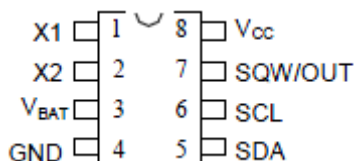


# ساخت ساعت دقیق « RTC »

## بکمک آی سی DS1307

آی سی های DS1307 که ساخت شرکت Dallas Semiconductor می باشد یک آی سی برای طراحی ساعت و تقویم دیجیتال استفاده می شود و بسیار دقیق می باشند . از این آی سی زمانی استفاده می شود که ما نمی خواهیم قسمتی از میکرووی ما مشغول محاسبه زمان شود . همچنین با استفاده از یک باتری پشتیبان می توان اطلاعات را در این آی سی نگهداری کرد البته دقت بالا این آی سی نیز مورد توجه است . این قطعه توسط ارتباط I2C کار می کند . پایه های این IC در زیر آمده است :



شکل 1-13

**Vcc** و **GND** : تغذیه آی سی می باشند .

**X1** و **X2** : یک اسیلاتور 32.768 KH بین این دو پایه قرار می گیرد .

## ریزپردازنده AVR

**VBAT**: این پایه را باید به یک باتری پشتیبان وصل کرد تا در هنگام قطع برق اطلاعات حفظ شوند. اما در هنگام شبیه سازی در Protuse این پایه را با یک مقاومت زمین کنید.

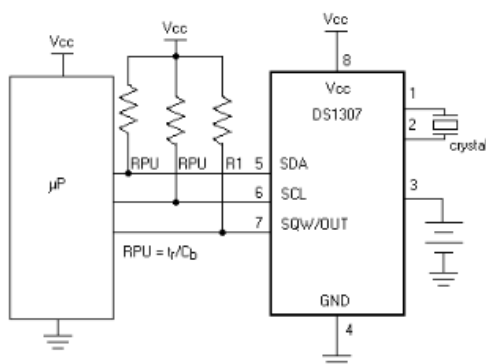
**SDA و SCL**: همانطور که گفتیم ارتباط این آی سی از نوع I2C می باشد و این دو پایه ، پایه های ارتباط می باشند. توجه کنید که این دو پایه باید توسط دو مقاومت Pull UP شوند .

**SQW/OUT**: اگر بیت SQWE در رجستر کنترلی این آی سی یک باشد یک موج مربعی بر روی این پایه ایجاد می شود که فرکانس آن توسط جدول زیر تعیین می شود . اگر قصد ندارید از این پایه استفاده کنید آن را توسط یک مقاومت یا زمین کنید و یا به منبع تغذیه وصل کنید .

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

جدول 1-13

در زیر نحوه وصل این تراشه به میکرو آمده است :



شکل 2-13

## رجیستر کنترلی DS1307

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

**RS0 و RS1**: همانطور که گفته شد جهت تعیین فرکانس SQW می باشد .

**SQWE**: چنانچه گفته شد یک شدن این بیت سبب ایجاد یک موج مربعی بر روی پایه SQW می شود .

**OUT**: در صورتی بیت SQWE صفر باشد بیت OUT وضعیت پایه SQW را مشخص می کند یعنی اگر صفر باشد پایه SQW هم صفر می شود .

## کتابخانه ds1307.h

**void rtc\_init( char rs, char sqwe, char out)**

جهت آماده سازی می باشد و باید قبل ارتباط باید این دستور نوشته شود . rs فرکانس موج مربعی SQW را مشخص می کند .

**void rtc\_get\_time(char &hour, char &min, char &sec)**

برای خواندن ساعت از آی سی می باشد .

**void rtc\_set\_time(char hour, char min, char sec)**

برای تنظیم ساعت می باشد و دادن مقدار اولیه به آن می باشد .

## void rtc\_get\_date(char &date, char &month, char &year)

برای خواندن تاریخ از آی سی استفاده می شود .

## void rtc\_set\_date(char date, char month char year)

جهت خواندن تاریخ استفاده می شود .

مثال : برنامه ساعت توسط DS1307

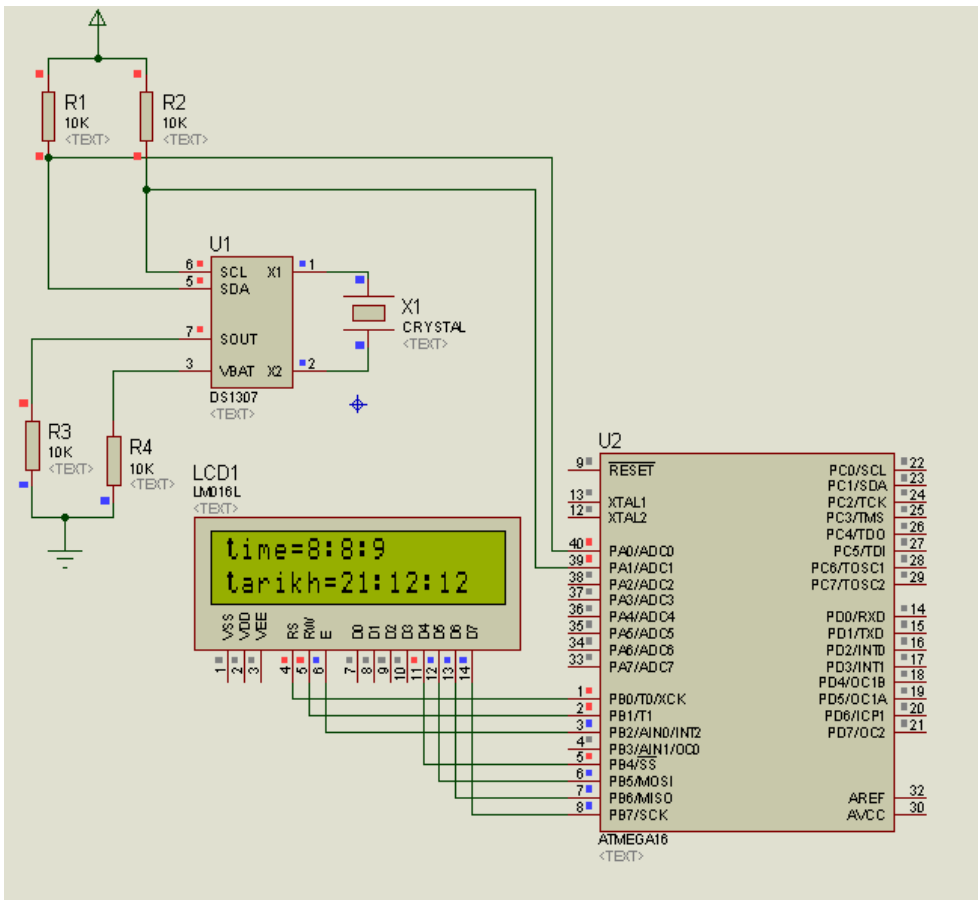
```
#include <mega16.h> // میکرو مورد استفاده
#include <stdio.h> //stdliB کتابخانه فراخوانی
#asm // شروع برنامه اسمبلی
.equ __i2c_port=0x1B ;PORTA // تعیین پورت A به عنوان پورت ارتباط دوسیمه
.equ __sda_bit=0 //SDA معرفی پین صفر پورت A به عنوان پایه
.equ __scl_bit=1 //SCL معرفی پین صفر پورت A به عنوان پایه
#endasm // پایان برنامه اسمبلی
#include <i2c.h> // معرفی کتابخانه ارتباط دوسیمه
#include <ds1307.h> //DS1307 فراخوانی کتابخانه
#asm // شروع برنامه اسمبلی
.equ __lcd_port=0x1B ;PORT // معرفی پورت A به عنوان پورت نمایشگر
#endasm // پایان برنامه اسمبلی
#include <lcd.h> // معرفی کتابخانه
#include <delay.h> // معرفی کتابخانه تاخیر زمانی
```

```

unsigned char a,b,c,r,m,s,zaman[13],tarikh[13]; // معرفی کتابخانه تاخیر زمانی
void main(void) { // برنامه اصلی
i2c_init(); // آماده سازی ارتباط دوسیمه
rtc_init(0,0,1); // آماده سازی آی سی برای برقراری ارتباط
lcd_init(16); // آماده سازی نمایشگر
rtc_set_time(8,8,8); // دادن مقدار اولیه به آی سی
rtc_set_date(21,12,12); // دادن مقدار اولیه تاریخ
while (1) { // حلقه بینهایت
    rtc_get_time(&a,&b,&c); // خواندن ساعت
    sprintf(zaman,"time=%d:%d:%d",a,b,c); // تبدیل ساعت خوانده شده به رشته
    lcd_gotoxy(0,0); // رفتن به سطر وستون مورد نظر
    lcd_puts(zaman); // نمایش ساعت
    lcd_putsf(" "); // حذف کاراکترهای باقیمانده از نمایش قبلی
    lcd_gotoxy(0,1); // رفتن به سطر دوم
    rtc_get_date(&r,&m,&s); // خواندن تاریخ
    sprintf(tarikh,"tarikh=%d:%d:%d",r,m,s); // تبدیل تاریخ خوانده شده به رشته
    lcd_puts(tarikh); // نمایش تاریخ
    lcd_putsf(" "); // حذف کاراکترهای باقیمانده از نمایش قبلی
} }

```

## ریزپردازنده AVR



شکل 3-13