AUTOMATIC CONTINUOUS VISITOR MONITORING SYSTEM USING AT89C51

A project submitted in partial fulfillment of the
Requirement for the award of degree of

Bachelor of Technology
In
Electronics and Communication Engineering
By

B.L.N.Praveen 13KQ1A0497
K.Rama Krishna 14KQ5A0413
V.Sai Thanuja 13KQ1A0495
M.Kondala Rao 13KQ1A04A4

Under the Guidance of
Mr. P.Siva Krishna, M.Tech.
Assistant Professor

Department of Electronics & Communication Engineering
PACE INSTITUTE OF TECHNOLOGY AND SCIENCES
(Affiliated to Jawaharlal Nehru Technological University, Kakinada) Vallur, NH-5,
Ongole, Prakasam District. Pin: 523272
ACKNOWLEDGEMENT

We thank the almighty for giving us the courage and perseverance in completing the main-project. This project itself is acknowledgements for all those people who have give us their heartfelt co-operation in making this project a grand success.

We extend our sincere thanks to Dr. M. VENU GOPAL RAO, B.E, D.M.M. chairman of our college, for providing sufficient infrastructure and good environment in the college to complete our course.

We are thankful to our secretary Mr. M. SRIDHAR, B.E, for providing the necessary infrastructure and labs and also permitting to carry out this project.

We are thankful to our principal Dr.C.V. SUBBA RAO, B.Tech, M.E, Ph.D, MISTE for providing the necessary infrastructure and labs and also permitting to carry out this project.

With extreme jubilance and deepest gratitude, we would like to thank Head of the E.C.E. Department, Mr. M.APPARAO, M.Tech, MBA, (Ph.D) for his constant encouragement.

We are greatly indebted to project guide, Mr.P.SIVA KRISHNA, M.Tech. Assistant Professor, Electronics and Communication engineering, for providing valuable guidance at every stage of this project work. We are profoundly grateful towards the unmatched services rendered by him.

My Special thanks to our project coordinator Mr. B. SIVA PRASAD, M.Tech. Associate Professor, Electronics and Communication engineering, for his support and valuable suggestions regarding project work.

Our special thanks to all the faculty of Electronics and Communication Engineering and peers for their valuable advises at every stage of this work.

Last but not least, we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the main project.
## INDEX

<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>V</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>VII</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>VIII</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>XI</td>
</tr>
</tbody>
</table>

## CHAPTER 1

INTRODUCTION 1

1.1 OBJECTIVE OF THE PROJECT 1

1.2 BACKGROUND OF THE PROJECT 1

## CHAPTER 2

INTRODUCTION TO EMBEDDED SYSTEMS 2

2.1 INTRODUCTION 2

2.2 APPLICATION OF EMBEDDED SYSTEM 4

2.3 MILITARY AND AEROSPACE SOFTWARE APPLICATION 4

2.4 COMMUNICATION APPLICATION 4

2.5 ELECTRONIC APPLICATIONS AND CONSUMER DEVICES 5

2.6 INDUSTRIAL AUTOMATION AND PROCESS CONTROL 5
CHAPTER 3
HARDWARE IMPLEMENTATION OF THE PROJECT

3.1 Project design

3.1.1 Block diagram of the project

3.2 POWER SUPPLY

3.3 MICROCONTROLLERS

3.3.1 Differences between micro processor and micro controller

3.3.2 Microcontroller architecture and features

3.3.3 Synchronous serial communication

3.3.4 Asynchronous serial communication

3.3.5 A Brief History of 8051

3.3.6 Necessity of Microcontrollers

3.3.7 Features of 8051 Microcontroller

3.3.8 Pin Diagram

3.3.9 Internal Architecture of 8051 microcontroller

3.3.10 Special Functional Register Memory (SFR)

3.4 LM358 COMPARATOR

3.4.1 Features of LM358

3.5 INRARED SENSORS

3.6 ALPHA-NUMERIC LCD DISPLAY
CHAPTER 4
SOFTWARE IMPLEMENTATIONS 26
4.1 KEIL SOFTWARE 26
    4.1.1 Introduction to Micro vision Keil (IDE) 26
    4.1.2 Basic Keil Tutorial 26
4.1.3 Starting a new Assembler Project 26
4.2 MICRO C FLASH 34
4.3 CODING 34

CHAPTER 5
WORKING PROCEDURE 39
    5.1 BEFORE EXECUTION 40
    5.2 AFTER EXECUTION 41

CONCLUSION 43
FUTURE ENHANCEMENT 44

REFERENCES 45
List of Figures

Figure 2.1 Block diagram of embedded system ........................................7

Figure 3.1 Block diagram wireless black box............................................7

Figure 3.2 Block Diagram of a Regulated Power Supply System.....................8

Figure 3.3.1 Architecture of 8051 microcontroller ................................10

Figure 3.3.2 Pin diagram of 8051 microcontroller .......................................15

Figure 3.3.3 shows internal architecture of 8051 microcontroller ....................19

Figure 3.3.4 Power control Register ..........................................................20

Figure 3.3.5 Timer control Register ..........................................................20

Figure 3.3.6 Timer mode Register ............................................................21

Figure 3.4 LM358 comparator .....................................................................22

Figure 3.5 Infrared sensor ..........................................................................23

Figure 3.6 Typical 16*2 LCD .....................................................................24
List of Tables

Table 3.3.8: Alternate usage pins of port3 ............................................................... 16
Table 3.3.8.2: shows about the interrupt sources ...................................................... 18
LIST OF ABBREVIATIONS

ALU ..................... Arithmetic and Logic Unit
CPU ...................... Central Processing Unit
DC ....................... Direct Current
ESD ..................... Electro Static Discharge
VCC ..................... Digital power supply
GND ..................... Ground
IE ....................... Interrupt Enable
IP ....................... Interrupt priority
ISP ..................... In-System Programmable
IEEE ................... Institute of Electrical and Electronics Engineers
INT ..................... Interrupt
I/O ..................... Input/output
μC ...................... Microcontroller
MCU ................... Microcontroller unit
ALE ..................... Address latch enable
SFR ..................... Special function registers
PCON .................. Power control register
TCON .................. Timer control registers
TMOD .................. Timer mode
ROM ................... Read only memory
RAM ................... Random access memory
UART .................. Universal asynchronous receiver/transmitter
WI-FI .................. wireless Fidelity
ABSTRACT

In the advancement of technology the art of reducing time to do a particular task has become more important. As there is a situation to know the number of people entering in an auditorium and counting only one person at a time. There is a need of knowing number of people presented at a particular instant.

The objective of Bi-directional visitor counter is to count the number of visitors presented in a given auditorium or a stadium at a given time.

The following are the drawbacks in the existed system:

1. The counter which is already existed will count as a single person even more than one person enter the hall at a time.

2. The capacity of the counter is only upto two digits.

Extension:

This project will eliminate the counting problem when more that 1 person enter into auditorium and also the capacity of counter will also be extended upto 6 digits.
CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT
This Project “Automatic Continuous Visitor Monitoring System using AT89C51 Microcontroller” is a reliable circuit that takes over the task of counting number of persons/visitors in the room very accurately. When somebody enters into the room then the counter is incremented by one and when any one leaves the room then the counter is decremented by one. The total number of persons inside the room is also displayed on the LCD displays.

The microcontroller does the above job. It receives the signals from the sensors, and this signal is operated under the control of software which is stored in ROM. Microcontroller AT89C51 continuously monitor the Infrared Receivers, When any object pass through the IR Receiver’s then the IR Rays falling on the receiver are obstructed, this obstruction is sensed by the Microcontroller.

1.2 BACKGROUND OF THE PROJECT
The software application and the hardware implementation help the microcontroller read the data received from with the help of IR sensors they can detect the entry and exit of a visitor. The data is send from IR transmitter to IR receiver. The system is totally designed using Sensor Technology and embedded system technology. The performance of the design is maintained by the controlling unit.
CHAPTER-2

INTRODUCTION TO EMBEDDED SYSTEM

2.1 INTRODUCTION

The microprocessor-based system is built for controlling a function or range of functions and is not designed to be programmed by the end user in the same way a PC is defined as an embedded system. An embedded system is designed to perform one particular task albeit with different choices and options.

Embedded systems contain processing cores that are either microcontrollers or digital signal processors. Microcontrollers are generally known as "chip", which may itself be packaged with other microcontrollers in a hybrid system of Application Specific Integrated Circuit (ASIC). In general, input always comes from a detector or sensors in more specific word and meanwhile the output goes to the activator which may start or stop the operation of the machine or the operating system.

An embedded system is a combination of both hardware and software, each embedded system is unique and the hardware is highly specialized in the application domain. Hardware consists of processors, microcontroller, IR sensors etc. On the other hand, Software is just like a brain of the whole embedded system as this consists of the programming languages used which makes hardware work. As a result, embedded systems programming can be a widely varying experience.

An embedded system is combination of computer hardware and software, either fixed incapability or programmable, that is specifically designed for particular kind of application device. Industrial machines, automobiles, medical equipment, vending machines and toys (as well as the more obvious cellular phone and PDA) are among the myriad possible hosts of an embedded system. Embedded systems that are programmable are provided with a programming interface, and embedded systems programming id specialized occupation.
Figure 2.1 Block diagram of embedded system

Figure 2.1 illustrates the Block diagram of Embedded System (ES consists of hardware and software part which again consists of programming language and physical peripherals respectively).

On the other hand, the microcontroller is a single silicon chip consisting of all input, output and peripherals on it. A single microcontroller has the following features:
1. Arithmetic and logic unit
2. Memory for storing program
3. EEPROM for nonvolatile and special function registers
4. Input/output ports
5. Analog to digital converter
6. Circuits
7. Serial communication ports

A microprocessor-based system which is built for controlling a function or range of functions and is not designed to be programmed by the end user in the same
way a PC is defined as an embedded system. An embedded system is designed to perform one particular task albeit with different choices and options.

2.2 APPLICATIONS OF EMBEDDED SYSTEM

We are living in the embedded world. You are surrounded with many embedded products and your daily life largely depends on the proper functioning’s of these gadgets, television, radio, CD layer of your living room, washing machines or microwave oven in your kitchen, card readers, access controllers, palm devices of your work space enable to do many of your tasks very effectively. Apart from all these, many controllers embedded in your car take care of your car operation between the bumper and most of the times tend to ignore all these controllers.

In recent days you are showered with variety of information about these embedded controllers in many places. All kind of magazines and journals regularly dish out details about latest technologies, new devices: fast applications which make you believe that your basic survival is controlled by these embedded products. Now you can agree to that fact these embedded products have successfully invaded into our world, you must be wandering about these embedded controllers or systems.

The computer you use to compose your mails, or create a document or analyze the database is known as standard desktop computer. These desktop computers are manufactured to serve many purpose and applications.

2.3 MILITARY AND AEROSPACE SOFTWARE APPLICATIONS

From in-orbit embedded system to jumbo jets to vital battlefield networks, designer’s performance, capability, and high-availability facilities consistently turn to the Linux OS, RTOS and LinuxOS-178RTOS for software certification to DO-178B rich in system resources and networking serviced, Linux OS provides an off-the-shelf software platform with hard real-time response backed by powerful distributed computing (COBRA), high reliability’s software certification, and long term support options.

2.4 COMMUNICATIONS APPLICATIONS

“Five-nine” availability, compact PCI hot swap support, and hard real-time response Linux OS delivers on these key requirements and more for today’s carrier-class systems. Scalable kernel configurations, distributed computing capabilities, intergraded communications stacks, and fault-management facilities make Linux OS
the ideal choice for companies looking for single operating system for all embedded telecommunication applications from complex central to single line/trunk cards.

2.5 ELECTRONIC APPLICATIONS AND CONSUMER DEVICES

As the number of powerful embedded processor in consumer devices continues to rise, the blue cat Linux operating system provides a highly reliable and royalty-free option for system designers. And as the wireless appliance revolution rolls on, web enabled navigation systems, radios, personal communication devices, phones and PDAs all benefit from the cost-effective dependability, proven stability and full product life cycle support opportunities associated with blue cat embedded Linux. Blue cat has teamed with industry leaders to make it easier to build Linux mobile phones with java integration.

2.6 INDUSTRIAL AUTOMATION AND PROCESS CONTROL SOFTWARE

Designers of industrial and process control systems know from experience that Linux works operating system provide the security and reliability that their industrial applications require. From ISO 9001 certification to fault-tolerance, secure portioning and high availability, we’ve got it all. The advantage of our 20 years of experience with the embedded system. Now a day’s embedded system widely using in the industrial areas to reduce to tike perform the particular task .This replacing the less work and also more efficient gives the accurate result.
CHAPTER-3

HARDWARE IMPLEMENTATION OF THE PROJECT

3.1 PROJECT DESIGN

This chapter briefly explains about the hardware implementation of the project. It discusses the design and working of design with the help of block diagram and circuit diagram and explanation of circuit diagram in detail. It explains the features, timer programming, serial communication, interrupts of AT89C51 microcontroller. It also explains the various modules used in this project.

Project design:
The implementation of the project design can be divided into two parts.

1. Hardware implementation
2. Firmware implementation

Hardware implementation deals in drawing the schematic on the plane paper according to the application testing the schematic design over the breadboard using the various IC’s to find if the design meets the objective carrying out the PCB layout of the schematic tested on breadboard finally preparing the board the testing the designed hardware.

The firmware part deals with the programming the controller so that it can control the operation of the IC’s used in the implementation. In the present work we have used the PROTEUS software for PCB circuit design, KEIL software development tool write and compile the source into the microcontroller. The firmware implementation is explained the next chapter.

The project description and principle are explained in this chapter using the block diagram and circuit diagram. The block diagram discusses about the required components of the design and working condition is explained using the circuit diagram and system wiring diagram.
3.1.1 Block diagram of the project:

**Figure: 3.1 block diagram of transmitter unit**
3.2 POWER SUPPLY

Block diagram of power supply

There are many types of power supply. Most are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronics circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. For example a 5V regulated supply can be shown as below

![Block Diagram of a Regulated Power Supply System](image)

Similarly, 12v regulated supply can also be produced by suitable selection of the individual elements. Each of the blocks is described in detail below and the power supplies made from these blocks are described below with a circuit diagram and a graph of their output.

Regulator:

Regulator eliminates ripple by setting DC output to a fixed voltage. Voltage regulator ICs are available with fixed (typically 5V, 12V and 15V) or variable output voltages. Negative voltage regulators are also available. Many of the fixed voltage regulator ICs has 3 leads (input, output and high impedance). They include a hole for attaching a heat sink if necessary. Zener diode is an example of fixed regulator which is shown here.

3.3 MICROCONTROLLERS

3.3.1 Differences between micro processor and micro controller

The micro controller is different from micro processor. The Microprocessor is meant the general purpose Microprocessor such as Intel’s X86 family (8086,80286,80386,80486 and the Pentium)or Motorola’s 680X0 family(68000,68010,68020,68030,68040,etc).These microprocessors contains no RAM, no ROM and the no I/O ports on the chip itself .For this reason, they are commonly referred to as general purpose Microprocessors.
A system designer using a general purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports and the timers externally to make them functional. Although the addition of external RAM, ROM and I/O ports makes this system bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand. This is not the case with microcontrollers.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports and the timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip. In many applications, for example a TV remote control, there is no need for the computing power of 486 or even as 8086 Microprocessor. These applications most often require some I/O operations to read the signals and turn on and off certain bits.

In Embedded system there is only one application software that is typically burned into ROM. An x86 PC contains or it is connected to various embedded products such as keyboard, printer, printer modem, disk controller, sound card, CD-ROM drives, mouse and so on. Each one of these peripherals has a Microcontroller inside it that performs only one task. For example, inside every mouse there is Microcontroller to perform the task of finding the mouse position and sending it to this PC.

3.3.2 Microcontroller Architecture and Features:

The basic internal designs of micro controllers are pretty similar. Figure shows the block diagram of a typical micro controller. All components are connected via an internal bus and are all integrated on one chip. The modules are connected to the outside world via I/O pins.
Figure: 3.3.1: Architecture of 8051 microcontroller

The following list contains the modules typically found in a microcontroller.

**CPU:**

The CPU of the controller contains the arithmetic and logic unit, the control unit, and the registers (stack pointer, program counter, accumulator register, register file).

**MEMORY:**

The memory sometimes split into program memory and data memory. In larger controllers, a DMA controller handles data transfer between peripheral components and the memory.

**INTERRUPT CONTROLLER:**

Interrupts are useful for interrupting the normal program flow in case of (important) external or internal events. In conjunction with sleep modes, they help to conserve power.

**TIMER/COUNTER:**

Most controller have at least one and more likely 2-3 timers/counters, which can be used to timestamp events, measure intervals, or count events. Many controllers also contain PWM (pulse width modulation) outputs, which can be used to drive motors or for safe breaking (antilock breaking system). Furthermore the PWM output
can, in conjunction with external filter, be used to realize a cheap digital/analog convertor.

**DIGITAL I/O:**

Parallel digital I/O ports are one of the main features of microcontrollers. The number of I/O pins varies from 3-4 to over 90, depending on the controller family and the controller type.

**ANALOG I/O:**

Apart from a few small controllers, most controllers analog/digital convertors, which differ in the number of channels (2-16) and their resolution (8-12 bits). The analog module also generally features an analog comparator. In some cases, this microcontroller includes digital/analog convertors.

### 3.3.3 Synchronous Serial Communication

Synchronous serial transmission requires that the sender and receiver share a clock wise with one another, or that the sender provides a strobe or another timing signals so that the receiver known to “read” the next bit of data.

Synchronous serial communication describes a serial communication in which "data is sent in a continuous stream at a constant rate." Synchronous communication requires that the clocks in the transmitting and receiving devices are synchronized – running at the same rate – so the receiver can sample the signal at the same time intervals used by the Transmitter. No start or stop bits are required. For this reason "synchronous communication permits more information to be passed over a circuit per unit time" than asynchronous serial communication. Over time the transmitting and receiving clocks will tend to drift apart, requiring resynchronization.

### 3.3.4 Asynchronous Serial Communication

Asynchronous communication describes a serial communication protocol in which a start signal is sent prior to each byte, character or code word and a stop signal is sent after each code word, so as to make the communication asynchronous. The start signal serves to prepare the receiving mechanism for the reception and registration of a symbol and the stop signal serves to bring the receiving mechanism to rest in preparation for the reception of the next symbol.

The universal asynchronous receiver/Transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel
forms. Serial transmission of digital information (bits) through a single wire or other medium is less costly than parallel transmission through multiple wires.

The UART usually does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signaling levels.

When the receiver has received all of bits in the word, it may check for the parity bits (both sender and receiver must agree on whether a parity bit is to be used), and then the receiver looks for a stop bit. If the stop bit does not appear when it is supposed to, the UART considers the entire word to be garbled and will report a framing. Regardless of whether the data was received correctly or not, the UART automatically, these bits are not passed to the host. If another word is ready for transmission, the start bit for the new word can be sent as soon as the stop bit for the previous word has been sent.

Because asynchronous data is “self synchronizing”, if there is no data to transmit, the transmission line can be idle.

### 3.3.5 A Brief History of 8051

In 1981, Intel Corporation introduced an 8-bit microcontroller called 8051. This microcontroller had 128 bytes of RAM, 4K bytes of chip ROM, two timers, one serial port, and four ports all on a single chip. At the time it was also referred as “A SYSTEM ON A CHIP”.

The 8051 is an 8-bit processor meaning that the CPU can work only on 8 bits data at a time. Data larger than 8 bits has to be broken into 8 bits pieces to be processed by the CPU. The 8051 has a total of four I/O ports each 8 bit wide.

There are many versions of 8051 with different speeds and amount of on-chip ROM and they are all compatible with the original 8051. this means that if you write a program for one it will run on any of them.

The 8051 is an original member of the 8051 family. There are two other Members in the 8051 family of microcontrollers. They are 8052 and 8031. Microcontrollers will have the same internal architecture, but they differ in the following aspects.

- 8031 has 128 bytes of RAM, two timers and 6 interrupts.
- 8051 has 4K ROM, 128 bytes of RAM, two timers and 6 interrupts.
- 8052 has 8K ROM, 256 bytes of RAM, three timers and 8 interrupts.
Of the three microcontrollers, 8051 is the most preferable. Microcontroller supports both serial and parallel communication.

In the concerned project 8052 microcontroller is used. Here microcontroller used is AT89S52, which is manufactured by ATMEL laboratories.

3.3.6 Necessity of Microcontrollers:

Microprocessors brought the concept of programmable devices and made many applications of intelligent equipment. Most applications, which do not need large amount of data and program memory, tended to be costly.

The microprocessor system had to satisfy the data and program requirements so, sufficient RAM and ROM are used to satisfy most applications. The peripheral control equipment also had to be satisfied. Therefore, almost all-peripheral chips were used in the design. Because of these additional peripherals cost will be comparatively high.

An example:

8085 chip needs:

An Address latch for separating address from multiplex address and data. 32-KB RAM and 32-KB ROM to be able to satisfy most applications. As also Timer/Counter, Parallel programmable port, Serial port, and Interrupt controller are needed for its efficient applications.

In comparison a typical Micro controller 8051 chip has all that the 8051 board has except a reduced memory as follows.

4K bytes of ROM as compared to 32-KB, 128 Bytes of RAM as compared to 32-KB.

Bulky:

On comparing a board full of chips (Microprocessors) with one chip with all components in it (Microcontroller)

Debugging:

Lots of in Microprocessor circuitry and program to debug. In Micro controller there is no Microprocessor circuitry to debug.

Slower Development time: As we have observed Microprocessors need a lot of debugging at board level and at program level, whereas, Micro controller do not have the excessive circuitry and the built-in peripheral chips are easier to program for operation.

So peripheral devices like Timer/Counter, Parallel programmable port, Serial Communication Port, Interrupt controller and so on, which were most often used were
integrated with the Microprocessor to present the Micro controller. RAM and ROM also were integrated in the same chip. The ROM size was anything from 256 bytes to 32Kb or more. RAM was optimized to minimum of 64 bytes to 256 bytes or more. Microprocessor has following instructions to perform:

- Reading instructions or data from program memory ROM.
- Interpreting the instruction and executing it.
- Microprocessor Program is a collection of instructions stored in a Nonvolatile memory.
- Read Data from I/O device
- Process the input read, as per the instructions read in program memory.
- Read or write data to Data memory.
- Write data to I/O device and output the result of processing to O/P device.

3.3.7 Features of 8051 Microcontroller:
Compatible with MCS-51® Products

- 8K Bytes of In-System Programmable (ISP) Flash MemoryEndurance: 1000 Write/Erase Cycles
- Operating voltages:
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- 4 KB on chip program memory.
- 128 bytes on chip data memory(RAM)
- 3 internal and 2 external interrupts.
- Bit as well as byte addressable RAM area of 16 bytes.
• Four 8-bit ports, (short models have two 8-bit ports).
• PORT P1 (Pins 1 to 8)
• PORT P2 (pins 21 to 28)
• PORT P3 (Pins 10 to 17)
• PORT P0 (pins 32 to 39)
• 16-bit program counter and data pointer.
• 1 Microsecond instruction cycle with 12 MHz Crystal.

3.3.8 Pin Diagram:

Figure 3.3.2: Pin diagram of 8051 microcontroller

VCC- Digital supply voltage.
GND-Ground
Input/output ports:

One major feature of a microcontroller is the versatility built into the input/output (I/O) circuits that connect the 8051 to the outside world. The main constraint that limits numerous functions is the number of pins available in the 8051 circuit. The DIP had 40 pins and the success of the design depends on the flexibility incorporated into use of these pins. For this reason, 24 of the pins may each used for one of the two entirely different functions which depend, first, on what is physically connected to it and, then, on what software programs are used to “program” the pins.
**Port 0 (pins 32 to 39):**

Port 0 pins may serve as inputs, outputs, or, when used together, as a bi-directional low-order address and data bus for external memory. To configure a pin as input, 1 must be written into the corresponding port 0 latch by the program. When used for interfacing with the external memory, the lower byte of address is first sent via PORT0, latched using Address latch enable (ALE) pulse and then the bus is turned around to become the data bus for external memory.

**Port 1 (pins 1 to 8):**

Port 1 is exclusively used for input/output operations. PORTS 1 pin have no dual function. When a pin is to be configured as input, 1 is to be written into the corresponding Port 1 latch.

**Port 2 (pins 21 to 28):**

Port 2 may be used as an input/output port. It may also be used to supply a high–order address byte in conjunction with Port 0 low-order byte to address external memory. Port 2 pins are momentarily changed by the address control signals when supplying the high byte a 16-bit address. Port 2 latches remain stable when external memory is addressed, as they do not have to be turned around (set to 1) for data input as in the case for Port 0.

**Port 3 (pins 10 to 17):**

Port 3 may be used to input/output port. The input and output functions can be programmed under the control of the P3 latches or under the control of various special function registers. Unlike Port 0 and Port 2, which can have external addressing functions and change all eight-port b se, each pin of port 3 maybe individually programmed to be used as I/O or as one of the alternate functions. The Port 3 alternate uses are shown in the below table:

**Table 3.3.8.1: Alternate usage pins of port3**

<table>
<thead>
<tr>
<th>Pin (SFR)</th>
<th>Alternate Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3.0-RXD (SBUF)</td>
<td>Serial data input</td>
</tr>
<tr>
<td>P3.1-TXD (SBUF)</td>
<td>Serial data output</td>
</tr>
<tr>
<td>P3.2-INTO 0 (TCON.1)</td>
<td>External interrupt 0</td>
</tr>
<tr>
<td>P3.3 - INTO 1 (TCON.3)</td>
<td>External interrupt 1</td>
</tr>
<tr>
<td>P3.4 - T0 (TMOD)</td>
<td>External Timer 0 input</td>
</tr>
<tr>
<td>P3.5 – T1 (TMOD)</td>
<td>External timer 1 input</td>
</tr>
</tbody>
</table>
Pin 9:

PIN 9 is the reset pin which is used to reset the microcontroller’s internal registers and ports upon starting up. (Pin should be held high for 2 machine cycles.)

Pins 18 & 19:

The 8051 has a built-in oscillator amplifier hence we need to only connect a crystal at these pins to provide clock pulses to the circuit.

Pin 40 and 20:

Pins 40 and 20 are VCC and ground respectively. The 8051 chip needs +5V 500mA to function properly, although there are lower powered versions like the Atmel 2051 which is a scaled down version of the 8051 which runs on +3V.

Pins 29, 30 & 31:

As described in the features of the 8051, this chip contains a built-in flash memory. In order to program this we need to supply a voltage of +12V at pin 31. If external memory is connected then PIN 31, also called EA/VPP, should be connected to ground to indicate the presence of external memory. PIN 30 is called ALE (address latch enable), which is used when multiple memory chips are connected to the controller and only one of them needs to be selected. We will deal with this in depth in the later chapters. PIN 29 is called PSEN. This is "program store enable". In order to use the external memory it is required to provide the low voltage (0) on both PSEN and EA pins.

Interrupts:

The AT89S52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timers interrupts (Timers0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10. Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once. Note that Table 5 shows that bit position IE.6 is unimplemented.

| P3.6 - WR  | External memory write pulse |
| P3.7 - RD  | External memory read pulse  |
Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software. The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 3.3.8.2: shows about the interrupt sources

Therefore all the port pins are explained and hence their working will be done in different ways depending on user requirement as like of input port or output port.
Thus there are 8 interrupt pins in 8051 microcontroller those all are explained in detail in the above table.

3.3.9 Internal Architecture Of 8051 Microcontroller

**Figure 3.3.3:** shows internal architecture of 8051 microcontroller

3.3.10 Special Functional Register Memory (SFR)

Special Function Register (SFR) is the upper area of addressable memory, from address 0x80 to 0xFF. A, B, PSW, DPTR are called SFR. This area of memory cannot be used for data or program storage, but is instead a series of memory-mapped ports and registers. All port input and output can therefore be performed by memory move predations on specified addresses in the SFR. Also, different status registers are mapped into the SFR, for use in checking the status of the 8051, and changing some operational parameters of the 8051.
**PCON (power control):**

The power control SFR is used to control the 8051’s power control modes. Certain operation modes of the 8051 allow the 8051 to go into a type of “sleep mode” which consumes much less power.

<table>
<thead>
<tr>
<th>SMOD</th>
<th>--</th>
<th>--</th>
<th>GF1</th>
<th>GF0</th>
<th>PD</th>
<th>IDL</th>
</tr>
</thead>
</table>

*Figure 3.3.4: Power control Register*

**TCON (Timer control):**

The timer control SFR is used to configure and modify the way in which the 8051’s two timers operate. This SFR controls whether each of the two timers is running or stopped and contains a flag to indicate that each timer has overflowed. Additionally, some non-timer related bits are located in TCON SFR. These bits are used to configure the way in which the external interrupt flags are activated, which are set when an external interrupt occurs.

<table>
<thead>
<tr>
<th>TF1</th>
<th>TR1</th>
<th>TF0</th>
<th>TR0</th>
<th>IE1</th>
<th>IT1</th>
<th>IE0</th>
<th>ITO</th>
</tr>
</thead>
</table>

*Figure 3.3.5: Timer control Register*

**TMOD (Timer Mode):**

The timer mode SFR is used to configure the mode of operation of each of the two timers. Using this SFR your program may configure each timer to be a 16-bit timer, or 13 bit timer, 8-bit auto reload timer, or two separate timers. Additionally you may configure the timers to only count when an external pin is activated or to count “events” that are indicated on an external pin.
Figure 3.3.6: Timer mode Register

i) TO (Timer 0 low/high):

These two SFRs taken together represent timer 0. Their exact behavior depends on how the timer is configured in the TMOD SFR; however, these timers always count up. What is configurable is how and when they increment in value.

j) T1 (Timer 1 Low/High):

These two SFRs, taken together, represent timer 1. Their exact behavior depends on how the timer is configured in the TMOD SFR; however, these timers always count up.

IP (Interrupt Priority):

The interrupt priority SFR is used to specify the relative priority of each interrupt. On 8051, an interrupt maybe either low or high priority. An interrupt may interrupt interrupts. For e.g., if we configure all interrupts as low priority other than serial interrupt. The serial interrupt always interrupts the system, even if another interrupt is currently executing. However, if a serial interrupt is executing no other interrupt will be able to interrupt the serial interrupt routine since the serial interrupt routine has the highest priority.

3.4 LM358 Comparator

The LM358 IC is a great, low power and easy to use dual channel op-amp IC. It is designed and introduced by national semiconductor. It consists of two internally frequency compensated, high gain, independent op-amps. This IC is designed for specially to operate from a single power supply over a wide range of voltages. The LM358 IC is available in a chip sized package and applications of this op amp include conventional op-amp circuits, DC gain blocks and transducer amplifiers. LM358 IC is a good, standard operational amplifier and it is suitable for your needs. It can handle 3-32V DC supply & source up to 20mA per channel. This op-amp is apt, if you want
to operate two separate op-amps for a single power supply. It’s available in an 8-pin DIP package.

![Image of LM358 comparator](image)

**Figure 3.4: LM358 comparator**

### 3.4.1 Features of LM358:

- It consists of two op-amps internally and frequency compensated for unity gain.
- The large voltage gain is 100 dB.
- Wide bandwidth is 1MHz.
- Range of wide power supplies includes single and dual power supplies.
- Range of Single power supply is from 3V to 32V.
- Range of dual power supplies is from + or -1.5V to + or -16V.
- The supply current drain is very low, i.e., 500 μA.
- 2mV low i/p offset voltage.
- Common mode i/p voltage range comprises ground.
- The power supply voltage and differential i/p voltages are similar.
- o/p voltage swing is large.

### 3.5 INFRARED SENSORS:

An infrared sensor is an electronic device that detects infrared radiation in order to sense some aspect of its surroundings. Infrared sensors can measure the heat of an object.

Infrared light is electromagnetic radiation with longer wavelengths than those of visible light, extending from the nominal red edge of the visible spectrum at 700nm to 0.1mm. This range of wavelengths corresponds to a frequency range of approximately 430 THz down to 300 GHz. In a motion detector, a emitter emits IR rays which gets reflected from obstacles on its way and the IR rays reaches the sensor itself at the center of the device. This part may be composed of more than one
individual sensor, each of them being made from piezoelectric materials, whether natural or artificial. These are materials that generate an electrical voltage when heated or cooled.

These pyre electric materials are integrated into a small circuit board. They are wired in such a way so that when the sensor detects an increase in the heat of a small part of its field of view, it will trigger the motion detector's alarm. It is very common for an infrared sensor to be integrated into motion detectors like those used as part of a residential or commercial security system.

Most motion detectors are fitted with a special type of lens, called a Fresnel lens, on the sensor face. A set of these lenses on a motion detector can focus light from many directions, giving the sensor a view of the whole area. Instead of Fresnel lenses, some motion detectors are fitted with small parabolic mirrors which serve the same purpose.

![Figure 3.5: Infrared sensor](image)

An infrared sensor can be thought of as a camera that briefly remembers how an area's infrared radiation appears. A sudden change in one area of the field of view, especially one that moves, will change the way electricity goes from the pyroelectric materials through the rest of the circuit. This will trigger the motion detector to activate an alarm. If the whole field of view changes temperature, this will not trigger the device. This makes it so that sudden flashes of light and natural changes in temperature do not activate the sensor and cause false alarms.

Infrared motion detectors used in residential security systems are also desensitized somewhat, with the goal of preventing false alarms. Typically, a motion detector like these will not register movement by any object weighing less than 40 pounds (18 kg). With this modification, household pets will be able to move freely around the house without their owners needing to worry about a false alarm. For
households with large pets, sensors with an 80-pound (36 kg) allowance are also made.

The robot used in this project will have two IR sensor elements; each will be mounted on left and right of the robot. Figure 5 below shows the positions of the two sensors.

3.6 ALPHA-NUMERIC LCD DISPLAY:

![Typical 16x2 LCD](image)

**Fig 3.6: Typical 16x2 LCD**

A liquid crystal display is a thin, flat electronic visual display that uses the light modulating properties of liquid crystals (LCs). LCs do not emit light directly. In liquid crystal displays (LCDs) of liquid crystal technology is the most common applications. An advanced VGA computer screen from the pervasive wrist watch and pocket calculator, this type of display has evolved into an important and ambidextrous interface.

Consist of a liquid crystal display, an array of tiny segments (called pixels) and to present the information that can be manipulated. This basic common idea is to all displays; alienate from simple calculators to a full color LCD television.

The primary factor was size, an LCD consisting of primarily with some liquid crystal material between them of two glass plates. There is no bulk amount picture tube. This gives LCDs practical for applications where size (as well as weight) is necessary.

In general, LCDs uses very low power than the cathode-ray tube (CRT) counterparts. Many LCDs are ruminative, means that they use only atmosphere light to illuminate
the display. Even displays that do consume much less power than CRT devices require an external light source (i.e. computer displays).
CHAPTER 4
SOFTWARE IMPLEMENTATIONS

4.1 KEIL SOFTWARE

4.1.1 Introduction to Micro vision Keil (IDE)

This tutorial will assist you in writing your first 8051 Assembly language program using the popular Keil Compiler. Keil offers an evaluation package that will allow the assembly and debugging of files 2K or less. This package is freely available at their web site. Keil’s website address is www.keil.com. The sample program included in the tutorial toggles Ports 1 and 2 on the 8051. The compiled program has been tested using the 8051 board from MicroDigitalEd.com. The program also works with other systems that have Port 1 and 2 available. Keil is a cross compiler. So first we have to understand the concept of compilers and cross compilers. After then we shall learn how to work with keil.

4.1.2 Basic Keil Tutorial

1. Open Keil from the Start menu
2. The Figure below shows the basic names of the windows referred in this document.

4.1.3 Starting a new Assembler Project

1. Select New μ Vision Project from the Project Menu.
2. Name the project ‘Toggle’
3. Click on the Save Button.
4. The device window will be displayed.
5. Select the part you will be using to test with. For now we will use the Dallas Semiconductor part DS89C450.
6. Double Click on the Dallas Semiconductor
7. Scroll down and select the DS89C450 Part
8. Click OK

9. Choose No.
Creating Source File

1. Click File Menu and select New.
2. A new window will open up

1. A new window will open up in the Keil IDE.
2. Copy the example to the right into the new window. This file will toggle Ports 1 and 2 with a delay.
3. Click on File menu and select Save As…
4. Name the file Toggle.a51
5. Click the Save Button.

6. Change file type to ASM Source File (*.a; *.src).
7. Select toggle.a51
8. Click Add button
9. Click Close button.
11. Expand the Source Group 1 in the tree menu to ensure that the file was added to the project.

**Creating HEX for the Part**

1. Click on Target 1 in Tree menu
2. Click on Project Menu and select Options for Target 1
3. Select Target Tab.
4. Change Xtal (MHz) from 33.0 to 11.0592.
5. Select Output Tab
6. Click on Create Hex File check box
7. Click OK Button.

8. Click on Project Menu and select Rebuild all Target Files
9. In the Build Window it should report ‘0 Errors (s), 0 Warnings’
10. You are now ready to Program your Part
Running the Keil Debugger

1. The Keil Debugger should be now running.
2. Click on Peripherals. Select I/O Ports, Select Port1.

1. Step through the code by pressing F11 on the Keyboard.
2. The Parallel Port 1 Box should change as you completely step through the code.
3. To exit out, Click on Debug Menu and Select Start/Stop Debug Session
4.2 MICRO C FLASH

The programs compiled with MICRO C operate under the strict control of the runtime engine. This means only correct instructions are carried out, keeping the high level of reliability peculiar SUPER FLASH. Since it does not produce a machine code, the programs do not have to be recompiled to be used on other platforms.

Below are some of the possibilities offered by MICRO C:

- Execution of calculations in floating point
- Execution of calculations which involve trigonometric functions
- Development of communication protocols
- Runtime modification of the characteristics of the Variables
- Deferred recording of Trends
- Deferred recording of Alarms
- Importation, processing, exportation of the data generated by the applications (Trend, Alarms, Recipes, etc.)
- Carrying out management of fully customized files
- Carrying out completely free control functions
- Implementation of control functions for the input data Implementation of general control functions of data coherence
- Reduction in the SUPER FLASH Variables needed for an application
- Possibility of protecting your know-how
- Making processing drivers seen by the system as a normal peripheral

4.3 CODING

Main:

```
#include<reg51.h>

#include<functions.h>

#include<stdio.h>

long int VisitorsCount,VisitorsCount1;

sbit Income=P1^0;

sbit OutGoing=P1^1;
```
void main()
{
    IP=0xc5;
    IE=0x80;

    VisitorsCount=0;
    VisitorsCount1=0;
    InitialiseVFD();
    ClearDisplay();
    DisplayStringOnVFD("Total   :",0x80,10);
    DisplayStringOnVFD("Visitors:",0xc0,10);

    while(1)
    {

        if(Income==0){SendDataToVFD(0xcf,0,10);SendDataToVFD('+',1,10);do{}while(Income==0);VisitorsCount++;VisitorsCount1++;SendDataToVFD(0xcf,0,10);SendDataToVFD(0x20,1,10);}

        if(OutGoing==0){SendDataToVFD(0xcf,0,10);SendDataToVFD('-',1,10);do{}while(OutGoing==0);if(VisitorsCount>0){VisitorsCount--;}SendDataToVFD(0xcf,0,10);SendDataToVFD(0x20,1,10);}

        ShowData(VisitorsCount);ShowData1(VisitorsCount1);
    }
}
void ShowData(long int Temp)
{
    char word[8];
    int i;
    for(i=1;i<=6;i++)
    {
        word[i]=(Temp%10)|0x30;
        Temp=Temp/10;
    }
    SendDataToVFD(0xc9,0,10);
    for(i=6;i>0;i--)
    {
        SendDataToVFD(word[i],1,10);
    }
}

void ShowData1(long int Temp)
{
    char word[8];
    int i;
    for(i=1;i<=6;i++)
    {
        word[i]=(Temp%10)|0x30;
    }

Temp=Temp/10;

SendDataToVFD(0x89,0,10);

for(i=6;i>0;i--)
{
    SendDataToVFD(word[i],1,10);
}

Visitor function display:

#include<reg51.h>
#include<functions.h>

sbit VFD=P3^7;
sbit VFDregsel=P3^6;

void SendDataToVFD(char DataWord, bit CmdOrData,int Delay)
{
    Delay=200;
    P0=DataWord;
    VFDregsel=CmdOrData;
    VFD=1;
    while(Delay>0)
    {
        Delay--;
    }
    VFD=0;
}
void ClearDisplay()
{
    SendDataToVFD(0x01,0,10);
}

void InitialiseVFD(void)
{
    SendDataToVFD(0x38,0,100);
    SendDataToVFD(0x0c,0,100);
    SendDataToVFD(0x01,0,100);
    SendDataToVFD(0x06,0,100);
}

void DisplayStringOnVFD(char *String, char Address, int Delay)
{
    SendDataToVFD(Address,0,100);
    while(*String)
    {
        SendDataToVFD(*String++,1,Delay);
    }
}
CHAPTER 5

WORKING PROCEDURE

When the system is powered, the compiler initially initializes the stack pointer and all other variables. It then scans the input ports (PortP1.0 first). In the meantime, when there is no interruption between the IR LED and the photo transistor of the first sensor pair, the output of the photo transistor is always at low voltage. In other words port P1.0 is at logic low level. Now when a transition takes place, i.e. a logic high level is received at port P1.0, the compiler sees this as an interruption to sense the passage of a person or an object between the IR LED and the phototransistor. As per the program, the count value is increased and this value is displayed on the 2-Digit Counter. Now the compiler starts scanning the other input pin-P1.1. Similar to the first sensor pair, for this sensor pair also the phototransistor conducts in absence of any interruption and P1.1 is at logic low level. In case of an interruption, the pin P1.1 goes high and this interruption is perceived by decreasing the value of count.
5.1 BEFORE EXECUTION:
5.2 AFTER EXECUTION:
CONCLUSION

This project describes an easy model for visitor counter without human intervention. By using the IR sensors the IR receiver can detect the IR rays from the IR transmitter, then microcontroller increments or decrements the count of visitors.

Circuit is implemented in the Kiel software and implemented on the microcontroller. The performance has been verified both in software and hardware design. The total circuit is completely verified functionally and is following the application software. It can be concluded that the design implemented in the present work provide portability, flexibility, and the data transmission is also done with low power consumption.
FUTURE ENHANCEMENT

The advent of new high-speed technology and the growing computer capacity provided realistic opportunity for finding the number of visitors presented in an auditorium or a stadium. In bidirectional visitor counter the voice alarm may be added to indicate room is full and person can’t enter in the room.
REFERENCES


