

```
//#include "C:\Users\caleb\Documents\CLOCK_PROJECT\clock2.h"
```

```
#include <16F887.h>
```

```
#device adc=8
```

```
#FUSES NOWDT          //No Watch Dog Timer
```

```
#FUSES HS             //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
```

```
#FUSES NOPUT         //No Power Up Timer
```

```
#FUSES NOMCLR        //Master Clear pin used for I/O
```

```
#FUSES NOPROTECT     //Code not protected from reading
```

```
#FUSES NOCPD         //No EE protection
```

```
#FUSES NOBROWNOUT   //No brownout reset
```

```
#FUSES NOIESO        //Internal External Switch Over mode disabled
```

```
#FUSES NOFCMEN       //Fail-safe clock monitor disabled
```

```
#FUSES NOLVP         //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
```

```
#FUSES NODEBUG       //No Debug mode for ICD
```

```
#FUSES NOWRT         //Program memory not write protected
```

```
#FUSES BORV40        //Brownout reset at 4.0V
```

```
#use delay(clock=4000000)
```

```
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
```

```
#define RTC_SCLK PIN_C3
```

```
#define RTC_IO  PIN_C5
```

```
#define RTC_RST PIN_C2
```

```

#define D 1

#define setn PIN_E0

#define incr PIN_E1

#define decr PIN_E2

#define check PIN_B7


#include <ds1302.c>

#include <lcd.c>


byte day,mth,year,dow,hour,min,sec;


int decode(int8 sec)
{
    int code;
    switch(sec)
    {
        case 0: code = 0x40;break;
        case 1: code = 0x79;break;
        case 2: code = 0x24;break;
        case 3: code = 0x30;break;
        case 4: code = 0x19;break;
        case 5: code = 0x12;break;
        case 6: code = 0x02;break;
        case 7: code = 0x78;break;
        case 8: code = 0x00;break;
    }
}

```

```

        case 9: code = 0x10;break;
    }
    return code;
}

void display7seg()
{
    int Sec_units, Min_units, Sec_tens, Min_tens, Hr_units, Hr_tens;
    int Result;

    rtc_get_time( hour, min, sec );

    Sec_tens = sec/10;
    Sec_units= sec%10;
    Min_tens = min/10;
    Min_units= min%10;
    Hr_tens = hour/10;
    Hr_units= hour%10;

    output_A(0x01);
    Result = decode(Hr_tens);
    output_B(Result);
    delay_ms(D);

    output_A(0x02);

```

```
Result = decode(Hr_units);
```

```
output_B(Result);
```

```
delay_ms(D);
```

```
output_A(0x04);
```

```
Result = decode(Min_tens);
```

```
output_B(Result);
```

```
delay_ms(D);
```

```
output_A(0x08);
```

```
Result = decode(Min_units);
```

```
output_B(Result);
```

```
delay_ms(D);
```

```
output_A(0x10);
```

```
Result = decode(Sec_tens);
```

```
output_B(Result);
```

```
delay_ms(D);
```

```
output_A(0x20);
```

```
Result = decode(Sec_units);
```

```
output_B(Result);
```

```
delay_ms(D);
```

```
}
```

```

void get_clock()
{
    rtc_get_time( hour, min, sec );
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Time: %02d:%02d:%02d  ",hour,min,sec);
}

```

```

void get_date()
{
    rtc_get_date( day, mth, year, dow);
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Date:%02d/%02d/20%02d",day,mth,year);
}

```

```

void set_clock()
{
    //  rtc_set_datetime(day,mth,year,dow,hour,min);
    rtc_set_datetime(12, 10,15, 2,5,59);
}

```

```

void main()
{
    rtc_init();

```

```
lcd_init();
```

```
set_clock();
```

```
while(true)
```

```
{
```

```
//  get_date();
```

```
//  get_clock();
```

```
    display7seg();
```

```
}
```

```
}
```