CHAPTER-1
INTRODUCTION

In day to day life with the increasing civilization people visit many public places such as shopping malls, conference halls, etc. As number of people visiting these places is increasing, it leads to congestion in the place and gets uneasy for the people in enjoying the services provided by that particular place. If the number of people inside a place is controlled then it may help the people in enjoying the services to the most.

In order to overcome this we need a system which can help in improvising the maintenance of the system such that it attracts the customers to visit again. The system should be designed in such a way that it provides information regarding the number of people inside the place which can help in preventing large number of people entering the arena at the same time. This can be done by implementing sensors at entrance and exit doors which can help us in maintaining the number of people inside such that we can allow people till the maximum capacity is reached. If this is to be done using manpower this becomes difficult to record the number of people entered as well as leaving. This system helps us in reducing the manpower and help in maintaining the system to provide utmost comfort to the public using that place's facilities.

In this project we use two push buttons, one at the entrance portal and the other at the exit portal which are connected to registers which record the number of people entered and exit. When the push button at the entrance is pressed the value increases that a person has entered, similarly when the push button at exit is pressed the value decreases by one that a person has left and a place for another person is there to enter the area.

CHAPTER-2

BLOCK DIAGRAM AND WORKING PRINCIPLE

2.1 BLOCK DIAGRAM

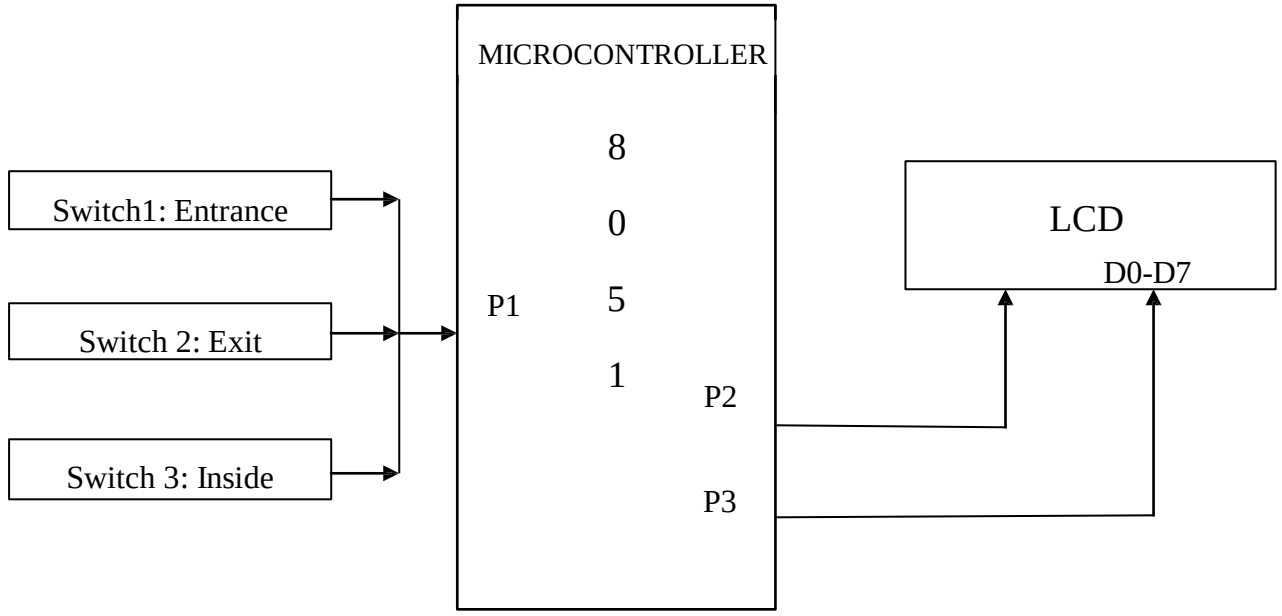

Figure 2.1: Block Diagram of the Project

2.2 WORKING PRINCIPLE

The person entering or moving out of a shopping mall or an auditorium and the persons present inside the room can be known by the switches as the switches are connected to the microcontroller (AT89C51) and the microcontroller is connected to the LCD. The persons coming inside and moving out presses the switch and the count increases on the LCD screen and the total no of persons inside the room is also displayed accordingly

CHAPTER-3

HARDWARE MODULES

MICROCONTROLLER-AT89C51:

FEATURES:

• 4K Bytes of In-System Reprogrammable Flash Memory – Endurance: 1,000 Write/Erase Cycles

• Fully Static Operation: 0 Hz to 24 MHz

• 128 x 8-bit Internal RAM

• 32 Programmable I/O Lines

• Two 16-bit Timer/Counters

• Six Interrupt Sources

• Programmable Serial Channel

DESCRIPTION:

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density non-volatile memory technology. The on-chip flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

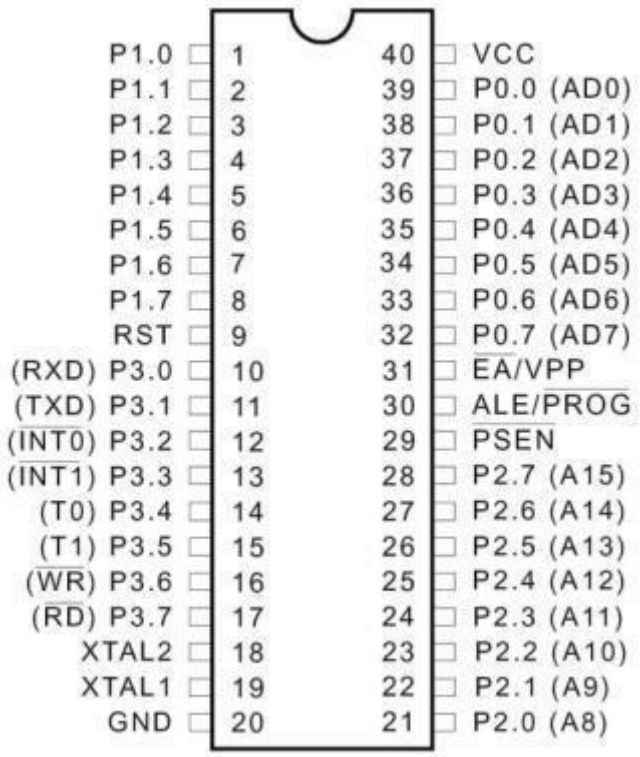

FIG-3.1: PIN DIAGRAM OF AT89C51

PIN DESCRIPTION:

PIN 9: PIN 9 is the reset pin which is used reset the microcontroller‟s internal registers and ports upon starting up. A high on this pin for two machine cycles while the oscillator is running resets the device.

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 builtin 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.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.

There are 4 8-bit ports: P0, P1, P2 and P3.

PORT P1 (Pins 1 to 8): Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pullups.

PORT P2 (pins 21 to 28): Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pullups.

Port 2 can also be used as a general purpose 8 bit port when no external memory is present, but if external memory access is required then Port 2 will act as an address bus in conjunction with PORT P0 to access external memory. Port 2 acts as A8-A15.

PORT P3 (pins 10-17): Port 3 is an 8-bit bi-directional I/O port with internal pullups. 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 pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below: P3.0- RXD (serial input port)

P3.1 -TXD (serial output 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)

PORT P0 (pins 32 to 39): 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 P0 can be used as a general purpose 8 bit port when no external memory is present, but if external memory access is required then PORT P0 acts as a multiplexed address and data bus that can be used to access external memory in conjunction with PORT P2. P0 acts as AD0-AD7.

LCD INTERFACING:

The three main control lines for LCD are referred to as **EN**, **RS**, and **RW**.

The **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) and then set the other two control lines and/or put data on the data bus. When the other lines are completely 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.

The **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 sould be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

The **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. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Pin Descriptions for LCD

Pin	Symbol	I/O	Description
1	VSS	-	Ground
2	VCC	-	+5V power supply
3	VEE	-	Power supply to control contrast
4	RS	I	RS=0 to select command register, RS=1 to select data register.
5	R/W	I	R/W=0 for write, R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8 bit data bus
8	DB1	I/O	The 8 bit data bus
9	DB2	I/O	The 8 bit data bus
10	DB3	I/O	The 8 bit data bus
11	DB4	I/O	The 8 bit data bus
12	DB5	I/O	The 8 bit data bus
13	DB6	I/O	The 8 bit data bus
14	DB7	I/O	The 8 bit data bus

LCD Command Codes:

Code (Hex)	Command to LCD Instruction Register
1	Clear Display screen
2	Return home
4	Decrement cursor (Shift cursor to left)
6	Increment cursor (Shift cursor to Right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display on, cursor off
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning of 1st line
0C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 Matrix

INITIALIZING THE LCD

Before you may really use the LCD, you must initialize and configure it. This is accomplished by sending a number of initialization instructions to the LCD.The first instruction we send must tell the LCD whether we'll be communicating with it with an 8-bit or 4-bit data bus. We also select a 5x8 dot character font. These two options are selected by sending the command 38h to the LCD as a command. As you will recall from the last section, we mentioned that the **RS** line must be low if we are sending a command to the LCD. Thus, to send this 38h command to the LCD we must execute the following 8051 instructions:

MOV DATA,#38h

We've now sent the first byte of the initialization sequence. The second byte of the initialization sequence is the instruction 0Eh. Thus we must repeat the initialization code from above, but now with the instruction. Thus the the next code segment is:

MOV DATA,#0Eh

The last byte we need to send is used to configure additional operational parameters of the LCD. We must send the value 06h.

MOV DATA,#06h

CLEARING THE DISPLAY

When the LCD is first initialized, the screen should automatically be cleared by the 44780 controller. However, it's always a good idea to do things yourself so that you can be completely sure that the display is the way you want it. Thus, it's not a bad idea to clear the screen as the very first opreation after the LCD has been initialiezd.An LCD command exists to accomplish this function. Not suprisingly, it is the command 01h. Since clearing the screen is a function we very likely will wish to call more than once, it's a good idea to make it a subroutine:

MOV DATA,#01h

LCD Display

LCD displays the number of persons present in the arena/place. When a person enters into the place then the register value is incremented and the resultant value is displayed in the display. Similarly when a person exits the place the value of the register is modifies and the resultant value is displayed over the LCD.

Buttons

This project needs 3 buttons. By pressing the first button the number of persons entering value is incremented, in parallel the number of persons inside is also incremented. By pressing the second button the number of persons leaving is incremented and in parallel the persons remaining inside is decremented. By pressing the third button the persons remaining inside is displayed.

CHAPTER-4

PROJECT IMPLEMENTATION

4.1 ALGORITHM

- Here we are using AT89C51 (8051) microcontroller.
- The switches and the LCD are interfaced with the given microcontroller.
- Data bus pins of the LCD (D0-D7) are connected to the port 3 of the microcontroller(p3.0-p3.7)
- The switches are connected to port 1 of the microcontroller(p1.1-p1.3) and the ground.
- here we use 3 registers namely R0,R1,R2 for the persons coming in ,out and the total number of persons present inside.

4.2 FLOWCHART

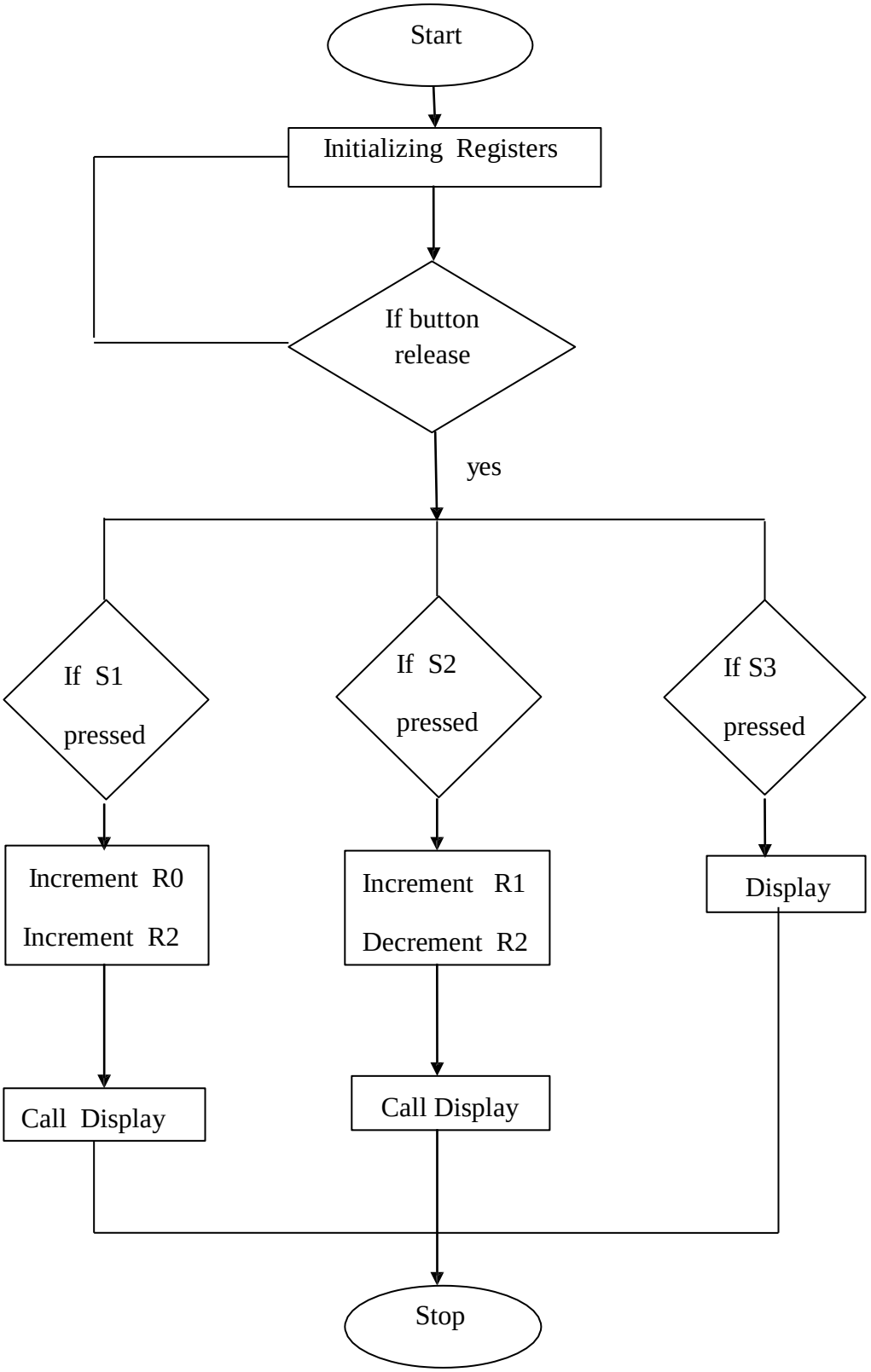

Fig 4.1: Flowchart

4.3 CIRCUIT DIAGRAM

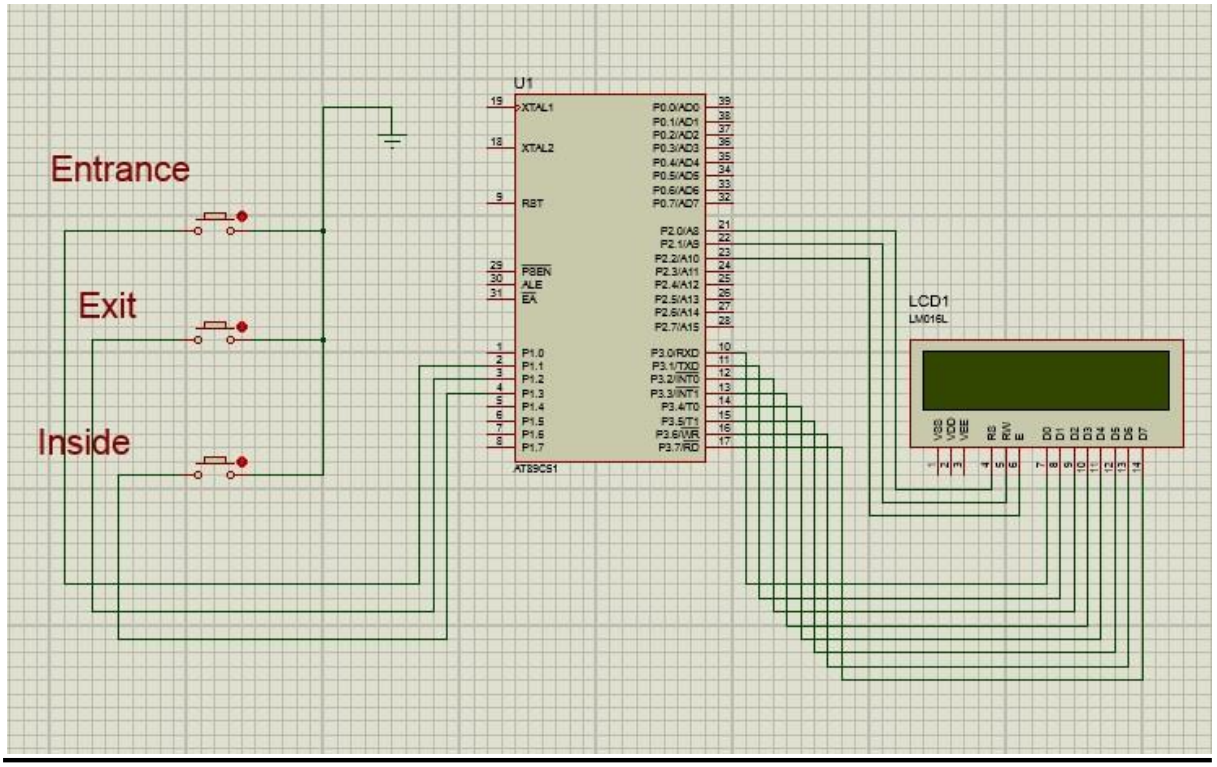

Fig 4.2 Circuit Diagram

Start:

First we Initialize the registers r0, r1, r2 where r0 holds the number of persons entering the place,r1 holds the number of persons leaving the place and r3 holds the total number of persons present inside the arena. It is required to check whether any switch is pressed or not in order to verify if there is an input .If any switch is pressed then check for the release of it because till it is released no operation should be performed on the registers because it may lead to incorrect values.

Step1: SWITCH 1:

When the switch one is pressed the condition is checked until that switch is released or not and then the values of the register 1 and 3 (r0, r1) is incremented by 1.

After incrementing the values in the register, LCD function is called so that we could display the message that IN: OUT: in the second line of that 16*2 LCD it displays the incremented values that the number of persons entered through the door.

Step2: SWITCH2:

When the switch two is pressed the condition is checked until that switch is released or not. Then the value of the register 2 (r2) is incremented by 1 and the value of the register 3(r3) is decremented by 1.

After incrementing the values in the register, LCD function is called so that we could display the message that IN: OUT:in the second line of that 16*2 LCD it displays the incremented values of number of persons left outside the door.

Step3: SWITCH 3:

When the switch three is pressed the condition is checked until that switch is released or not. Then the value which are present in the register 3(r3) are sent to the LCD by calling the LCD function .this displays the message that number of persons present inside the room.

Stop:

At the end of the day when the system is stopped then all the register values are initialized to zero.

CHAPTER-5

ADVANTAGES AND DISADVANTAGES

5.1 ADVANTAGES

1. It is of Low cost
2. It is very easy to use
3. It can be implemented to a single door

5.2 DISADVANTAGES

It is used only when one single person cuts the rays of the sensor hence it cannot be used when two person cross simultaneously.

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The main aim of this project is to count the number of persons inside the place. If a person passes through the door the register value is incremented by one and also the register value which reflects the number of persons inside is also incremented and if a person comes out of the room, register value is incremented by one and the register value which reflects the number of persons inside is decremented by one. These both register values are displayed on the LCD and also the number of people inside is also displayed on the LCD.

6.2 FUTURE SCOPE

The Bidirectional Visitor Counting System can be used in wide range in future. It is helpful in many ways. Some Examples are:

- It can be used in Schools, Hospitals, Bus Stands, Railway Stations, Banks,

  Police Stations, Lifts, other public places.
- It can be used in Parking places.
- Lights can be turned ON/OFF according to the number of people in the room.
- We can check the ambient light intensity and then decide if the light needs to be turned ON or not.
- By modifying this circuit and using two relays we can achieve a task of opening and closing the door.

REFERENCES

1. Mazidi & Mazidi, "The 8051 Micro Controller and Embedded Systems using Assembly and c", Pearsons publication, 2nd edition.
2. Subrata Ghoshal, "8051 Micro Controller-Internals, Programming&Interfacing" Ghoshal
3. The 8051 microcontroller: KENNETH J. AYALA

APPENDIX-A

SOURCE CODE

PROGRAM

```
org 00h
MOV R0,#'0'
MOV R2,#'0'
MOV R1,#'0'
find1:
jnb p1.1,Enter
jnb p1.2,Exit
jnb p1.3,result
sjmp find1
Enter:
acall startlcd
mov dptr,#data2
acall strt
cjne r0,#'9',f
inc r2
acall value2
mov A,r2
acall datawrt
mov r0,#'0'
acall value1
mov A,r0
acall datawrt
sjmp find1
f:acall value2
mov A,r2
acall datawrt
inc r0
acall value1
```

```
MOV A,R0
ACALL DATAWRT
sjmp find1
find12: jmp find1
Exit:
acall startlcd
mov dptr,#data3
acall strt
cjne r1,#'9',ef
inc r3
acall value1
mov A,r3
acall datawrt
mov r1,#'0'
acall value2
mov A,r1
acall datawrt
ljmp find1
ef:acall value2
mov A,r3
acall datawrt
inc r1
acall value1
MOV A,R1
ACALL DATAWRT
sjmp find12
result:
acall startlcd
mov dptr,#data4
acall strt
loop: cjne r1,#'0',loop1
loop2:cjne r3,#'0',loop3
```

```
sjmp go
loop1: cjne r0,#'0',loop4
mov r0,#'9'
dec r1
dec r2
sjmp loop
loop4: dec r0
dec r1
sjmp loop
loop3: dec r2
dec r3
sjmp loop2
go: lcall value1
mov a,r0
lcall datawrt
lcall value2
mov a,r2
lcall datawrt
stop: jmp stop
strt:
CLR A
MOVC A,@A+DPTR
JZ aga
ACALL DATAWRT
inc dptr
sjmp strt
aga: RET
startlcd:
MOV A,#80H
ACALL COMNWRT
MOV A,#38H
ACALL COMNWRT
MOV A,#0EH
ACALL COMNWRT
MOV A,#06H
ACALL COMNWRT
RET
value1:
MOV A,#0C3H
ACALL COMNWRT
RET
```

```
    VALUE2:
    MOV A,#0C2H
    ACALL COMNWRT
    RET
    COMNWRT:
    MOV P3,A
    CLR P2.0
    CLR P2.1
    SETB P2.2
    ACALL DELAY
    CLR P2.2
    ACALL DELAY
    RET
    DATAWRT:
    MOV P3,A
    SETB P2.0
    CLR P2.1
    SETB P2.2
    ACALL DELAY
    CLR P2.2
    ACALL DELAY
    RET
    DELAY: MOV R5,#50
    HERE2: MOV R6,#200
    HERE:DJNZ R6,HERE
    DJNZ R5,HERE2
    RET
    org 200h
    data2: DB 'Persons Entered',0
    data3: DB 'Persons Left ',0
    data4: DB 'Total Persons inside',0
end
```