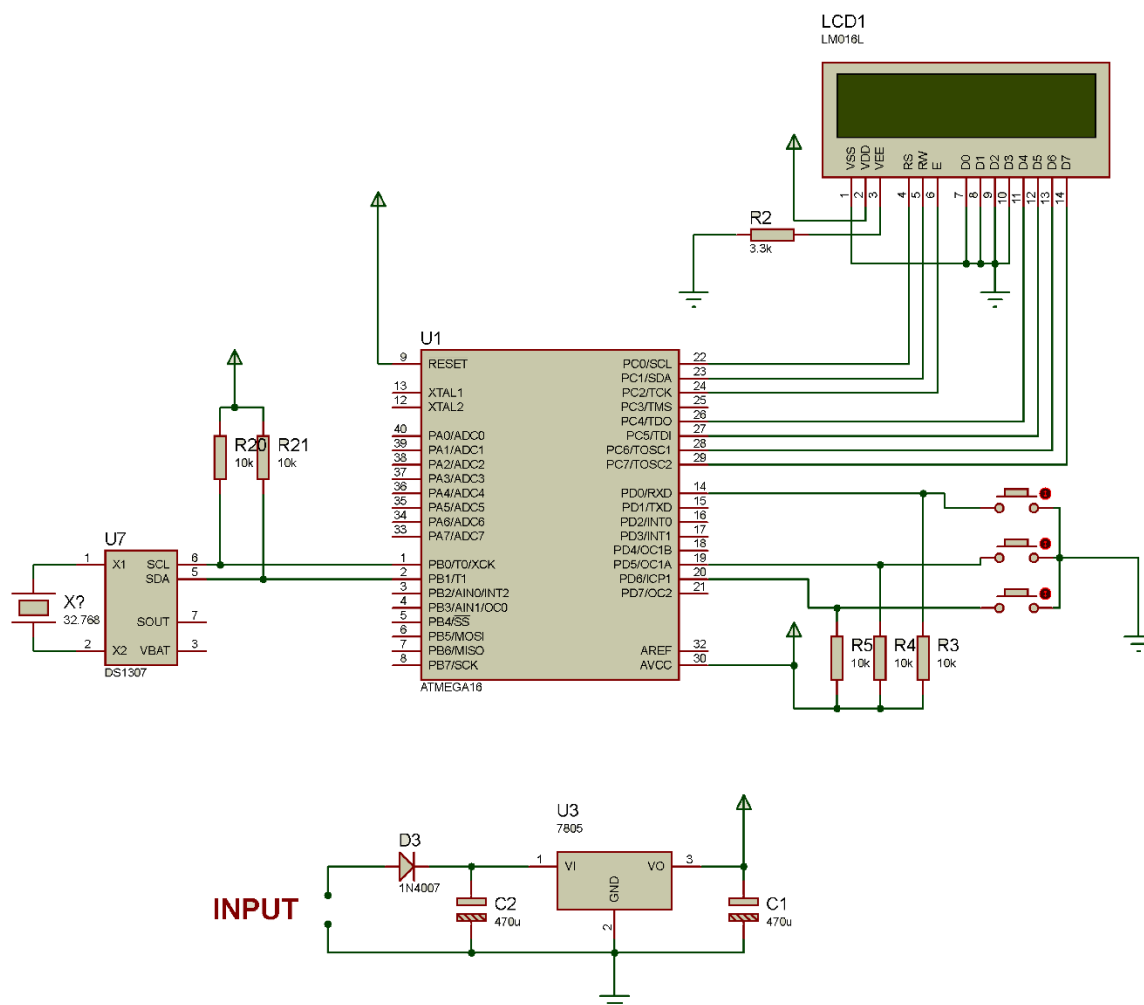


پروتئوس مدار تاریخ-زمان



LCD

VDD و VSS: مثل پایه های مثبت و منفی باتری هستند. VDD به 5V و VSS به زمین وصل است.

VEE: پررنگ یا کم رنگ بودن نوشته های روی صفحه را مشخص می کند. هر چقدر شدت جریان بیشتر باشد، نوشته ها هم پررنگتر می شوند. این پایه را می توان به یک پتانسیومتر وصل کرد که در این صورت هر چه مقاومت کمتر شود، نوشته ها پررنگتر می شوند. می توان پایه را به یک فوتورزیست وصل کرد که رنگ نوشته ها بر اساس نور محیط تعیین می شود.

RS : مشخص می کند که دیتایی که از طریق باس داده انتقال می یابد از نوع دستور (up code) است یا از نوع داده. اگر $RS=0$ دستور و اگر $RS=1$ باشد از نوع داده است.

RW : اگر $RW=0$ باشد میکرو روی LCD می نویسد و اگر 1 باشد میکرو از LCD می خواند.

E : اگر این پایه فعال نباشد نمی توان روی LCD نوشت یا از روش خواند. با یک فعال می شود.

D : این پایه ها باس داده هستند. پایه D0 تا D3 برای خواندن و پایه های D4 تا D7 برای نوشتن هستند. D0 تا D3 به زمین وصل شده اند چون به خواندن نیاز نبوده.

پایه های ۱۶ و ۱۵ : برای تعیین شدت نور صفحه LCD استفاده می شوند.

قسمت تغذیه

ورودی (INPUT): ولتاژ ورودی دستگاه و یک باتری 9V است.

دیود محافظ : چون دیود جریان را فقط از یک طرف رد می کند، اگر منفی و مثبت باتری را درست و ثل نکنیم صدمه ایی به دستگاه وارد نمی شود.

C2 : خازن تصبیت کننده ولتاژ ورودی رگولاتور.

C1 : ثبیت کننده ولتاژ خروجی رگولاتور.

رگولاتور 7805 : رگولاتور کار تنظیم ولتاژ را انجام می دهد. این رگولاتور سه پایه است و ولتاژی بین 7.5 تا 12 ولت را دریافت کرده و 5V را خارج می کند که مدار از آن استفاده خواهد کرد.

میکرو

دکمه ها : هر کدام از دکمه ها به یک مقاومت pull up وصل شده اند تا از ورود مقدار ناشناخته به میکرو جلوگیری شود.

RESET میکرو : این پایه active low است که یعنی اگر به زمین وصل شود یا به آن صفر داده شود اطلاعات روی میکرو را reset می کند. پایه را به 5V وصل کردیم تا از ریست ناخواسته جلوگیری شود.

آی سی DS1307

این آی سی بر روی ماژول Tiny RTC I2C قرار داشته و وظیفه تولید تاریخ و زمان را دارد (میکرو فقط ساعت و زمان تولید شده را نمایش می دهد). آی سی با پروتکل I2C کار می کند.

پروتکل I2C : پروتکلی برای انتقال اطلاعات به صورت سریال است و همیشه از دو پایه SDA و SCL استفاده می کند.

X1 , X2 : پایه های کریستال ساعت هستند. برای تنظیمات خود آی سی استفاده شده است.

SDA : پایه Serial Data برای انتقال و دریافت دیتا استفاده می شود. به صورت خروجی و ورودی کار می کند. میکرو اطلاعات تاریخ و ساعت را از طریق این پایه می خواند.

SCL : پایه Serial Clock ورودی است و کلاک میکرو را وارد می کند.

در هر دو پایه SDA و SCL از مقاومت pull up برای جلوگیری از ایجاد مقدار ناشناخته استفاده شده است (مقدار روی پایه همیشه یا یک منطقی یا صفر منطقی است).

VBAT : پایه ولتاژ باتری پشتیبان است که تاریخ و زمان را نگه می دارد. این باتری لیتیومی (3V) فقط برای خود آی سی و ماژول استفاده می شود و ولتاژ کار دستگاه را فراهم نمی کند.

SQW/OUT : پایه SQUARE WAVE / OUTPUT Driver به صورت خروجی است و موج مربعی تولید می کند که برای ایجاد تاریخ و زمان در تنظیمات آی سی استفاده می شود.

سوال استاد : چرا مقدار فرکانس کریستال عجیبه ؟

یک کیلو در کامپیوتر 2^{10} یا 1024 است که باعث ایجاد خرد ارقام در محاسبات فرکانس می شود.

با نرم افزار پروتئوس ، مدار را شبیه سازی می کنیم تا ببینیم درست کار می کند یا نه.

- شیوه اجرای مدار تاریخ و زمان در پروتئوس :

ابتدا اولین دکمه (بالا ترین) را فشار می دهیم تا وارد منو تاریخ و ساعت شویم. با فشار دادن دکمه دوم و سوم می توانیم بالا و پایین رفته و بین دو گزینه تاریخ و زمان یکی را انتخاب کنیم. روز set time رفته و دکمه اول را می زنیم تا وارد تنظیمات ساعت شویم. با زدن دکمه دوم و سوم می توانیم عدد ساعت را تغییر دهیم. بعد دوباره دکمه اول را می زنیم تا وارد تنظیمان دقیقه شویم. و با زدن دوباره دکمه اول ، وارد تنظیمات ثانیه می شویم. بعد از تنظیم ثانیه با زدن دکمه اول تنظیمات ذخیره شده و به تنظیمات اولیه باز می گردیم. برای تنظیم تاریخ نیز دوباره دکمه اول را فشار داده ، این بار در منو گزینه set date را انتخاب می کنیم و به همان صورت قبل تنظیمات تاریخ را انجام می دهیم.

- کد ویژن :

سه کاربرد دارد:

۱- یک محیط برنامه نویسی به زبان C در اختیار ما قرار می دهد.

۲- فایل خروجی hex به ما می دهد.

۳- با آن می توانیم پروگرامر را کنترل کرده و فایل hex را روی میکرو بریزیم.

در ابتدا در کد ویژن در منو File وارد New می شویم. دو گزینه در اختیار ما می گذارد.

۱- Source file : که آخرین فایلی که در سیستم با آن کار می کردیم را برای ما load می کند.

۲- Project : یک پروژه جدید می سازد. که ما این گزینه را انتخاب می کنیم.

در ابتدا از شما می پرسد آیا می خواهید از CodeWizard استفاده کنید؟ بله

کد ویزارد یک رابط گرافیکی دارد که با انتخاب چندین گزینه تنظیمات مربوط به پایه ها ، رجیسترها ، وقفه ها و ... را خودش برای شما در کد می نویسد در صورتی که با کد ویزارد کار نکنید خودتان باید این تنظیمات را در کد بنویسید.

در ادامه باید نوع میکرو خود را انتخاب کنید که برای ما Atmega است. صفحه تنظیمات کد ویزارد باز می شود. در قسمت chip نوع میکرو (Atmega 16A) و کلاک آن را تنظیم می کنید. در قسمت port هر چهار تا پورت A, B, C, D شما را نشان می دهد. که برای پایه های آن ها می توانید ورودی و خروجی بودن را تعیین کنید. اگر پایه را ورودی انتخاب کنید (In) در مقابل آن دو حالت P (استفاده از مقاومت pull up) و حالت T (Tri-state یعنی سیم می تواند به سه حالت ۱- اتصال به ۱ منطقی ۲- اتصال به صفر منطقی ۳- شناور دارد).

اگر پایه را خروجی (out) انتخاب کنید در مقابل ولتاژ آن را می توانید ۰ یا ۱ منطقی قرار دهید. در قسمت external Interrupts وقفه های خارجی و مد کاری آن ها (حساس به سطح یا لبه) را مشخص می کنیم. در قسمت timers از شمارنده استفاده می کنیم و منبع کلاک و مد کاری آن را مشخص می کنیم. در قسمت Bit-Banged I2C Bus Interface تنظیمات پورتهای را انجام می دهیم که از پروتکل I2C جهت تبادل اطلاعات با ماژول تاریخ و زمان استفاده می کند. پورت B را انتخاب می کنیم. بعد تعیین می کنیم کدام پایه پورت B به SDA (پورت داده) و SCL (پورت کلاک) ماژول وصل شود. از قسمت پایین اسم ماژول DS1307 را انتخاب می کنیم. آن را Enabled می کنیم و خروجی آن را نیز ۰ قرار می دهیم.

در قسمت LCD , Alphanumeric LCD را enable می کنیم. در قسمت Characters/Line تعیین می کنیم در هر خط LCD چند کاراکتر نوشته شود. (۸ تا). در AVR LCD Module تعیین می کنیم هر پایه LCD به کدام پایه از پورت C میکرو ما وصل شود. در آخر از منو بالای صفحه گزینه Genarete, save and exit را انتخاب می کنیم. در آخر سه بار save می زنیم تا سه فایل (فایل hex, فایل کد ویژن، فایل زبان C) را به ما بدهد.

- کد :

```
#include <mega16a.h> کتابخانه Atmega16A
```

```
#include <i2c.h> کتابخانه پروتکل i2c
```

```
#include <ds1307.h> کتابخانه ماژول ds1307
```

```
#include <alcd.h> کتابخانه LCD
```

#include <delay.h> کتابخانه delay

#include <stdlib.h> برای استفاده از تابع itoa

unsigned char h,m,s,l,v; تعدادی متغیر برای ساعت و دقیقه و ثانیه و ... تعریف کرده ایم.

unsigned char wday,day,month,year; متغیر برای روز و ماه و سال

char resh[10]; تعدادی رشته برای نگهداری عدد ساعت و ماه و سال تعریف کرده ایم.

char hresh[5];

char mresh[5];

char sresh[5];

//-----

//-----

void settime(void) تابع تنظیم ساعت

{

unsigned char l;

//-----

lcd_clear(); LCD را پاک می کنیم.

lcd_puts("SET HOUR"); رشته SET HOUR را نمایش می دهد.

delay_ms(1000); یک ثانیه صبر می کند.

l=1; متغیر l را برابر ۱ می کنیم تا وارد حلقه زیر شویم.

while (l)

{

```

lcd_gotoxy(0,1);    در LCD یک خط پایین می آید.

itoa(h,resh);    عدد ساعت را که از ماژول ds1307 دریافت کردیم به رشته تبدیل می کند.

lcd_puts(resh);    رشته را روی LCD نمایش می دهد.

lcd_puts(":");    یک ":" چاپ می کند.

itoa(m,resh);    دقیقه را به رشته تبدیل می کند.

lcd_puts(resh);    رشته را روی LCD چاپ می کند.

lcd_puts(":");    یک ":" چاپ می کند.

itoa(s,resh);    ثانیه را به رشته تبدیل می کند.

lcd_puts(resh);    رشته را روی LCD چاپ می کند.

lcd_puts(" ");

delay_ms(500);    نیم ثانیه صبر می کند بعد از اجرای مجدد هر حلقه

    اگر دکمه افزایش زده شد چک می کند h (ساعت) کمتر از h++;
    ۲۴ باشد بعد آن را یک واحد افزایش می دهد.

    اگر دکمه کاهش زده شد چک می کند h (ساعت) بیشتر از h--;
    ۰ باشد بعد آن را یک واحد کاهش می دهد.

    اگر دکمه تنظیمات زده شد l را برابر ۰ می کند تا از این حلقه خارج شویم
    if (PIND.0==0)l=0;
}

//-----

این قسمت برای تنظیم دقیقه است که کاملاً شبیه بالا است و فقط شرط ها انتهای آن فرق دارد.

lcd_clear();

lcd_puts("SET MINUTE");

```

```
delay_ms(1000);
```

```
l=1;
```

```
while (l)
```

```
{
```

```
    lcd_gotoxy(0,1);
```

```
    itoa(h,resh);
```

```
    lcd_puts(resh);
```

```
    lcd_puts(":");
```

```
    itoa(m,resh);
```

```
    lcd_puts(resh);
```

```
    lcd_puts(":");
```

```
    itoa(s,resh);
```

```
    lcd_puts(resh);
```

```
    lcd_puts("  ");
```

```
    delay_ms(500);
```

```
    if (PIND.5==0 & m<60 )m++; اگر دکمه افزایش زده شد چک می کند m (دقیقه) کمتر از 60 باشد و m را یک واحد افزایش می دهد.
```

```
    •
```

```
    if (PIND.6==0 & m>0 )m--; اگر دکمه کاهش زده شد چک می کند m (دقیقه) بیشتر از 0 باشد و m را یک واحد افزایش می دهد.
```

```
    •
```

```
    if (PIND.0==0)l=0;
```

```
}
```

تنظیمات ثانیه مانند بالا است و فقط شرط های آن فرق دارد.

```
lcd_clear();
```

```
lcd_puts("SET SECOND");
```



```

delay_ms(1000);
l=1;
while (l)
{
    lcd_gotoxy(0,1);
    itoa(h,resh);
    lcd_puts(resh);
    lcd_puts(":");
    itoa(m,resh);
    lcd_puts(resh);
    lcd_puts(":");
    itoa(s,resh);
    lcd_puts(resh);
    lcd_puts(" ");
    delay_ms(500);
    if (PIND.5==0 & s<60 )s++;
    if (PIND.6==0 & s>0 )s--;
    if (PIND.0==0)l=0;
}

```

تنظیمات جدید ما را در ماژول ds1307 ذخیره می کند. rtc_set_time(h,m,s);

LCD را پاک می کند. lcd_clear();

رشته SAVE SETTING را چاپ می کند. lcd_puts("SAVE SETTINGS");

delay_ms(1000);

}

```
void setdate(void)
```

```
{  
    زیر برنامه تنظیم تاریخ
```

```
    unsigned char l;
```

```
    //unsigned char c;
```

```
    //char s1[5];
```

```
    lcd_clear();
```

```
    lcd_puts("SET YEAR");
```

```
    delay_ms(1000);
```

```
    l=1;
```

```
    while (l)
```

```
{تنظیمات سال
```

```
    lcd_gotoxy(0,1);
```

```
    itoa(year,resh); عدد سال دریافت شده از ماژول تاریخ زمان را به رشته تبدیل می کند.
```

```
    lcd_puts(resh);
```

```
    lcd_puts(":");
```

```
    itoa(month,resh); عدد ماه دریافت شده از ماژول تاریخ زمان را به رشته تبدیل می کند.
```

```
    lcd_puts(resh);
```

```
    lcd_puts(":");
```

```
    itoa(day,resh); عدد روز دریافت شده از ماژول تاریخ زمان را به رشته تبدیل می کند.
```

```
    lcd_puts(resh);
```

```
    lcd_puts("  ");
```

```
    delay_ms(500);
```

اگر دکمه افزایش زده شد با چک کردن شرط سال را year++;
افزایش می دهد.

اگر دکمه کاهش زده شد با چک کردن شرط سال را year--;
کاهش می دهد.

```
if (PIND.0==0)l=0;  
}
```

تنظیمات ماه

```
lcd_clear();  
lcd_puts("SET MONTH");  
delay_ms(1000);  
l=1;  
while (l)  
{  
    lcd_gotoxy(0,1);  
    itoa(year,resh);  
    lcd_puts(resh);  
    lcd_puts(":");  
    itoa(month,resh);  
    lcd_puts(resh);  
    lcd_puts(":");  
    itoa(day,resh);  
    lcd_puts(resh);  
    lcd_puts(" ");  
    delay_ms(500);
```

```
if (PIND.5==0 & month<60 )month++;  
if (PIND.6==0 & month>0 )month--;  
if (PIND.0==0)l=0;  
}
```

تنظیمات روز

```
lcd_clear();  
lcd_puts("SET DAY");  
delay_ms(1000);  
l=1;  
while (l)  
{  
    lcd_gotoxy(0,1);  
    itoa(year,resh);  
    lcd_puts(resh);  
    lcd_puts(":");  
    itoa(month,resh);  
    lcd_puts(resh);  
    lcd_puts(":");  
    itoa(day,resh);  
    lcd_puts(resh);  
    lcd_puts("  ");  
    delay_ms(500);  
    if (PIND.5==0 & day<24 )day++;  
    if (PIND.6==0 & day>0 )day--;  
    if (PIND.0==0)l=0;
```

```

}

rtc_set_date(wday,day,month,year);
lcd_clear();
lcd_puts("SAVE SETTINGS");
delay_ms(1000);
}

```

زیر برنامه منو

```
void menu()
```

```
{
```

```
l=1;
```

```
while(l)
```

```
{
```

```
lcd_clear();
```

if (v==0) اگر دکمه تنظیمات زده شود v برابر ۰ شده و وارد شرط زیر می شود

```
{
```

lcd_puts(">>>>set time<<<<"); رشته مقابل را در LCD چاپ می کند.

lcd_gotoxy(0,1); به خط بعدی می رود.

lcd_puts(" set date "); رشته set date را چاپ می کند.

```
delay_ms(1000);
```

```
}
```

```
else
```

{ شبیه بالاست }

```
lcd_puts("  set time  ");
```

```
lcd_gotoxy(0,1);
```

```
lcd_puts(">>>>set date<<<<");
```

```
delay_ms(1000);
```

```
}
```

```
if (PIND.5==0)v=0;      اگر دکمه وسط زده بشه v برابر ۰ میشه
```

```
if (PIND.6==0)v=1;      اگر دکمه آخر زده بشه v برابر ۱ میشه
```

```
if (PIND.0==0 & v==0) set time  (از منو ۰ برابر v که حالی در بشه زده تنظیمات زده بشه در حالی که v برابر ۰ (از منو
```

انتخاب شده) باشد وارد شرط می شویم.

```
{
```

```
settime();  زیر برنامه اجرا می شود.
```

```
l=0;
```

```
}
```

```
if (PIND.0==0 & v==1) set date  (از منو ۱ برابر v که حالی در بشه زده بشه در حالی که v برابر ۱ (از منو
```

انتخاب شده) باشد وارد شرط می شویم.

```
{
```

```
setdate();  زیر برنامه اجرا می شود.
```

```
l=0;
```

```
}
```

```
}
```

```
}
```

```
void main(void)                      زیر برنامه main
```

```
{ در این قسمت پورت ها و رجیستر های آن ها تنظیم می شود. }
```

```
PORTA=0x00;
```

```
DDRA=0x00;
```

```
PORTB=0x00;
```

```
DDRB=0x00;
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
PORTD=0x00;
```

```
DDRD=0x00;
```

```
//I2C Bus initialization
```

```
//I2C Port: PORTB
```

```
//I2C SDA bit: 1                      پورت دیتا به پایه ۱
```

```
//I2C SCL bit: 0                      پورت کلاک به پایه ۰
```

```
//Bit Rate: 100 kHz                      فرکانس کاری.
```

```
//Note: I2C settings are specified in the
```

```
//Project|Configure|C Compiler|Libraries|I2C menu.
```

```
i2c_init(); I2C مقدار دهی اولیه می شود. این که پورت های دیتا و کلاک به کدام پایه ها وصل شود؛
```

```
//DS1307 Real Time Clock initialization
```

```
//Square wave output on pin SQW/OUT: Off
```

```
//SQW/OUT pin state: 0
```

```
rtc_init(0,0,0);          ماژول ds1307 مقداردهی اولیه می شود.
```

```
//Alphanumeric LCD initialization
```

```
//Connections are specified in the
```

```
//Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
```

```
//RS - PORTC Bit 0
```

```
//RD - PORTC Bit 1
```

```
//EN - PORTC Bit 2
```

```
//D4 - PORTC Bit 4
```

```
//D5 - PORTC Bit 5
```

```
//D6 - PORTC Bit 6
```

```
//D7 - PORTC Bit 7
```

```
//Characters/line: 8
```

```
lcd_init(16);            LCD با ۱۶ کاراکتر در دو سطر پیکربندی می شود.
```

```
while (1)
```

```
{
```

```
    rtc_get_time(&h,&m,&s);          اعداد ساعت و دقیقه و ثانیه را از ds1307 می خواند.
```

```
    rtc_get_date(&wday,&day,&month,&year); ds1307 را از اعداد سال و ماه و روز
```

```
می خواند.
```

```
    lcd_clear();
```


عدد را به رشته تبدیل می کند. itoa(h,hresh);

itoa(m,mresh);

itoa(s,sresh);

رشته را در LCD نشان می دهد. lcd_puts(hresh);

lcd_puts(":");

lcd_puts(mresh);

lcd_puts(":");

lcd_puts(sresh);

lcd_gotoxy(0,1);

برای چاپ سال دو رقم ۲۰ اول آن را خودمان جدا چاپ می کنیم. lcd_puts("20");

عدد سال به رشته تبدیل می شود. itoa(year,hresh);

عدد ماه به رشته تبدیل می شود. itoa(month,mresh);

عدد روز به رشته تبدیل می شود. itoa(day,sresh);

lcd_puts(hresh);

بین عدد سال و ماه و روز علامت "/" را چاپ می کند. lcd_puts("/");

lcd_puts(mresh);

lcd_puts("/");

lcd_puts(sresh);

بین هر باز اجرای loop ، ۹۰۰ میلی ثانیه توقف داریم. delay_ms(900);

اگر دکمه تنظیمات (دکمه اول) زده بشه وارد منو می شویم. if (PIND.0==0)menu();

}

}

