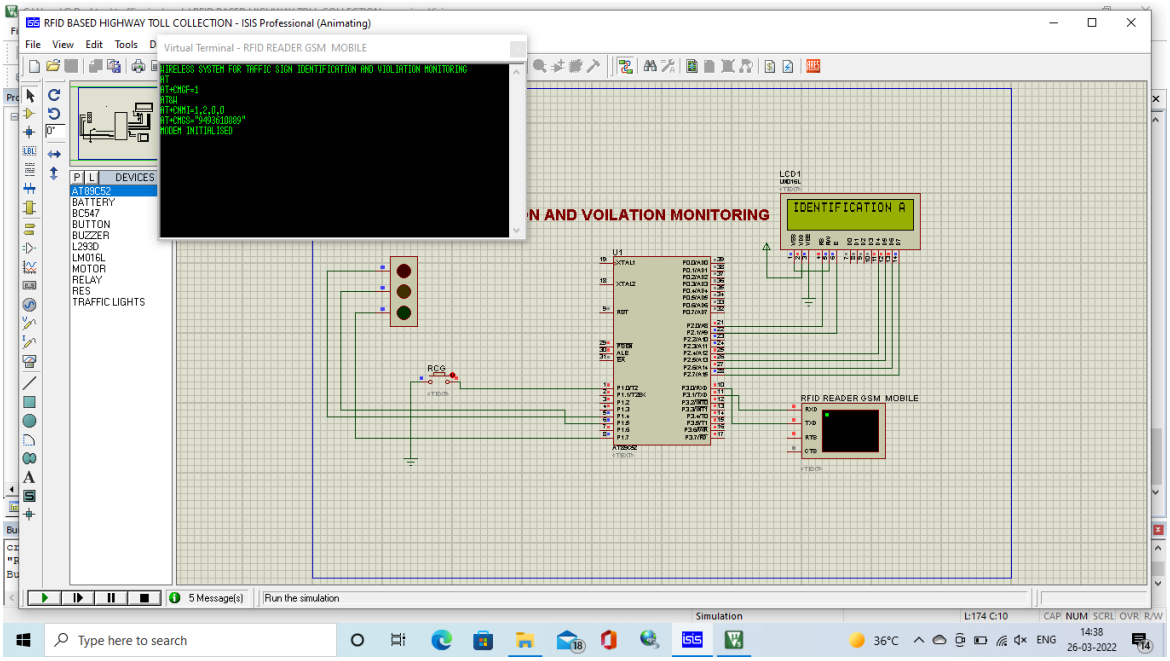

Figure 5.3 before signal

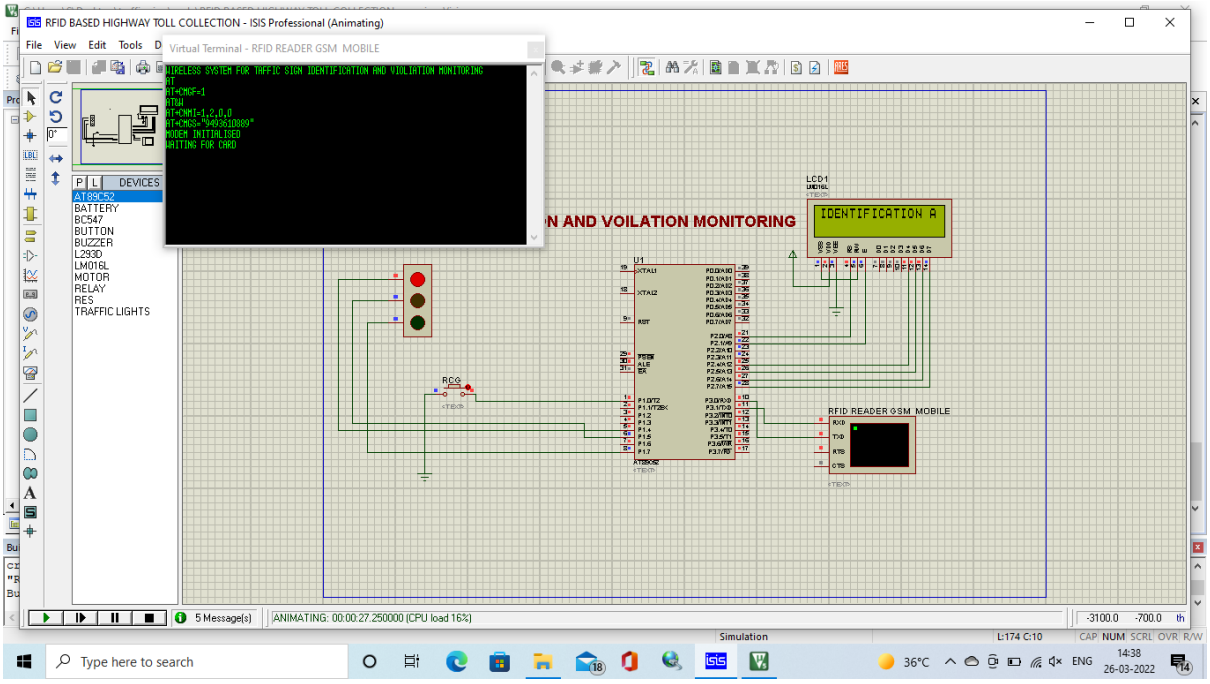

Figure 5.4 after signal

The above figures shows the ouput of the project .i.e how the process takes place if a person crosses the traffic rules and how the fine is implemented for them .This total process takes place through database system.

CHAPTER 6

CONCLUSION

We have proposed a system for automatic detection and penalty management of signal violation which will in turn help to decrease the number of accidents. The system also analyses the traffic flow on a given road at a given time according to the circumstances of the road. The proposed architecture is portable, accurate and can be installed at a reasonable cost. The system alleviates the need for traffic police at every signal to manually identify the violations. The system shows promising results on automatic detection, since the detection of the tag identification is more precise, reliable and efficient in active RFID, leading to implement corrective actions. The experiments are done using four test vehicles. The frequency of RFID reader is 125 Hz and the range of the RFID reader is 1 meter. In real time environment, if the average road lane is 3.5m, then approximately two 125 Hz readers are required for one road lane. Many vehicles may be moving nearby and possibly blocking or attenuating some of the RFID signals, especially with large vehicles like trucks. A possible solution is the use of RFID readers of higher frequency range since their reading range is high.

References

1. Perez J, Seco F, Milanes V, Jimenez A, Díaz JC, De Pedro T, "An RFID-based intelligent vehicle speed controller using active traffic signals", Sensors, 2010 Jun 9, 10(6), pp: 5872-87.

2. Cheng-an ZT, Min-lu J, "A RFID Based Traffic Information Acquisition System and Vehicle Positioning Method", Journal of Electronics & Information Technology, 2010, pp: 11-15. International Journal of Pure and Applied Mathematics Special Issue 326

3. Pandit AA, Mundra AK, Talreja J, "RFID tracking system for vehicles (RTSV)", In Computational Intelligence, Communication Systems and Networks (IEEE), 2009, CICSYN'09, First International Conference on 2009 Jul 23, pp. 160-165.

4. Min Z, Wenfeng L, Zhongyun W, Bin L, Xia R, "A RFID-based material tracking information system", In Automation and Logistics, 2007 IEEE International Conference on 2007 Aug 18, pp. 2922-2926.

5. Zhang X, Tentzeris M, "Applications of fast-moving RFID tags in high-speed railway systems", International Journal of Engineering Business Management, 2011 Feb 1, pp:3:6. 6. Khali H, Araar A, Abdulla EZ, "Suitability of passive RFID technology for fast moving vehicle identification"

6. Journal of Emerging Trends in Computing and Information Sciences, January 2014.
7. Li C, "Automatic vehicle identification (AVI) system based on RFID", In Anti-Counterfeiting Security and Identification in Communication (ASID), 2010 IEEE International Conference, 2010 Jul 18, pp. 281-284.

APPENDIX B

SOURCE CODE

```
#include<reg51.h>

#include<string.h>

unsigned convert (unsigned int);

#define lcd_data P2

sbit  lcd_rs   =  P2^0;

sbit  lcd_en   =  P2^1;

sbit rcg = P1^0;

sbit r = P1^4;

sbit o = P1^5;

sbit g = P1^7;

void delay(unsigned int v)

{

unsigned int i,j;

for(i=0;i<=v;i++)

for(j=0;j<=1275;j++);

}

void lcdcmd(unsigned char value)              // LCD COMMAND

 {
```

```
 lcd_data=value&(0xf0); //send msb 4 bits
 lcd_rs=0;     //select command register
  lcd_en=1;   //enable the lcd to execute command
        delay(3);
  lcd_en=0;
  lcd_data=((value<<4)&(0xf0));  //send lsb 4 bits
  lcd_rs=0;   //select command register
  lcd_en=1;  //enable the lcd to execute command
         delay(3);
         lcd_en=0;
  }
 void lcd_init(void)
 {
lcdcmd(0x02);
lcdcmd(0x02);
 lcdcmd(0x28);  //intialise the lcd in 4 bit mode
 lcdcmd(0x28);  //intialise the lcd in 4 bit mode
lcdcmd(0x0e);       //cursor blinking
lcdcmd(0x06);       //move the cursor to right side
lcdcmd(0x01);       //clear the lcd
```

```
}
void lcddata(unsigned char value)
 {
    lcd_data=value&(0xf0); //send msb 4 bits
    lcd_rs=1;   //select data register
    lcd_en=1;           //enable the lcd to execute data
          delay(3);
    lcd_en=0;
    lcd_data=((value<<4)&(0xf0)); //send lsb 4 bits
    lcd_rs=1;   //select data register
    lcd_en=1;           //enable the lcd to execute data
          delay(3);
   lcd_en=0;
   delay(3);
   }
  void msgdisplay(unsigned char b[]) // send string to lcd
  {
   unsigned char s,count=0;
  for(s=0;b[s]!='\0';s++)
 {
 count++;
```

```
 if(s==16)
lcdcmd(0xc0);
  if(s==32)
  {
  lcdcmd(1);
  count=0;
  }
 lcddata(b[s]);
}
}
void tx(unsigned char *tx)
{
  for(;*tx != '\0';tx++)
    {
          SBUF=*tx;
         while(TI == 0);
          TI=0;
        }
}

void tx1(unsigned char tx)
```

```
 SBUF=tx;
      while(TI == 0);
      TI=0;
}
   unsigned char receive()
{
 unsigned char rx;
  while(RI == 0);
   rx=SBUF;
         RI=0;
         return rx;
}
void main(void)
{
  unsigned char count,str1[12];
 r=o=g=0;
 TMOD=0x20;
 TH1=0xfd;//// 9600
 SCON=0x50;
 TR1=1;
 lcd_init();
```

```
 msgdisplay("WIRELESS SYSTEM FOR TAFFIC SIGN
IDENTIFICATION AND VIOLIATION MONITORING");

        tx("WIRELESS SYSTEM FOR TAFFIC SIGN IDENTIFICATION
AND VIOLIATION MONITORING\r\n");

  tx("AT\r\n");

    delay(250);

    tx("AT+CMGF=1\r\n");

    delay(250);

    tx("AT&W\r\n");

    delay(250);

    tx("AT+CNMI=1,2,0,0\r\n");

    delay(250);

    tx("AT+CMGS=\"9493610889\"\r\n");

    tx("MODEM INITIALISED\r\n");

            tx("AT+CMGS=\"77889954625\"\r\n");

    tx("MODEM INITIALISED\r\n");

    tx1(0x1A);

 while(1)

     {

  loop:r=0;o=0;g=1;

  if(rcg == 0)

      {
```

```
 r=1;o=0;g=0;
 tx("WAITING FOR CARD\r\n");
      for(count=0;count < 12;count++)
      {
      str1[count]= receive();
       }
  if((str1[8]=='7' && str1[9]=='3') || (str1[9]=='7' && str1[10]=='3') )
            {
tx("AT+CMGS=\"9493610889\"\r\n");
                  tx("VECHILE NO : TS 00 AA 0000\r\n");
tx1(0x1A);
                  tx("SIGNAL JUMP\r\n");
tx1(0x1A);
                 tx("FINE 2000/-\r\n");
tx1(0x1A);
            }
  if((str1[8]=='8' && str1[9]=='4') || (str1[9]=='8' && str1[10]=='4') )
            {
tx("AT+CMGS=\"7799885566\"\r\n");
                  tx("VECHILE NO : TS 22 AA 1243\r\n");
tx1(0x1A);
```

```
tx("SIGNAL JUMP \r\n");

tx1(0x1A);

tx("FINE 2000/-\r\n");

tx1(0x1A);

}

goto loop;

  }

 }

}
```