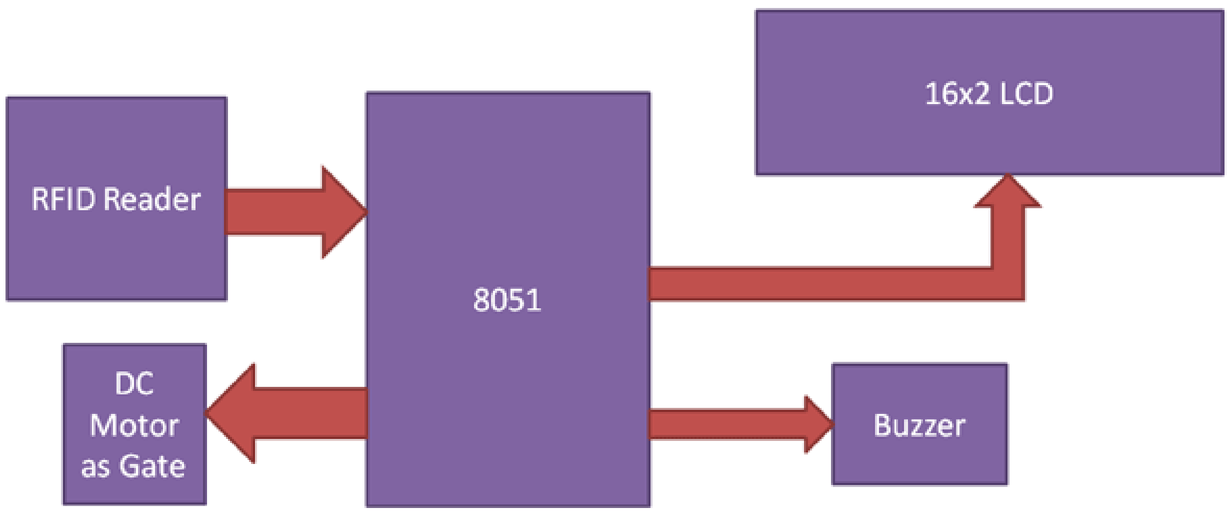
INTRODUCTION

Radio frequency identification (RFID) refers to the use of radio frequency wave to identify and track the tag implanted into an object or a living thing. It is a wireless mean of communication that use electromagnetic and electrostatic coupling in radio frequency portion of the spectrum to communicate between reader and tag through a variety of modulation and encoding scheme. Modulation refers to the variation in the amplitude, frequency or phase of a high frequency carrier signal to convey information. Encoding is a process of converting information from one format to another. RFID system usually consists of RFID reader and tag. It is very useful because it can uniquely identify a person or a product based on the tag incorporated. It can be done quickly and this usually takes less than a second. A prototype of the system has been designed and fabricated. The RFID reader used in the system is passive type which has maximum range of detection of around 5cm above the reader. It operates at frequency of 125 kHz and 12V power supply. The system has ability to uniquely identify and take ticket for persons. The users only need to place their RFID tag on the reader to take ticket.

Automated fare collection (AFC) systems are used in many urban public transport systems around the world. As the designation suggests, these are typically designed with the specific purpose of automating the ticketing system, easing public transport use for passengers and adding efficiency to revenue collection operations. In addition, AFC systems are used to enable integrated ticketing across different public trans-port modes and operators in urban areas. This chapter gives you an introduction about the Internet of Things and its real time applications. The main idea behind this project is to collect the fare automatically using the Internet of Things in a cost efficient manner. Internet of Things allows objects to sensed and controlled remotely across existing network infrastructure.

1. MATERIALS AND METHODS

The materials and methods are selected by designing the block diagram. The Block diagram -1 shows components required to design RFID based Automatic Bus Fare collection system using 89C51.



Components Required

1. 89S51 Microcontroller
2. RFID Reader module
3. 16 x 2 LCD
4. DC Motor driver L293D and DC Motor
5. Buzzer with a BC547 NPN transistor.

The RFID module is interfaced with 89S51 in serial communication mode using USART. The motor and buzzer are interfaced in I/O mode. The LCD is interfaced in 4 bit mode.

2. RFID READER INTERFACE

2.1 RFID Description:

Radio Frequency Identification or simply RFID is a wireless technology generally used for automatic identification and data collection. RFID technology is used for accessing data from a uniquely identify RFID card or tag by combining the radio frequency and microchip technologies i.e. the data is retrieved or stored into the RFID cards without making any physical contact.With the help of RFID technology we can create smart systems that can be used personnel identification, product monitoring, individual or organization security, transportation, maintenance of inventory and supply chain tracking**. RFID** is short for Radio Frequency Identification. Generally a RFID system consists of 2 parts. A Reader, and one Transponder, also known as Tag or RFID card. A typical RFID system is shown in the figure 3.1.

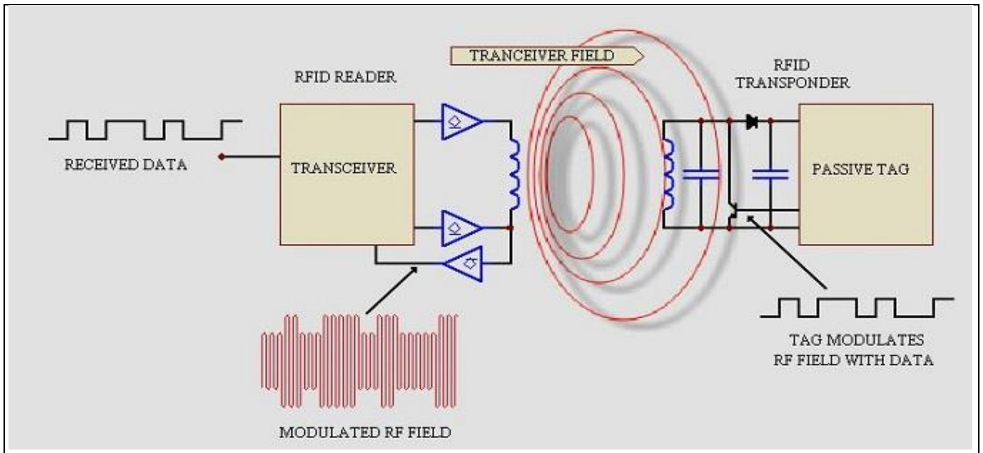


Figure 3.1 A typical RFID System

RFID Card or Tag consists of the data in the embedded microchip as shown in the figure. The tags are three types read, write and read/write tags classified in terms of how data can be encoded on tags. Here the discussion is limited to read only tag, figure 3.2 shows the Read only tag. Read-only tags contain data such as a serialized tracking number, which is pre-written onto them by the tag manufacturer or distributor. These are generally the least expensive tags because they cannot have any additional information included as they move throughout the supply chain. Any updates to that information would have to be maintained in the application software that tracks SKU movement and activity.

	
Figure 3.3 RFID TAGS internal structure	RFID Tag

The communication between the Reader and Card is using Radio Technology whereas the communication between RFID Reader and the host device like a microcontroller is through serial protocol.

RFID tags are generally classified into two types: Active RFID Tags and Passive RFID Tags.

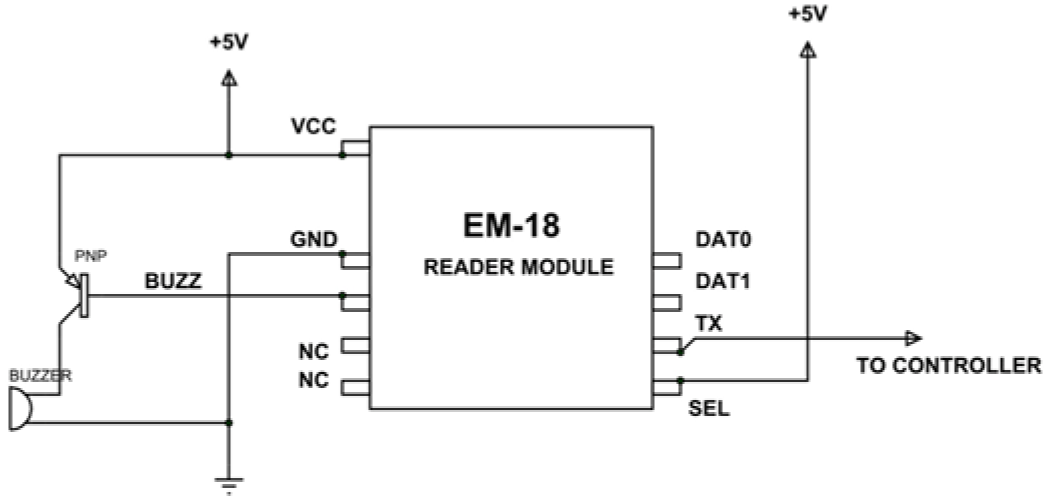
Active RFID Tags are battery powered and they have an electronically erasable and programmable memory. So, the necessary data can be written in to the memory by the user and read using a RFID Reader.

As they are battery powered devices, Active RFID Tags can communicate with the RFID Reader over distances larger than 100 feet.

Passive RFID Tags on the other hand, do not contain any battery. Passive RFID Tags consists of a coil and a microchip. The coil acts as an antenna as well as the source of power through induction.

The microchip is a read only type memory and the user can only read the data that already stored by the manufacturer. Passive RFID Tags are used for short distance communication, usually 10 feet or less.

The most commonly used RFID reader is EM-18 reader module as shown in the figure . This module has a built in antenna which is used to power the RFID Cards and also extract the information from the Card’s microchip.



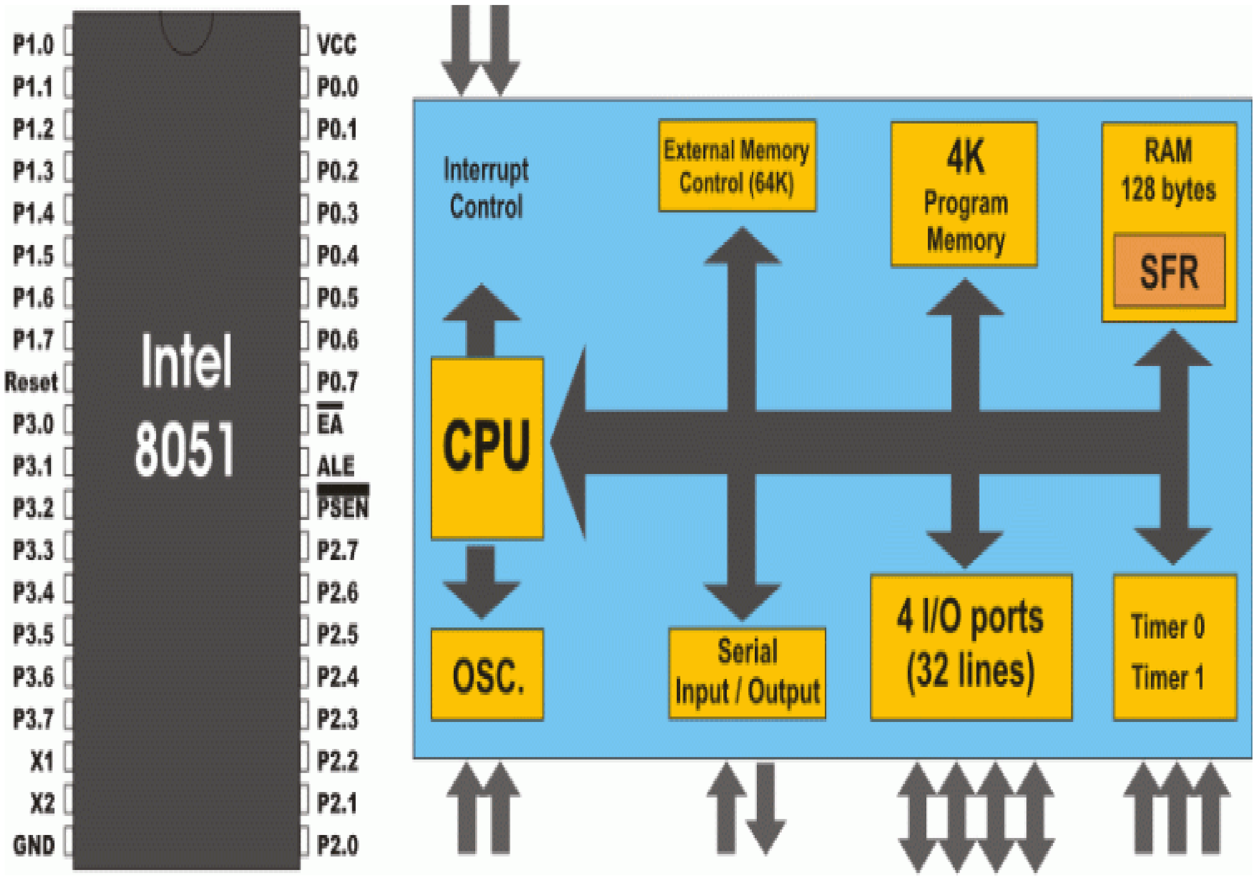
2.2 MAX232 IC:

RFID module cannot be connected directly with the 8051 microcontroller. For this purpose, MAX232 IC is used here. MAX232 can convert TTL/CMOS logic levels to RS232 logic levels for serial communication with the PC and CMOS/ TTL logic levels to RS 232 logic levels for serial communication with microcontroller. Actually, the microcontroller used here operates at TTL logic. The Tx pin output is connected to 89c51 pin 10 of Rx as shown in the figure.

When RFID tags come nearby RFID reader device, electromagnetic interrogation pulse gets triggered which in turn transmits the student's digital data to the reader. This advanced cloud based system ensures that students are under complete surveillance right from the time they enter the school premises. RFID walk through attendance gate for schools ensures that SMS is triggered automatically to the parents mobile once their ward passes through the Smart Gate Reader.

3. FEATURES OF AT 89C51

89C51 is an eight bit microcontroller with 40 pin DIP operating at 5V with operating frequency range from 10Mhz to 40Mhz. It internally contains CPU and supporting peripheral devices like Program memory, RAM, I/O Ports, Timer and Serial I/0 Port as shown in the fig.



It has the following features

It has 8 bit A register and B register to perform arithmetic operations

It has 16 bit Program counter and Data Pointer DPTR

It has 8 bit Stack Pointer (SP)

It has four 8-bit I/O ports named as Port 0, port1, port 2 and prot3

It has two16 bit timers Timer 0 and Timer 1

It has128 byte internal RAM

It has 4K byte internal ROM called flash memory

It has serial communication port Txd and Rxd pins with SFR SBUF, SCON, PCON

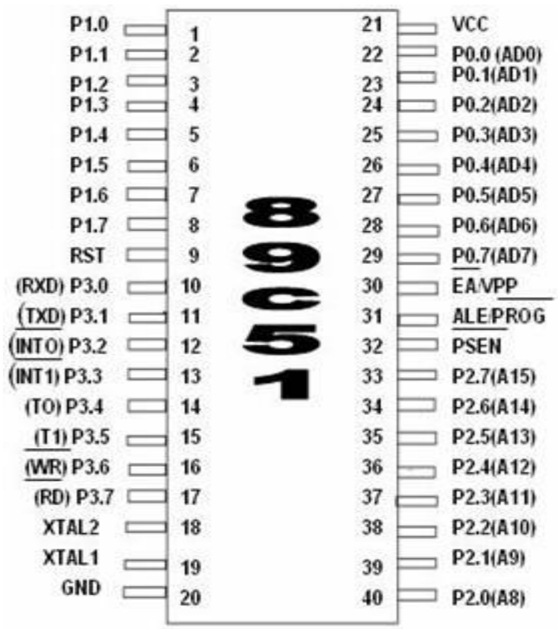
It has two external interrupts INT0 and INT1, RESET interrupts and 3 internal Interrupts T0, T1 and Serial communication interrupts.

The table shows the special features of 89C51 microcontroller

Part Number	AT89C51
ROM	4K
RAM	128 bytes
I/O Pins	32
Timers	2
Interrupts	6
Vcc	5 V
Packaging	40

4. PIN DETAILS OF 89C51

Hardware designer must know the pin details of the microcontroller to design an embedded system. 89C51 is a 40 pin device DIP as shown in the fig.



Port 0:is a dual-purpose port on pins 32-39 of the 89C51 IC. In minimum – component designs, it is used as a general purpose I/O Port. For larger designs with external memory, it becomes a multiplexed address and data bus AD0-AD7.

Port 1: is a dedicated I/O port on pins 1-8. The pins, designated as P1.0. P1.1. P1.2 etc. are available for interfacing to external devices as required. No alternate functions are assigned for Port 1 pins; thus they are used solely for interfacing to external devices.

Port 2:(pints 21-28) is a dual – purpose port serving as general purpose I/O, or as the high byte of the address bus A8-A15 for designs with external code memory or more than 256 bytes of external data memory.

Port 3:is a dual – purpose port on pins 10-17. As well as general – purpose I/O, these pins are multifunctional with each having an alternate purpose related to special features of the 89C51).

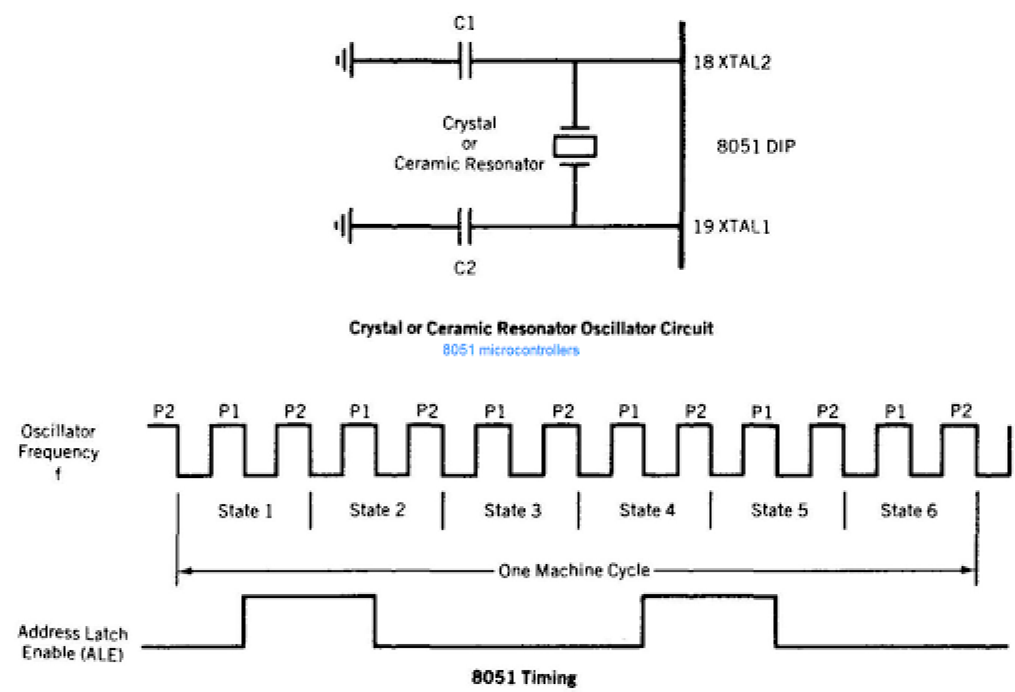
Now coming to the other pin functions.

PSEN: This is an output pin. PSEN stands for “program store enable.” In an 8031-based system in which an external ROM holds the program code, this pin is connected to the OE pin of the ROM.

VCC: Pin 40 provides supply voltage to the chip. The voltage source is +5V.

GND: Pin 20 is the Ground pin.

XTAL1 and XTAL2: The 8051 has an on-chip oscillator but requires an external clock to run it. Most often a quartz crystal oscillator is connected to inputs XTALI (pin 19) and XTAL2 (pin 18). The quartz crystal oscillator connected to XTAL1 AND XTAL2 also needs two capacitors of 30 pF value. One side of each capacitor is connected to the ground as shown in this figure;



Crystal or Ceramic Resonator Oscillator Circuit

8051 microcontrollers

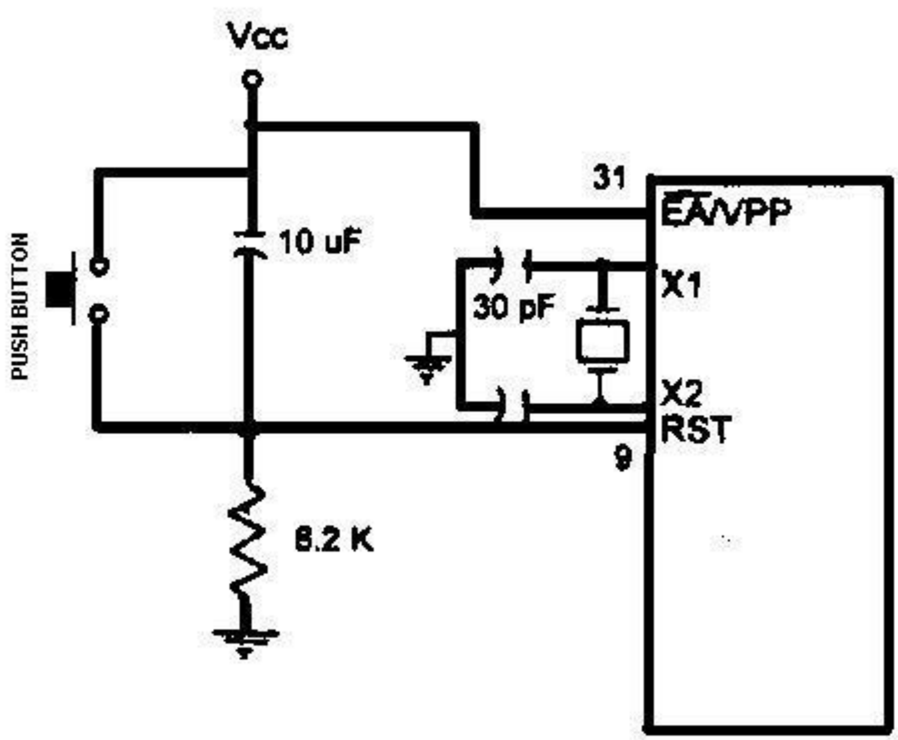
8051 Timing

Instruction time Tins= 12 x no of machine cycle / osc frequency

Tins = 12 x 1 / 11.0592Mhz = 1.085µsec.

EA: 89C51 come with on-chip ROM to store programs. In such cases, the EA pin is connected to VCC for giving power to save and erase program from the memory.

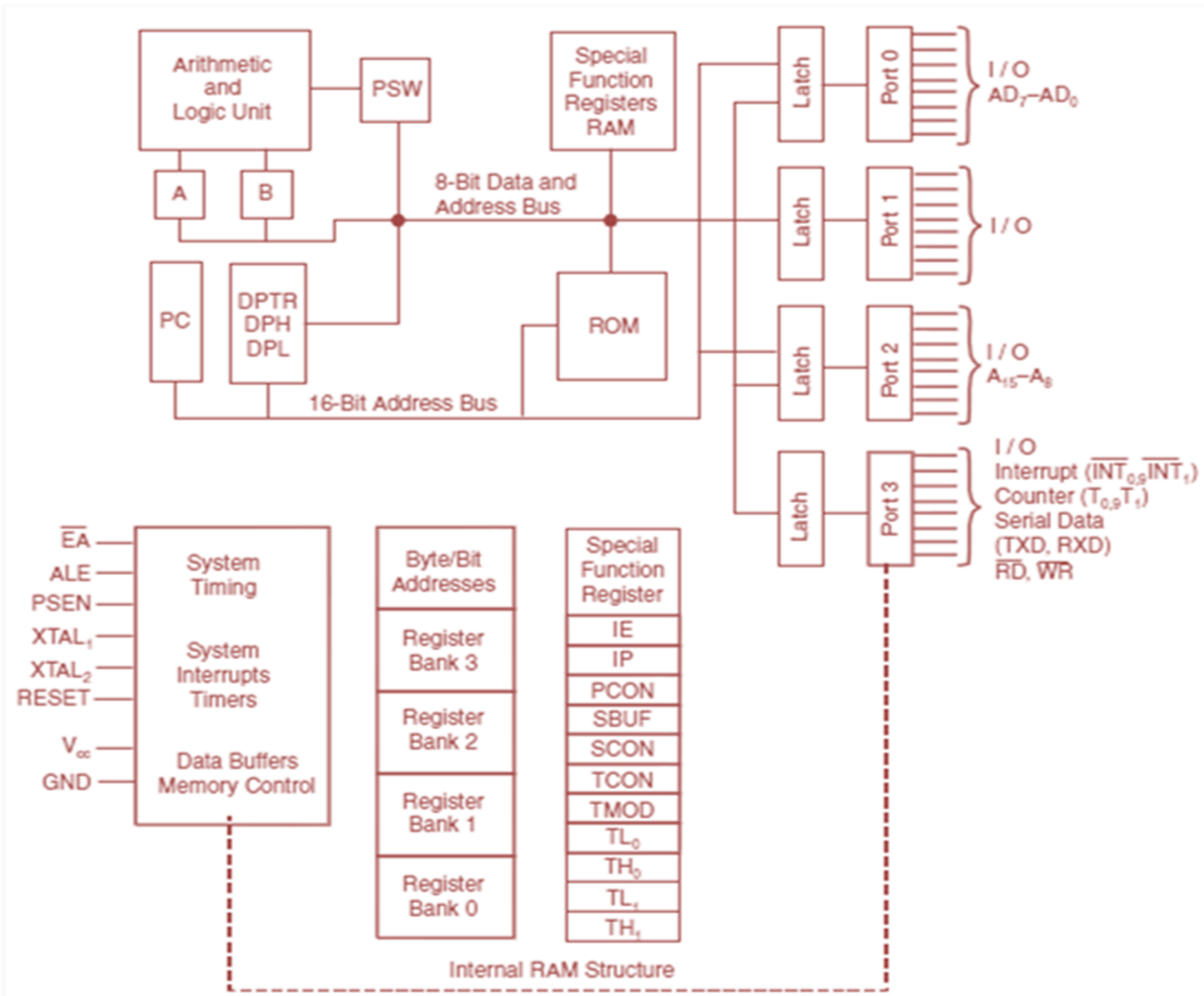
RST(RESET): The RST input on pin 9 is the master rest for the 89C51. When this signal is brought high for a least two machine cycles, the 89C51 internal registers are loaded with appropriate values for an orderly system start-up. For normal operation, RST is low. Figure shows permanent connections of Reset Pin.



ALE: (address latch enable) is an output pin and is active high. When connecting an 89C51 to external memory, port 0 provides both address and data. In other words, the 89C51 multiplexes address and data through port 0 to save pins. The ALE pin is used for de-multiplexing the address and data by connecting to the G pin of the 74LS373 chip.

5. PROGRAMMING MODEL OF 89C51

Programming model explains registers, which are handled by the programmer for various purposes to programming the 89C51 in assembly language and high-level language. Fig shows the programming model of 89C51 microcontroller.

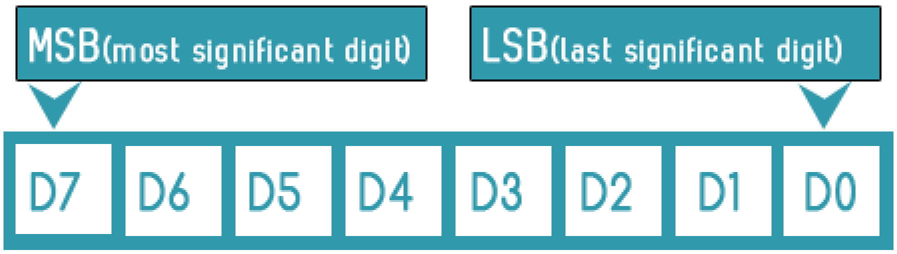


5.1 CPU Registers

The following registers are CPU registers

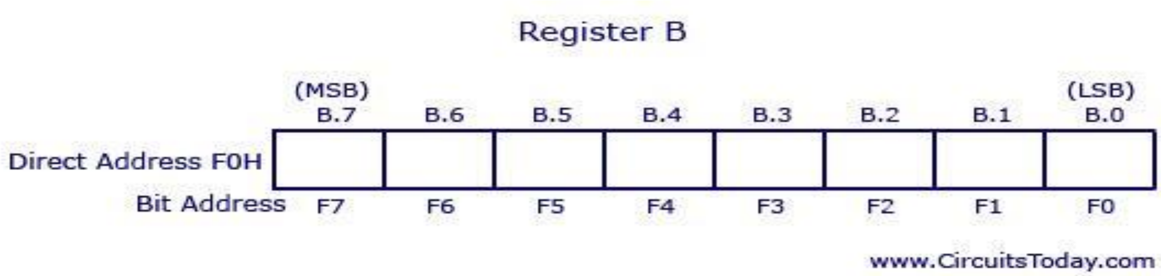
5.1.1 A Register

A register is an 8 bit register as shown in the fig. It is part of arithmetic and logical unit. Before performing any arithmetic and logical operations one of the operand must be in the A register. After performing arithmetic and logical operation the result is stored in the A register.



5.1.2 B registers

B register is an 8-bit register. It is used along with A register during byte by byte multiplication and division operation so it is also part of the arithmetic and logical unit. In multiplication operation, one of the operand is in 'A' register and the second operand is in B register after multiplication LSB value is stored in 'A' register and MSB value is stored in B register. During division numerator is stored in 'A' register and the denominator is stored in B register after division operation quotient is stored in 'A' register and the remainder is stored in B register.

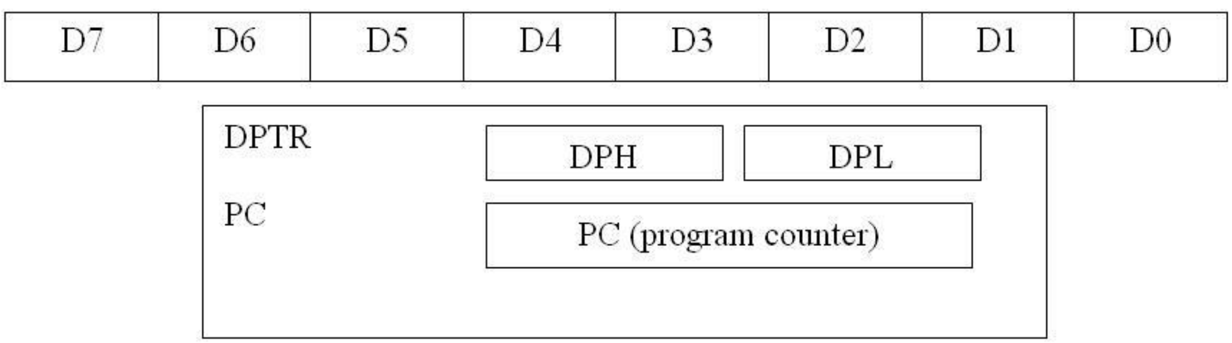


5.1.3 Program Counters PC

Program counter is 16-bit register to hold the program memory address. It keeps the track of the address during program execution. It can address 64Kb memory locations. It address internal as well as external memory whose address range is from 0000h – 0FFFh internal memory and 1000h – FFFFh external memory.

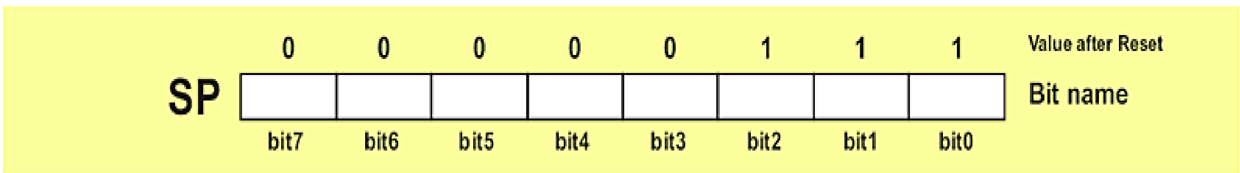
5.1.4 DPTR:

DPTR is also 16 bit register to point the external data memory. This is also one of the special function registers DPH and DPL. It can address 64Kb memory locations.



5.1.5 Stack Pointer (SP)

Stack pointer is one of the SFR which holds the stack memory address. When the system is reset SP has the address 0x07. Using PUSH and POP instruction the stack pointer is incremented and decremented it self to store the data and retrieve the data from stack memory



5.1.6 Program Status Word (PSW)

Program status word is an 8 bit special function register. it is used to reflect the condition of the result during and after performing the arithmetic and logical operations. It has four important flags to reflect the conditions. CY, AC, OV and P flags.

Bank Selection: By programming the PSW bits RS1 and RS0 the banks are selected.

RS1	RS0	Bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

Program status word(D0H)

CY	AC	F0	RS1	RS0	OV	--	P

CY—PSW.7	Carry flag
AC – PSW.6	Auxiliary carry
F0– PSW.5	General purpose
RS1-PSW.4	Register bank selector bit 1
RS0-PSW.3	Register bank selector bit 0
OV-PSW.2	Overflow flag
--	User defined bit
P—PSW.0	Parity flag

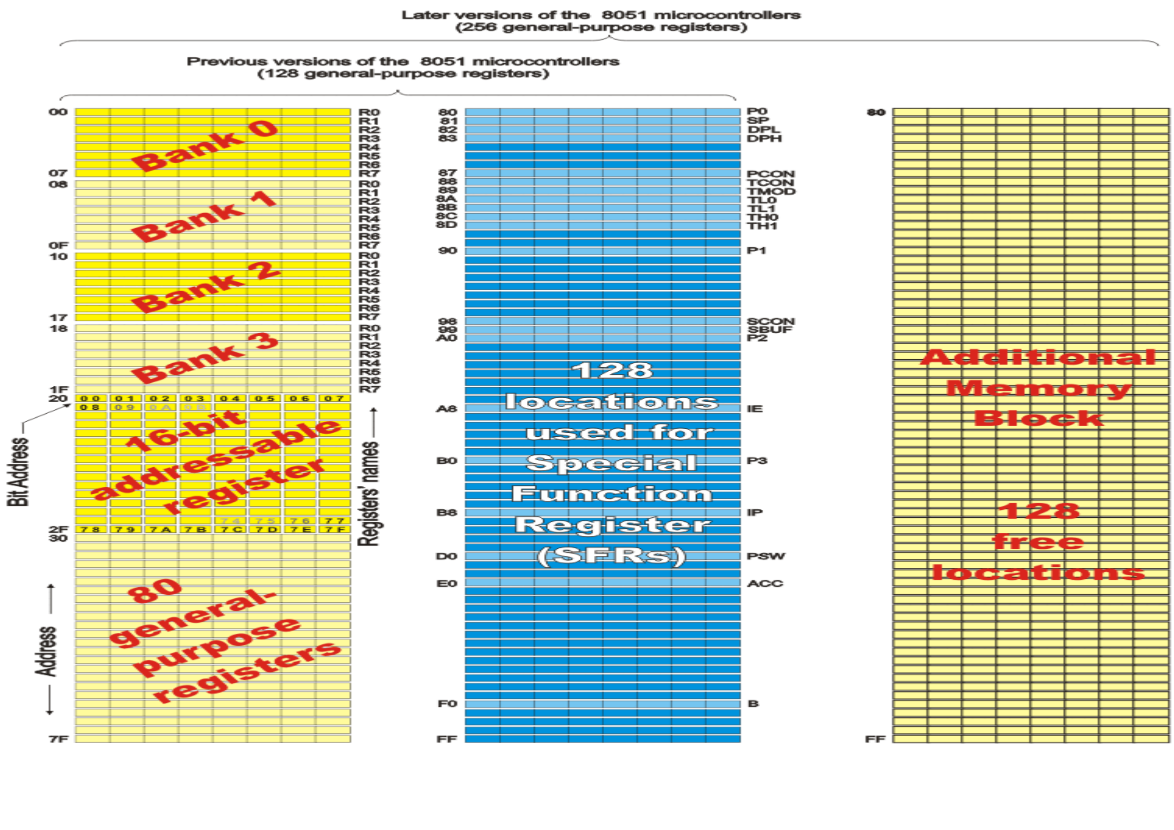
6. MEMORY ORGANIZATION

6.1 Internal ROM Flash Memory

89C51 has 4 Kb internal ROM flash memories. Its address range is from 0000h to 0FFFh. To use the AT89C51 to develop a microcontroller-based system requires a ROM burner that supports flash memory. Notice that in flash memory you must erase the entire contents of ROM in order to program it again. The PROM burner itself does this erasing of flash.

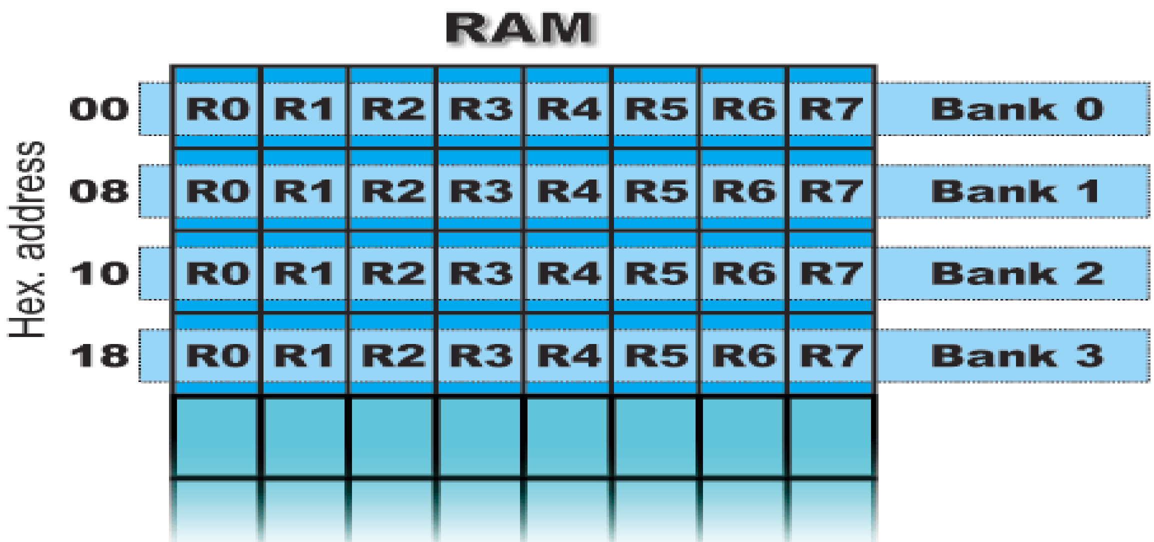
6.2 Internal RAM Memory

89C51 has 128-byte internal RAM memory. This 128 byte is divided into three parts Bank memory, bit addressable memory and temporary data storage called scratch pad or general purpose registers. Its address range is from 00h to 7Fh. as shown in the fig.



6.2.1 Bank Registers

It has four bank registers named as Bank 0, Bank1, Bank2 and Bank 3. Each bank has 8 byte registers named as R0-R7. The table shows the address range of each bank and bank selection bits in PSW.

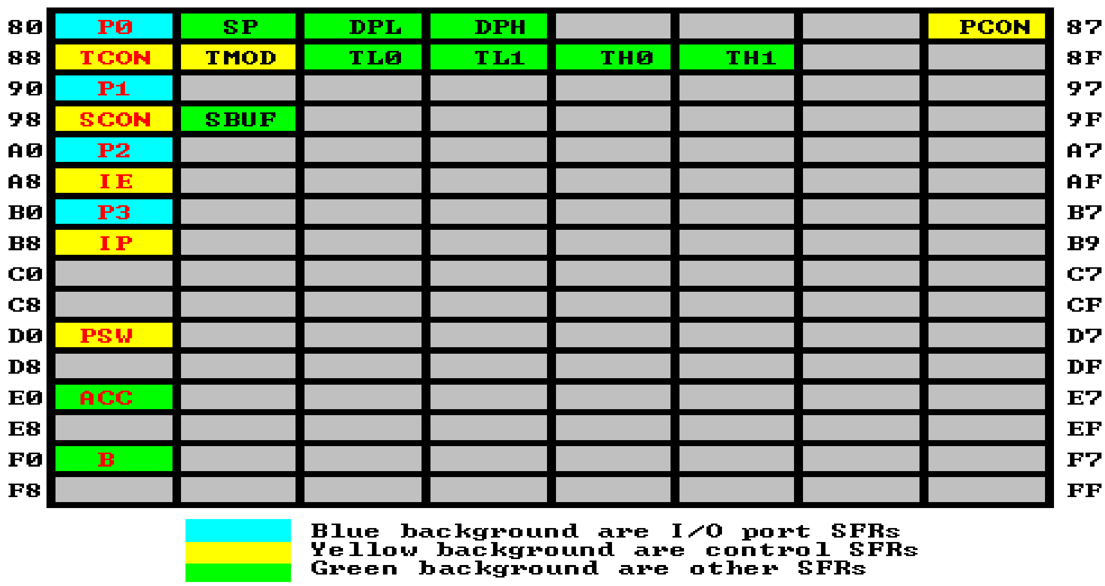


6.2.2 Bit-Addressable RAM

Bit addressable RAM Of the 128-byts internal RAM of the 89C51, only 16 bytes of it are bit addressable. The rest must be accessed in byte format. The bit – addressable RAM locations are 20H to 2FH. These 16 bytes provide 128 bits of RAM bit – address ability since 16 x 8 = 128. They are addressed as 0 to 127 (in decimal) or 00 to 7FH. Therefore, the bit addresses 0 to 7 are for the first byte of internal RAM location 20H, and 8 to 0FH are the bit addresses of the second byte, RAM location 21H, and so on. The last byte of 2FH has bit address byte of 2FH has bit addresses of 78H to 7FH. Note that internal RAM locations 20-2FH is both byte-addressable and bit addressable.

7. SPECIAL FUNCTION REGISTERS (SFR)

The CPU and supporting devices like I/O port, serial communication port, Timer and interrupt do not have registers itself. An internal 128 byte RAM is allotted for this purpose for temporary data storage and programming the supporting devices called Special Function Registers. Its address range is from 80h to FFh. Some SFR are byte addressable some SFR are bit and byte addressable as shown in the fig.



7.1 Timers

89C51 has two 16 bit timers' timer 0 and timer 1. Since the 89C51 has an 8-bit architecture, each 16-bit timer is accessed as two separate registers of low byte and high byte. Each timer is discussed separately. Timer0 and Timer 1 uses TL0, TH0, TL1, TH1, TMOD and TCON special function registers. Timer0 and Timer 1 are used for time delay generation and event counter. When internal clock frequency is used the timer act as a timer when external clock is used the timer act as a counter.

The timer operates in 4 modes. Mode-0, Mode-1, Mode-2 and Mode-3. Mode 1 and Mode 2 are frequently used by the programmer for time delay generation and baud rate generation.

7.1.1 Timer 0 Register

The 16-bit register of time 0 is accesses as low byte and high byte. The low byte register is called TL0 (timer 0 low byte) and the high byte register is referred to as TH0 (timer 0 high byte). These registers can be accessed like any other register. These registers can also be read like any other register.

7.1.2 Timer 1 Register;

Timer 1 is also 16 bits, and its 16- bit register is split into two bytes, referred to as TL1 (timer 1 low byte) and TH1 (timer 1 high byte). These registers are accessible in the same way as the registers of timer 0.

7.2 TMOD (Timer Mode) Register

Both timers 0 and 1 uses the same register, called TMOD, to set the various timer operation modes. TMOD is an 8-bit register in which the lower 4 bits are set aside for timer 0 and the upper 4 bits are set aside for timer 1. In each case, the lower 2 bits are used to set the timer mode and the upper 2 bits to specify the operation.

7.3 TCON Register

MSB four bits are used to control Timer 1 and Timer 0.

0	0	0	0	0	0	0	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

7.4 Serial Communication Port(UART)

The 89c51 two pins for serial communication Rx and Tx. 89C51 has UART module for asynchronous serial communication using RS 232 protocol. It uses three special function registers to perform serial communication and control.

7.4.1 SBUF (Serial Buffer) Register

SBUF is an 8-bit register used solely for serial communication in the 89C51. For a byte of data to be transferred via the TxD line, it must be placed in the SBUF register. Similarly, SBUF holds the byte of data when it is received by the 89C51 RxD line. SBUF can be accessed like any other register in the 89C51.The moment a byte is written into SBUF, it is framed with the start and stop bits and transferred serially via the TxD pin. Similarly, when the bits are received serially via RxD, the 89C51 de frames it by eliminating the stop and start bits, making a byte out of the data received, and then placing it in the SBUF. Some baud rates are shown below:

Baud Rate= 9600 TH1(Decimal)=-3 TH1(HEX) = FD

Baud Rate= 4800 TH1(Decimal)=-6 TH1(HEX) = FA

Baud Rate= 2400 TH1(Decimal)=-12 TH1(HEX) = F4

Baud Rate= 1200 TH1(Decimal)=-24 TH1(HEX) = E8

7.4.2 SCON(Serial Control) Register

The SCON register is an 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things. Following SCON bits are explained; SMO SCON.7 Serial port mode specifies

SM1/ SCON.6= Serial port mode specifies

SM2/ SCON.5= Used for multiprocessor communication (Make it 0)

REN/ SCON.4= Set / cleared by software to enable / disable reception.

TB8/ SCON.3= Not widely used.

RB8/ SCON.2= Not widely used.

T1/ SCON.1= Transmit interrupt flag. Set by hardware at the beginning of the stop bit in mode1. Must by cleared by software.

R1/ SCON.0= Receive interrupt flag. Set by hardware halfway through the stop bit time in mode1. Must be cleared by software.

7.5 Interrupts

89C51 has 6 interrupts in which INT0 and INT1are external interrupts, Timer0 , Timer1 and Serial interrupts are internal interrupts and RESET is one of the interrupt. All the interrupts are vectored interrupts. To enable interrupt operations the 89C51 uses three SFRs IE,IP and TCON. When any interrupt is received by the microcontroller it transfer the control sequence to the ISS.

7.5.1 IE SFR

EA	--	--	ES	ET1	EX1	ET0	EX0

	--	--	PS	PET1	PEX1	PET0	PEX0

8. INTERFACING 16 X 2 LCD , MOTOR AND BUZZER

8.1 Interfacing 16 x 2 LCD

LCD is one of the important display to display the results in alpha numeric character. In this project a 16 x 2 LCD is interfaced with 89C51 microcontroller in 4 bit mode to display the attendance details.

The 89C51 has 4 I/O ports named as Port 0, Port 1, Port 2 and Port 3 in which Port 1 is used to interface 16 x2 LCD. The LCD has RS, RW, EN and 8 data lines D0-D7. Here the LCD is interfaced with 89C51 in 4-bit mode. The interfacing of 89C51 Port 1 with LCD is shown in the figure.

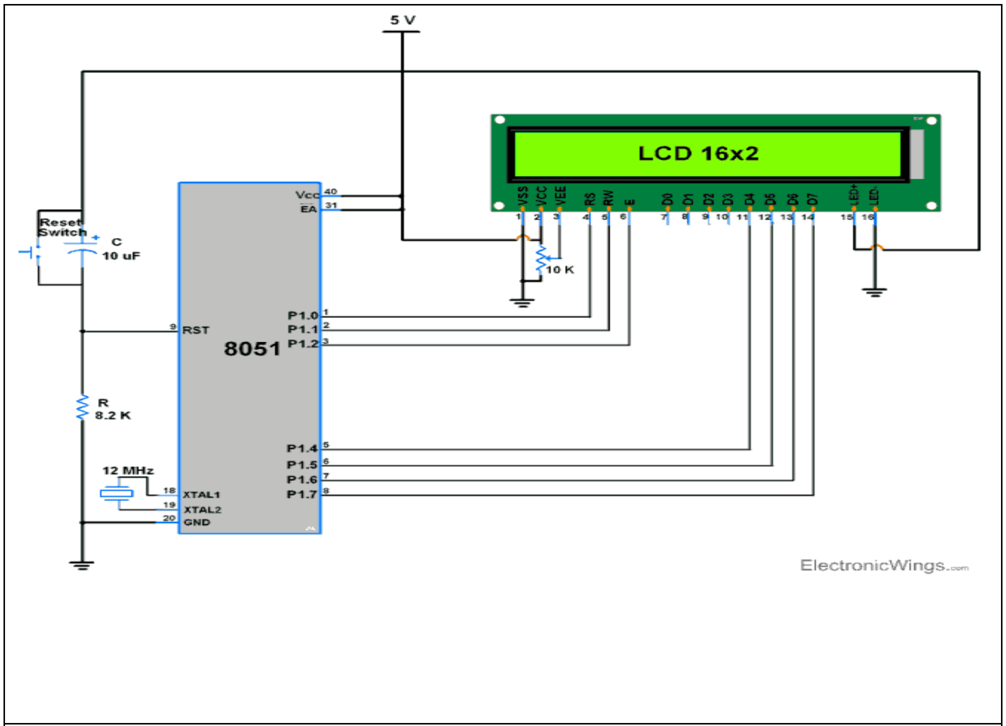


Figure 5.1 Interfacing LCD in 4 bit mode

LCD pin 1 VSS is connected to GND and 1st terminal of the trimmer potentiometer. LCD pin 2 VCC is connected to 5V DC and 3rd terminal of the trimmer potentiometer. LCD pin 3 VEE is connected to center tap 2nd terminal of the trimmer potentiometer to adjust the LCD resolution and light intensity.

8.2 RS/RW/EN:

Register Select(RS):

The LCD has two register namely a Data register and Command register. Any data that needs to be displayed on the LCD has to be written to the data register of LCD. Command can be issued to LCD by writing it to Command register of LCD. This signal is used to differentiate the data/cmd received by the LCD.

If the RS signal is **LOW** then the LCD interprets the 8-bit info as **Command** and writes it **Command register** and performs the action as per the command. If the RS signal is **HIGH** then the LCD interprets the 8-bit info as **data** and copies it to **data register**. After that the LCD decodes the data for generating the 5x7 pattern and finally displays on the LCD.

Read/Write(RW):

This signal is used to write the data/cmd to LCD and reads the busy flag of LCD. For write operation the RW should be **LOW** and for read operation the R/W should be **HIGH**.

Enable(EN):

This pin is used to send the enable trigger to LCD. After sending the data/cmd, Selecting the data/cmd register, Selecting the Write operation. An HIGH-to-LOW pulse has to be sent on this enable pin which will latch the info into the LCD register and triggers the LCD to act accordingly.

89C51 P1.0 is connected to LCD pin 4 RS (Register Select) to select the command and data registers in the SDRAM. P1.1 is connected to LCD pin 5 RW(Read Write) to perform read or write operations on the registers. P1.2 is connected to LCD pin 6 EN(Enable) to enable read write operations on the registers.

8.3 4 bit mode D4-D7

Data Bus: As shown in the above figure and table, an alphanumeric LCD has an 8-bit data bus referenced as D0-D7. As it is an 8-bit data bus, we can send the data/cmd to LCD in bytes. It also provides the provision to send the data/cmd in chunks of 4-bit, which is used when there is limited number of GPIO lines on the microcontroller.

For 4 bit mode only 4 data lines are used. Hence D4-D7 is used as data lines in 4 bit mode in which P1.4 is connected to LCD pin 11 D4 data pin , P1.5 is connected to LCD pin 12 D5 data pin, P1.6 is connected to LCD pin 13 D6 data pin and P1.7 is connected to LCD pin 14 D7 data pin as shown in the circuit diagram. D0-D3 is not used.

While programming the 16 x 2 LCD the following functions are used
LCD initialization function
LCD command function
LCD data function
LCD string function

The following commands are used in this project to control the LCD display
0x02 it is used to inform the LCD to operate in 4 bit mode
0x28 command is used to initialize the 16 x 2 LCD in 4 bit mode
0x0C command is used for display ON cursor OFF
0x01 command is used to clear display
0x80 command is used to bring the cursor at home position
Steps used to program 16 x 2 LCD in 4 bit mode

- In 4-bit mode, data/command is sent in a 4-bit (nibble) format.
- To do this 1st send Higher 4-bit and then send lower 4-bit of data/command.
- Only 4 data (D4 - D7) pins of 16x2 of LCD are connected to the microcontroller and other control pins RS (Register select), RW (Read/write), E (Enable) is connected to other GPIO Pins of the controller.
- Therefore, due to such connections, we can save four GPIO pins which can be used for another application.

Initialization function

1. Wait for 15ms, Power-on initialization time for LCD16x2.
2. Send 0x02 command which initializes LCD 16x2 in 4-bit mode.
3. Send 0x28 command which configures LCD in 2-line, 4-bit mode, and 5x8 dots.
4. Send any Display ON command (0x0E, 0x0C)
5. Send 0x0E command (Display ON, Cursor Blinking)

Command write function

1. First, send a higher nibble of command.
2. Make RS pin low, RS=0 (command reg.)
3. Make RW pin low, RW=0 (write operation) or connect it to ground.
4. Give High to Low pulse at Enable (E).
5. Send lower nibble of command.
6. Give High to Low pulse at Enable (E).

Data write function

1. First, send a higher nibble of data.
2. Make RS pin high, RS=1 (data reg.)
3. Make RW pin low, RW=0 (write operation) or connect it to ground.
4. Give High to Low pulse at Enable (E).
5. Send lower nibble of data.
6. Give High to Low pulse at Enable (E).

String character function

The string character function read the data from serial buffer by invoking data write function the string characters are displayed on the 16 x 2 LCD.

8.4 DC Motor interface:

A motor Driver IC is connected between microcontroller and DC motor. Motor driver is a little current amplifier. It takes a low current signal and gives out a high current signal which can drive a motor. It can also control the direction of motor. We can use any dual H-bridge IC like L293D or L298.

Here, the motor driver input pins IN1, IN2 are connected to the P2.5 and P2.4 respectively to control the motor directions. DC motor is connected to output terminals 3 and 6 of L293D. EN1 pin is connected to the 5V DC to drive the motor as shown in the circuit diagram.

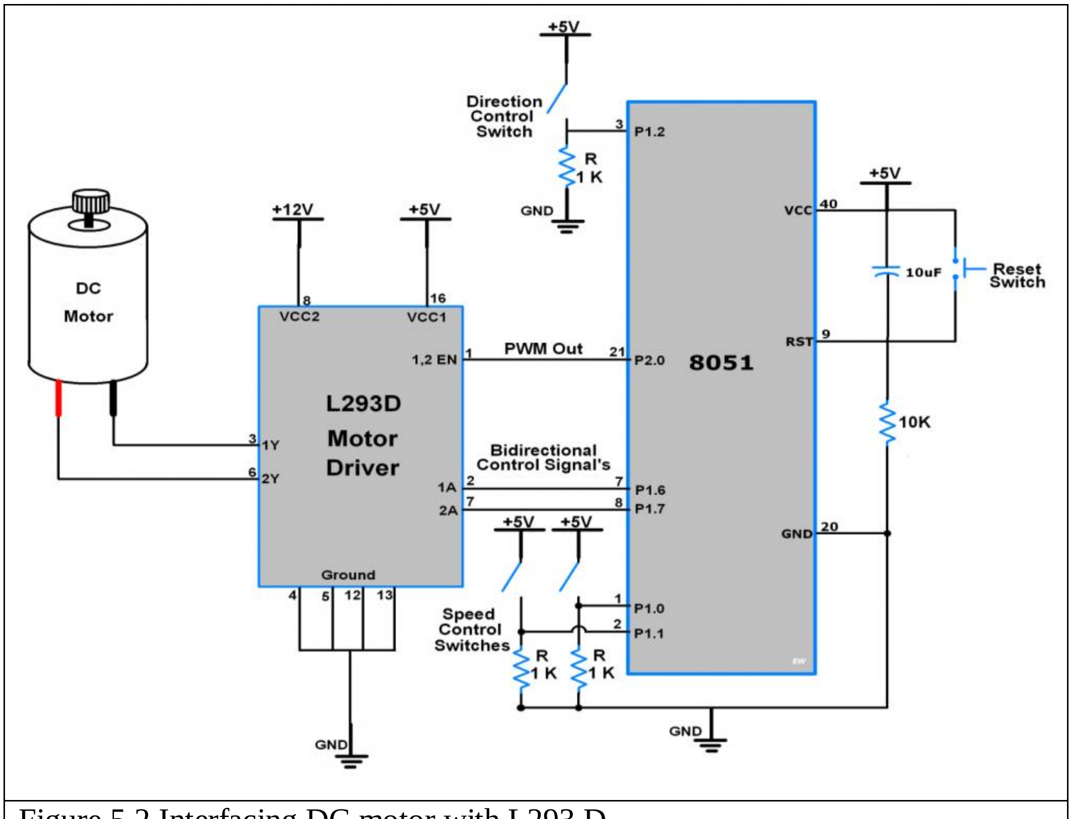


Figure 5.2 Interfacing DC motor with L293 D

8.5 Interfacing Buzzer:

The Buzzer is interfaced with 89S51 in I/O mode. The pin is defined using sbit. The buzzer is connected at the collector load of a NPN transistor BC 547. When the base voltage applied through the I/O pin the transistor becomes ON and enable the buzzer to make the beep sound to identify whether the RFID tag is accessed or not. Figure shows the interfacing diagram of the buzzer.

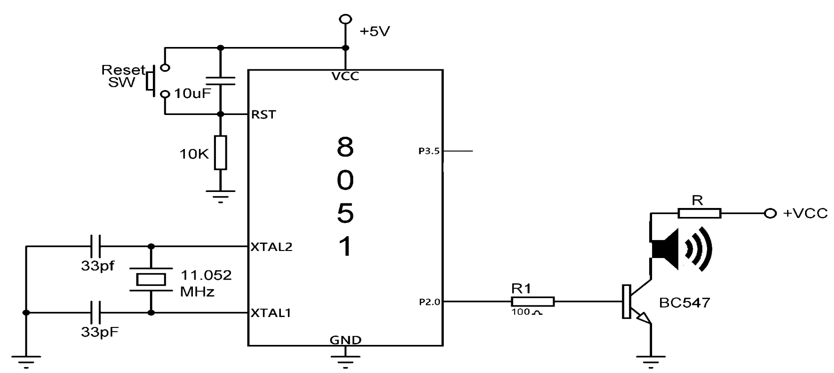


Figure 5.3 Buzzer interfacing

9. RFID BASED AUTOMATIC BUS FARE COLLECTION SYSTEM CIRCUIT OPERATION

RFID based automatic Bus Fare collection system consists of RFID Reader, RFID Tag, LCD display, motor, buzzer and microcontroller unit. RFID can be interfaced to microcontroller through USART. Data is transferred from RFID cards to reader and from there to microcontroller.

Radio frequency technology is used in many applications. Passive RFID tags are used in this project. Passive tags contain 10 digit number tag inbuilt in it, In real time, one can issue passive tags to the students, with their name/roll numbers as their tags.

RFID reader contains a copper winding in it. This winding acts as an antenna. When the tag is placed near the reader, due to the induced mutual inductance energy, data is transferred to reader. Reader then transfers data to the microcontroller. Microcontroller checks for the data continuously, if any data is received, microcontroller compares the data in data base. If the RFID tag is found it gives the bus fare to the concern people and the motor is rotated to open the door and give a buzzer sound. Then the motor is rotated in opposite direction to close the door. The system waits for scanning the next RFID card.

The Software is coded to scan the RFID using UART communication technique, by initializing the UART and read the character from the SBUF register by checking the Receive interrupt by the microcontroller and it is compared with data base using the following code.

```
void uart_init()
{
 TMOD=0x20;
 SCON=0x50;
 TH1=0xfd;
 TR1=1;
```

```
}
char rxdata()
{
  while(!RI);
  ch=SBUF;
  RI=0;
  return ch;
}
```

Main function it self compares the string data to make the bus fare and enable motor and buzzer.

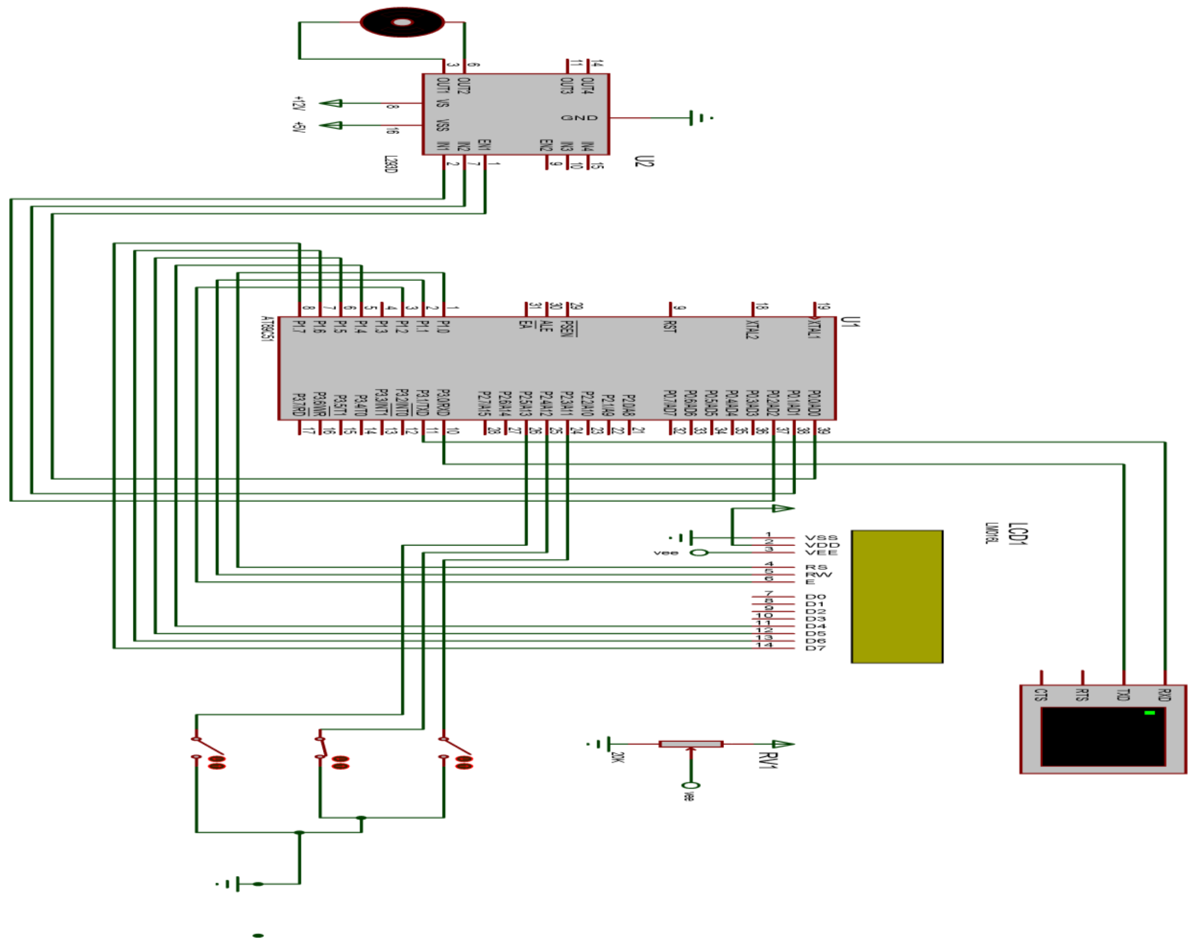
```
if(strncmp(rfid,"1234567890_",10)==0)
{
counter1++;
lcdcmd(1);
lcdstring("M.S.KAMALESH");
delay(200);
lcdcmd(1);
lcdstring(" BUS FARE ");
delay(200);
lcdcmd(1);
lcdstring("SELECT STOP");
delay(300);
lcdcmd(0xc0);
lcdstring("STP1 STP2 STP3");
delay(500);     //delay
lcdcmd(1);
        if(Stop1==0)
           {
           lcdcmd(0x0c);
            lcdstring("STP 1 SELECTED");
           delay(200);
```

```
lcdcmd(1);
lcdstring("STOP1=10RS");
delay(100);

lcdcmd(0xc0);
lcdstring("THANK YOU");
delay(100);

}
            else if(Stop2==0)
            {

lcdcmd(0x0c);
 lcdstring("STP 2 SELECTED");
delay(200);

lcdcmd(1);
lcdstring("STOP2=20RS");
delay(100);

lcdcmd(0xc0);
lcdstring("THANK YOU");
delay(100);
            }
                        else if(Stop3==0)
                        {

lcdcmd(0x0c);
 lcdstring("STP 3 SELECTED");
 delay(200);
```

```
        lcdcmd(1);
        lcdstring("STOP3=30RS");
        delay(100);

        lcdcmd(0xc0);
        lcdstring("THANK YOU");
        delay(100);

                        }

        else
        {
        lcdstring("STP NOT SELECTED");
        delay(250);
                                }
                        }

        Motor1=1;
        Motor2=0;
delay(300);
        Motor1=0;
        Motor2=0;
delay(200);
        Motor1=0;
        Motor2=1;
      }
```

The Complete circuit diagram is shown in the figure – and the coding is given in the annuxre.

10. HARDWARE DESIGN AND DESCRIPTION



As shown figure in above RFID Based Bus Fare collection system circuit diagram, 16X 2 LCD is connected in four bit mode with Microcontroller. LCD’s RS, RW and EN Pins are directly connected at PORT 1 Pin number P1.0, P1.1, and P1.2. D4, D5, D6 and D7 pins of LCD is directly connected at pin P1.4, P1.5, P1.6 and P1.7 of port 1. Motor driver is connected at Port pin number P0.1 and P0.2.

11. S0FTWARE DESIGN AND DESCRIPTION

While Programming the 89C51 Microcontroller for RFID based Automatic Bus Fare System, First of all we include header files and define input and output pin and variables.

```
#include<reg51.h>
#include<string.h>
#include<stdio.h>
#define LCDPORT P1
sbit rs=P1^0;
sbit rw=P1^1;
sbit en=P1^2;
sbit Stop1=P2^3;
sbit Stop2=P2^4;
sbit Stop3=P2^5;
sbit Speaker=P2^6;
sbit In1=P0^2;
sbit In2=P0^1;
char i,rx_data[50];
char rfid[13],ch=0;
//char j,Stop[5];                              //my cmd
int counter1, counter2, counter3;
unsigned char result[1];
 void delay(int itime)
{
   int i,j;
   for(i=0;i<itime;i++)
   for(j=0;j<1275;j++);
}
void daten()
{
rs=1;
```

```
rw=0;
   en=1;
delay(5);
   en=0;
}void lcddata(unsigned char ch)
{
   LCDPORT=ch& 0xf0;
daten();
   LCDPORT=(ch<<4) & 0xf0;
daten();
}
void cmden(void)
{
rs=0;
   en=1;
delay(5);
   en=0;
}
void lcdcmd(unsigned char ch)
{
   LCDPORT=ch& 0xf0;
cmden();
   LCDPORT=(ch<<4) & 0xf0;
cmden();
}
void lcdstring(char *str)
{
   while(*str)
   {
lcddata(*str);
      str++;
```

```
    }
}
void lcd_init(void)
{
lcdcmd(0x02);
lcdcmd(0x28);
lcdcmd(0x0e);
lcdcmd(0x01);
}
void uart_init()
{
 TMOD=0x20;
 SCON=0x50;
 TH1=0xfd;
 TR1=1;
}
char rxdata()
{
  while(!RI);
   ch=SBUF;
   RI=0;
    return ch;
}
//main function
void main()
{
Speaker=1;
uart_init();
lcd_init();

/*lcdstring("--DOOR IS OPEN--");
```

```
                        In1=1;
                        In2=0;
                        delay(300);
                        In1=0;
                        In2=0;
                        delay(300);
lcdcmd(0x01);
lcdstring("-DOOR IS CLOSE-");
          In1=0;
          In2=1;
          delay(300);
          In1=1;
          In2=1;
          delay(300);*/

lcdcmd(0x01);
delay(5);
lcd_init();
lcdstring("---RFID BASED---");
lcdcmd(0xc0);
delay(5);
lcdstring("BUS TICKETSYSTEM");
delay(5);
lcd_init();
lcdstring("---PASSENGER----");
lcdcmd(0xc0);
lcdstring("ENJOY TRAVEL");
delay(5);
```

```
        //while loop create
while(1)
  {
            {
            lcd_init();

lcdstring("--DOOR IS OPEN--");
                In1=1;
                In2=0;
                delay(300);
                In1=0;
                In2=0;
                delay(300);
lcdcmd(0x01);
lcdstring("-DOOR IS CLOSE-");
        In1=0;
        In2=1;
        delay(300);
        In1=1;
        In2=1;
        delay(300);
                }
lcdcmd(1);
lcdstring("SCAN YOUR CARD:");
lcdcmd(0xc0);
i=0;
    for(i=0;i<10;i++)
rfid[i]=rxdata();
rfid[i]='\r';
lcdcmd(1);
lcdstring("RFID No. is:");
```

```
lcdcmd(0xc0);
        for(i=0;i<10;i++)
lcddata(rfid[i]);
delay(500);     //first ID card dat if statement
        if(strncmp(rfid,"078$001824_",10)==0)
        {
counter1++;
lcdcmd(1);
lcdstring("M.S.KAMALESH");
delay(200);
lcdcmd(1);
lcdstring(" BUS FARE ");
delay(200);
lcdcmd(1);
lcdstring("SELECT STOP");
delay(300);
lcdcmd(0xc0);
lcdstring("STP1 STP2 STP3");
delay(500);     //delay changed 100-500
lcdcmd(1);
         if(Stop1==0)     //else   0-1
           {
                        lcdcmd(0x0c);
              lcdstring("STP 1 SELECTED");
                        delay(200);
                        lcdcmd(1);
                        lcdstring("STOP1=10RS");
                        delay(100);
                        lcdcmd(0xc0);
                        lcdstring("THANK YOU");
                        delay(100);
```

```
}
        else if(Stop2==0)
        {
                lcdcmd(0x0c);
   lcdstring("STP 2 SELECTED");
          delay(200);
                lcdcmd(1);
                lcdstring("STOP2=20RS");
                delay(100);
                lcdcmd(0xc0);
                lcdstring("THANK YOU");
                delay(100);
        }
                else if(Stop3==0)
                {
                        lcdcmd(0x0c);
    lcdstring("STP 3 SELECTED");
           delay(200);
                        lcdcmd(1);
                        lcdstring("STOP3=30RS");
                        delay(100);
                        lcdcmd(0xc0);
                        lcdstring("THANK YOU");
                        delay(100);

                }
                        else
                        {
                                lcdstring("STP NOT SELECTED");
                                delay(250);
```

```
                                        }
                              }
// 2 ID if statement
           else if(strncmp(rfid,"5$00107595_",10)==0)
       {
counter2++;
lcdcmd(1);
lcdstring("P.N.RAGAVENDRAN");
delay(200);
lcdcmd(1);
lcdstring(" BUS FARE ");
delay(200);
lcdcmd(1);
lcdstring("SELECT STOP");
delay(300);
lcdcmd(0xc0);
lcdstring("STP1 STP2 STP3");
 delay(500);    //delay changed 100-500
lcdcmd(1);
          if(Stop1==0)
          {
                    lcdcmd(1);
                    lcdstring("STP 1 SELECTED");
                    delay(80);

                    lcdcmd(1);
                    lcdstring("STOP1=10RS");
                    delay(250);

                    lcdcmd(0xc0);
                    lcdstring("THANK YOU");
```

```
            delay(100);

    }
            else if(Stop2==0)
            {
            lcdcmd(1);
            lcdstring("STP 2 SELECTED");
            delay(80);

            lcdcmd(1);
            lcdstring("STOP2=20RS");
            delay(250);

            lcdcmd(0xc0);
            lcdstring("THANK YOU");
            delay(100);
            }

                    else if(Stop3==0)
                    {
                            lcdcmd(1);
               lcdstring("STP 3 SELECTED");
               delay(80);

                            lcdcmd(1);
                            lcdstring("STOP3=30RS");
                            delay(250);

                            lcdcmd(0xc0);
                            lcdstring("THANK YOU");
                            delay(100);
```

```
                }

                    else
                    {
                        lcdstring("NOT VALIED");
                        delay(250);
        }       }
        //3 ID if statement
            else if(strncmp(rfid,"3$00107320_",10)==0)
      {
counter1++;
lcdcmd(1);
lcdstring("RAMAN");
delay(200);
lcdcmd(1);
lcdstring(" BUS FARE ");
delay(200);
lcdcmd(1);
lcdstring("SELECT STOP");
delay(300);
lcdcmd(0xc0);
lcdstring("STP1 STP2 STP3");
delay(100);
lcdcmd(1);
 if(Stop1==0)
        {
            lcdcmd(0x0c);
         lcdstring("STP 1 SELECTED");
            delay(200);
            lcdcmd(1);
            lcdstring("STOP1=10RS");
```

```
            delay(100);
            lcdcmd(0xc0);
            lcdstring("THANK YOU");
            delay(100);

}
            else if(Stop2==0)
            {

                        lcdcmd(0x0c);
   lcdstring("STP 2 SELECTED");
             delay(200);

                        lcdcmd(1);
                        lcdstring("STOP2=20RS");
                        delay(100);

                        lcdcmd(0xc0);
                        lcdstring("THANK YOU");
                        delay(100);
            }
                        else if(Stop3==0)
                        {
                                    lcdcmd(0x0c);
     lcdstring("STP 3 SELECTED");
               delay(200);

                                    lcdcmd(1);
                                    lcdstring("STOP3=30RS");
                                    delay(100);
```

```
                lcdcmd(0xc0);
                lcdstring("THANK YOU");
                delay(100);
            }
        }

else
    {
lcdcmd(1);
lcdstring("CARD DON'T EXIST");
lcdcmd(0xc0);
lcdstring("TRY ANOTHER CARD");
                }

        }
    }
```

CONCLUSION

The fare collection problem has been eliminated Moreover, the project phase is completed successfully by using smart card . This project is made with pre-planning, that it provides flexibility in operation. This innovation has made more desirable and economical. This project “RFID BASED AUTOMATIC BUS FARE COLLECTION SYSTEM “ is designed with the hope that it is very much economical and helpful for passengers and as well as conductors during Journey.

REFERENCES

[1] S. Lahiri, “RFID Sourcebook”, New Jersey: IBM Press, 2006.

[2] S. Shepard, “RFID Radio Frequency Identification”, s.l.: MacGraw-Hill, 2005.

[3] Radio-frequency identification. Wikipedia (online). Wikimedia Foundation, Inc., 2009.

[4] Sato DCS & Labeling Worldwide, “The RFID Guidebook (Revision 8)”, 2004.