DEEPAK GUPTA

Digital Clock Using 8051

VASEEM DURRANI

A digital clock keeps track of the time and uses RTC(real Time Clock).The time is displayed on the LCD, which is connected to Port 2.The RTC used is DS12887 and the controller used is 89v51RD2.Three switches acts as a input for seconds, minutes and hours.

The above circuit is build on proteus and crystal frequency has been internally set. The connection of the RTC DS12C887 with the microcontroller is shown in the circuit diagram.It uses update interrupt to keep the track of the time. Every time the update interrupt comes, the clock is incremented by one second. The output is displayed on the LCD. The clock uses external interrupt 2 of the microcontroller AT89C51 for setting the time. A user can set time by pressing the switch connected to pin 13 of the microcontroller, which is interrupt 2. The hour and minutes can be set using pin 5 and pin 6 of the controller AT89C51 respectively. Once the time is set, the user needs to press the start pin (pin 8 of controller) to start the clock.

The component used are 8051

The main features of 8051 microcontroller are:

i. RAM – 128 Bytes (Data memory)

ii. ROM – 4Kbytes (ROM signify the on – chip program space)

iii. Serial Port – Using UART makes it simpler to interface for serial communication. iv. Two 16 bit Timer/ Counter

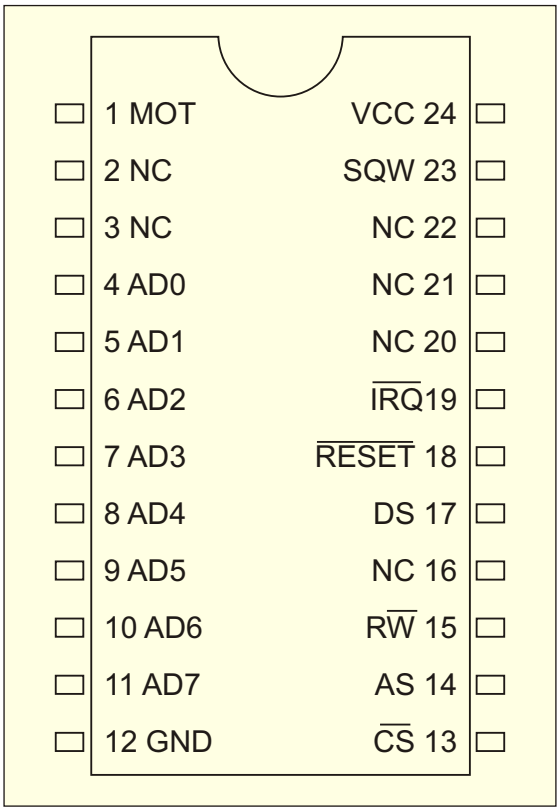
v. Input/output Pins – 4 Ports of 8 bits each on a single chip.

vi. 6 Interrupt Sources

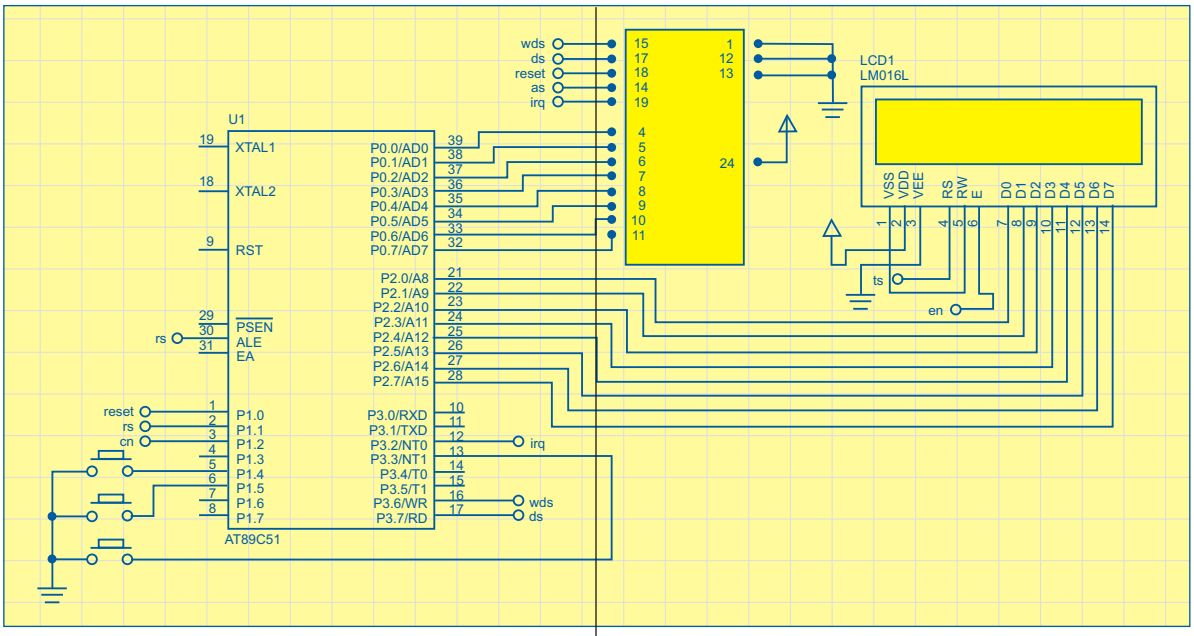
vii. 8 – bit ALU (Arithmetic Logic Unit) viii. Harvard Memory Architecture – It has 16 bit Address bus (each of RAM and ROM) and 8 bit Data Bus.

ix. 8051 can execute 1 million one-cycle instructions per second with a clock frequency of 12MHz.

DS12887 RTC



The real-time clock (RTC) is a widely used device that provides accurate time and date for many applications. It provides time components of hour, minute, and second, in addition to the date/calendar components of year, month, and day. The RTC chip uses an internal battery, which keeps the time and date even when the power is off. One of the most widely used RTC chips is the DS 12887 from Dallas Semiconductor/Maxim Corp. The DS 12887 supports both 12-hour and 24-hour clock modes with AM and PM in the 12-hour mode. It also supports the Daylight Savings Time option. The DS 12887 uses CMOS technology to keep the power consumption low and it has the designation DS12C887, where C is for CMOS. The DS12887 has a total of 128 bytes of nonvolatile RAM. It uses



14 bytes of RAM for clock/calendar and control registers, and the other 114 bytes of RAM are for general-purpose data storage.

Pins of DS12887

Vcc

Pin 24 provides external supply voltage to the chip. The external voltage source is +5V. When VC9 falls below the 3V level, the external source is switched off and the internal lithium battery provides power to the RTC.

GND

Pin 12 is the ground.

ADO-AD7

The multiplexed address/data pins provide both addresses and data to the chip. Addresses are latched into the DS 12887 on the falling edge of the AS (ALE) signal. ADO – AD7 of the DS 12887 are connected directly to the 8051 and there is no need for any 74xx373 latches, since the DS 12887 provides the latch internally.

AS (ALE)

AS (address strobe) is an input pin. On the falling edge it will cause the addresses to be latched into the DS 12887. The AS pin is used for demultiplexing the address and data and is connected to the ALE pin of the 8051 chip.

MOT

This is an input pin that allows the choice between the Motorola and Intel microcontroller bus timings. The MOT pin is connected to GND for the Intel timing. That means when we connect DS 12887 to the 8051, MOT = GND.

DS

Data strobe or read is an input. When MOT = GND for Intel timing, the DS pin is called the RD (read) signal and is connected to the RD pin of the 8051.

R/W

Read/Write is an input pin. When MOT = GND for the Intel timing, the R/W pin is called the WR (write)

Address Location	Function	Decimal Range	Data Mode Range Binary(Hex)_____ BCD	
0	Seconds	0-59	00-3B 59	00-
1	Seconds alarm	0-59	00-3B 59	00-
2	Minutes	0-59	00-3B 59	00-
3	Minutes alarm	0-59	00-3B 59	00-
4	Hours-12hr mode	1-12	01-0C AM 12 AM	01-
	Hours-24hr mode	0-23	81-8C PM 92 PM	81-
	Hours-12hr mode	1-12	0-17	0-23
5	Hours Alarm,12 hr	1-12	01-0C AM 12 AM	01-
	Hours Alarm,12 hr	1-12	81-8C PM 92PM	81-
	Hours Alarm,24 hr	0-23	0-17	0-23
6	Day of week,sun=1	1-7	01-07 07	01-
7	Day of month	1-31	01-1F 31	01-
8	Month	1-12	01-0C 12	01-
9	Year	0-99	00-63 99	00-

signal and is connected to the WR pin of the 8051.
CS
Chip select is an input pin and an active low signal. During the read (RD) and write (WR) cycle time of Intel timing, the CS must be low in order to access the chip. It must be noted that the CS works only when the external Vcc is connected. In other words "when Vcc falls below 4.25V, the chip-select input is internally forced to an inactive level regardless of the value of CS at the input pin." This is called the write-protected state. When the DS 12887 is in write-protected state, all inputs are ignored.
IRQ
Interrupt request is an output pin and active low signal. To use IRQ, the interrupt-enable bits in register B must be set high.
SQW
Square wave is an output pin. We can program the DS 12887 to provide up to 15 different square waves. The frequency of the square wave is set by programming register A RESET
Pin 18 is the reset pin. It is an input and is active low (normally high). In most applications the reset pin is connected to the Vcc pin. In applications where this pin is used, it has no effect on the clock, calendar, or RAM if it is forced low. The low on this pin will cause the reset of the IRQ and clearing of the SQW pin.

Address map of the DS12887

The DS12887 has a total of 128 bytes of RAM space with addresses 00 -7FH. The first ten locations, 00 – 09, are set aside for RTC values of time, calendar, and alarm data. The next four bytes are used for the control and status registers. They are registers A, B, C, and D and are located at addresses 10-13 (OA – OD in hex). Notice that their hex addresses match their names. The next 114 bytes from addresses OEH to 7FH are available for data storage. The entire 128 bytes of RAM are accessible directly for read or write except the following:

1) Registers C and D are read-only.
2) D7 bit of register A is read-only.
3) The high-order bit of the seconds byte is read-only.

Program Code

```
#include<reg51.h>
#include<absacc.h>
#define dataport P2
#define port P1
sbit reset = port ^ 0;
sbit rs = port ^ 1;
sbit rw = port ^ 2;
sbit e = port ^ 3;
sbit dig_hr1 =port ^ 4;
sbit dig_min1 =port ^ 5;
sbit start=port ^ 6;

int hr0 ,hr1 =0;
int min0,min1 =0; int sec0,sec1 =0;
unsigned char temp,hr,min,sec,num[60]={0X00,0X01,0X02,0X03,0X04,0X05,0X06,0X07,0X08,0X09,0X10,0X11,0X12,0X13,0X14,0X15,0X16,0X17,0X18,0X19,0X20,0X21,0X22,0X23,0X24,0X25,0X26,0X27,0X28,0X29,0X30,0X31,0X32,0X33,0X34,0X35,0X36,0X37,0X38,0X39,0X40,0X41,0X42,0X43,0X44,0X45,0X46,0X47,0X48,0X49,0X50,0X51,0X52,0X53,0X54,0X55,0X56,0X57,0X58,0X59};

void delay(unsigned int msec )
{
        int i ,j ;
        for(i=0;i<msec;i++)
        for(j=0; j<1275; j++);
}

void lcd_cmd(unsigned char item)
{
        dataport = item;
        rs= 0;
        rw=0;
        e=1;
        delay(1);
        e=0;
        return;
}
// DATA SENDING FUNCTION
void lcd_data(unsigned char item)
{
        dataport = item;
        rs= 1;
        rw=0;
        e=1;
        delay(1);
        e=0;
        return;
}

void lcd_data_string(unsigned char *str)
{
        int i=0;
        while(str[i]!='\0')
        {
                lcd_data(str[i]);
```

```
                i++;
                delay(1);
        }
        return; }

lcd_data_int(int time_val)
{
        int int_amt;
        int_amt=time_val/10;
        lcd_data(int_amt+48);
        int_amt=time_val%10;
lcd_data(int_amt+48);
}

void lcd()                          // Function to
initialize LCD
{
        lcd_cmd(0x38);
delay(5);
        lcd_cmd(0x0F);          delay(5);
        lcd_cmd(0x80);
        delay(5);
}

void set_rtc_time()                 // Function to
set time in RTC
{
        XBYTE[10]=0x20;
        XBYTE[11]=0x82;
        XBYTE[0]=0x00;
        XBYTE[2]=min;
        XBYTE[4]=hr;
        XBYTE[7]=0x01;
        XBYTE[8]=0x01;
        XBYTE[9]=0x10;
        XBYTE[1]=0xFF;
        XBYTE[3]=0xFF;
        XBYTE[5]=0xFF;
        XBYTE[11]=0x12;
}

void set_hr1()
{
        hr1++;
        if(hr1>23)
        hr1=0;
        lcd_cmd(0xc3);
        lcd_data_int(hr1);
        lcd_data(':');
        hr0=hr1;
}

void set_min1()
{
        min1++;
        if(min1>59)
        min1=0;
        lcd_cmd(0xc6);
        lcd_data_int(min1);
        min0=min1;
}

void set_time()     interrupt 2                 //
Time set
{
        lcd_cmd(0x01);
        if(start==0)
        {
                lcd_data_string("SET TIMING");
                lcd_cmd(0xc3);
                lcd_data_int(hr1);
                lcd_data(':');
                lcd_data_int(min1);
                while(start==0)
                {
                        delay(10);
                        if(dig_hr1==0)
                        set_hr1();
                        if(dig_min1==0)
                        set_min1();
                }
        }
        lcd_cmd(0x01);
        hr=num[hr1];
        min=num[min1];
        set_rtc_time();
}

bcdconv(unsigned char mybyte)
{
        unsigned char x,y;
        x=      mybyte & 0x0F;
        x=x | 0x30;
        y= mybyte & 0xF0;
        y=y>>4;
        y=y | 0x30;
        lcd_data(y);
        lcd_data(x);
}

void read_rtc_display() interrupt 0     //  Alarm
interrupt
{       lcd_cmd(0x01);
        lcd_cmd(0x80);
        lcd_data_string("TIME:");
        lcd_cmd(0X87);
```

```
        reset=0;
        reset=1;
        XBYTE[11]=0x12;
        hr=XBYTE[4];
        temp=0x87;
        bcdconv(hr);
        lcd_data(':');
        min=XBYTE[2];
        bcdconv(min);
        lcd_data(':');
        sec=XBYTE[0];
        bcdconv(sec);
}

void main()
{
        reset=1;
        lcd();
        XBYTE[10]=0x20;
        XBYTE[11]=0x12;
        lcd_cmd(0x01);
        IE=0x85;
        while(1);
}
```