به نام خدا

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

عنوان:

پروژه دربازکن الکترونیکی خانه

همراه با ساعت، تاریخ شمسی و اندازه گیری دمای بیرون

طراحی و برنامه نویسی: سید محسن قاسمیان

چکیــده

این پروژه را بعد از ساخت، می‌توانید به راحتی در کنار آیفون منزل تان نصب کنید که شامل یک میکروکنترلر ATMEGA8 ، یک نمایشگر LCD و یک کیبورد می باشد. بر روی نمایشگر اطلاعاتی هم چون دما، ساعت دقیق، تاریخ شمسی، و روز هفته نمایش داده می شود.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

کلمات کلیدی

قفل رمز, ساعت, تاریخ شمسی, میکروکنترلر avr , زبان سی, کدویژن, پروتکل I2C , دما, دمای منفی, دمای مثبت

فهرست مطالب

با کلیک کردن روی هر یک از عنوان زیر به موضوع مربوطه هدایت می‌شوید.

فهرست مطالب ۳
فهرست شکل‌ها ۴
۱ مقدمه ۵
۱-۱ هدف ۵
۲ اصول کار سیستم ۶
۳ برنامه نویسی C ۸
۳-۱ معرفی متغیر ها و کاربرد آنها ۸
۳-۲ توابع موجود در فایل اصلی ۹
۳-۲-۱ تابع خواندن دما به صورت وقفه ۹
۳-۲-۲ تابع دریافت دما و تبدیل به سانتیگراد ۹
۳-۲-۳ تابع ارسال اطلاعات به نمایشگر ۱۰
۳-۲-۴ تابع کنترل نور نمایشگر ۱۰
۳-۲-۵ تابع اسکن صفحه کلید ۱۱
۳-۲-۶ تابع بازنشانی متغیر های مهم ۱۲
۳-۲-۷ تابع نمایش پیغام ذخیره سازی ۱۲
۳-۲-۸ تابع دریافت رمز عبور ۱۳
۳-۲-۹ تابع ساختن رمز اولیه بعد از پروگرم کردن ۱۴
۳-۲-۱۰ تابع چک کردن رمزعبور ۱۵
۳-۲-۱۱ تابع تغییر رمز عبور ۱۶
۳-۲-۱۲ تابع تغییر ساعت و تاریخ ۱۷
۳-۲-۱۳ تابع اصلی برنامه ۱۹
۳-۳ توابع موجود در فایل هدر ۲۱
۳-۳-۱ متغیرهای سراسری این هدر ۲۱
۳-۳-۲ تابع تقسیم ساده ۲۱
۳-۳-۳ تابع تبدیل تاریخ میلادی به شمسی ۲۲
۳-۳-۴ تابع بدست آوردن روز هفته ۲۳
۳-۳-۵ تابع تعریف کاراکتر خاص ۲۳
۳-۳-۶ تابع کاراکتر فرصت های باقی مانده ۲۴
۳-۳-۷ تابع نمایش فارسی روز هفته ۲۵
۳-۳-۸ تابع انتخاب روز هفته و نمایش آن ۲۶
۳-۳-۹ تابع دریافت تاریخ و ساعت ۲۷
۴ مراجع ۲۸
۵ اطلاعات مقاله و نرم افزار ها ۲۹

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

فهرست شکل‌ها

شکل ۱-۱: شماتیک مدار ۶

ıoo.com

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۱ مقدمه

این پروژه در اصل یک قفل امنیتی است که می توان جلوی درب اصلی منزل (در کنار آیفون) آن را نصب کرد. از امکانات آن می توان به موارد زیر اشاره نمود:

نمایش دما از ۱۲۵ تا ۵۵- درجه سانتیگراد، نمایش ساعت دقیق، نمایش تاریخ شمسی همراه با روز هفته به صورت فارسی، نمایش تعداد فرصت های باقیمانده برای وارد کردن رمز صحیح. و قفل شدن سیستم به مدت ۱۵ دقیقه در صورتی که ۴ بار رمز اشتباه وارد شود. امکان تغییر رمز و ست کردن ساعت و تاریخ و ...

۱-۱ هدف

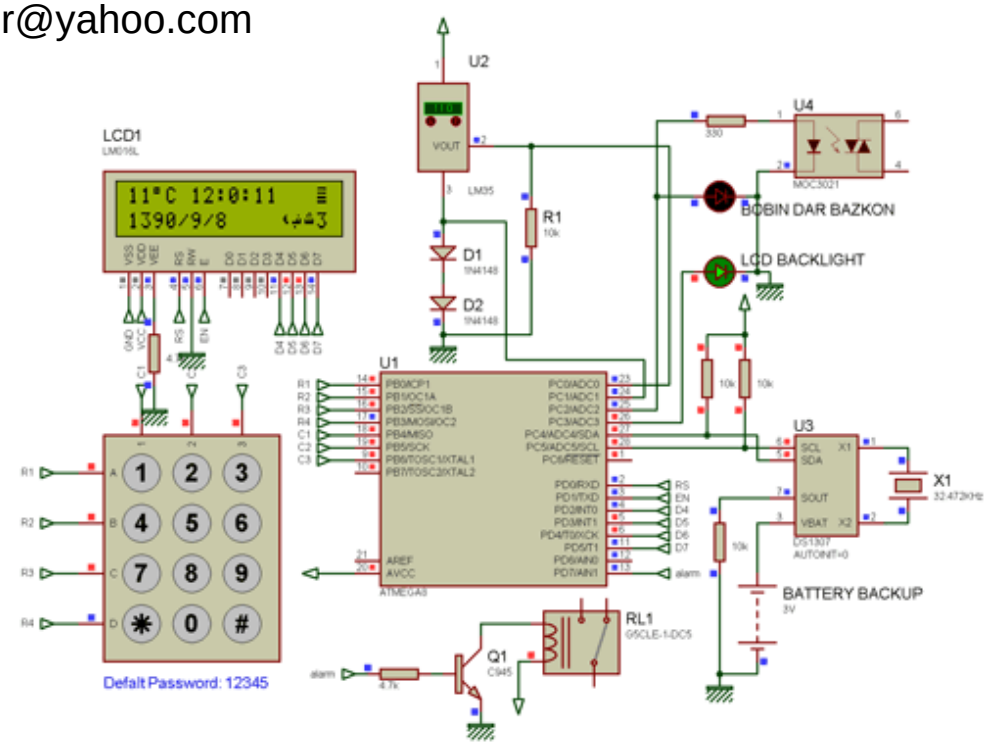
هدف از انجام این پروژه آشنایی با آیسی ساعت DS1307 و کار با پروتکل I2C است که جهت ارتباط با این آیسی استفاده شده است. نحوی خواندن و وارد کردن اطلاعات به DS1307 و همچنین توابع تبدیل تاریخ میلادی به شمسی و سایر توابع وابسته. مدیریت زمان در میکروکنترلر و سرکشی به تمامی وظایف بدون ایجاد کوچکترین وقفه زمانی و همچنین نوشته برنامه به صورت توابع و فراخوانی تو در توی آنها را می توان از جمله ویژگی های آموزشی این پروژه به شمار آورد.

اهداف آینده: با ارتقاع سخت افزار میکروکنترلر و ایجاد توابعی خاص به راحتی می توان برای هر نفر یک نام کاربری ایجاد کرد و تمامی ورود و خروج های اشخاص را همراه با ساعت و تاریخ در سیستم ثبت کرد و همه اینها توسط یک نفر که Admin می باشد قابل روئیت و کنترل است.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۲ اصول کار سیستم

sbargh.ir
@sbargh
sbargh.ir@yahoo.com



شکل ۱-۱: شماتیک مدار

در صورتی که برای اولین بار بعد از پروگرم کردن مدار راه اندازی شود ابتدا از شما درخواست می شود که به عنوان Admin یک رمز برای سیستم انتخاب کنید، که بعد از وارد کردن یک رمز ۵ رقمی سیستم پیغام ذخیره سازی یا عدم ذخیره سازی به شما میدهد که بعد از فشار دادن کلید # سیستم آماده راه اندازی می شود. نکته: در صورتی که این مراحله را انجام ندهید هر بار که سیستم روشن شود از شما درخواست ساختن رمز می کند. بعد از طی کردن این مرحله میکرو درخواستی برای DS1307 مبنی بر ارسال متغیر های ساعت و تاریخ می کند، در صورتی که تاریخ سیستم کوچکتر از ۱۳۹۰ باشد میکرو به صورت پیش فرض تاریخ و ساعت مشخصی را در DS1307 ست می کند و این آیسی راه اندازی می شود، تا زمانی که vcc مدار برقرار باشد. حال اگر باتری پشتیبان در مدار باشد و vcc قطع شود آنگاه DS1307 به طور خودکار تغذیه خود را از باتری تهیه می کند و به کار خود ادامه میدهد. بنابراین بعد از روشن شدن مجدد سیستم میکرو عمل ست کردن را دیگر اجرا نخواهد کرد و وارد منوی اصلی می شویم.

در اینجا دما، ساعت، تعداد فرصت های باقی مانده، تاریخ شمسی و روز هفته نمایش داده می شود. برای باز کردن درب ابتدا کلید # را فشار می‌دهیم سپس از شما درخواست رمز عبور می شود، در صورتی که رمز صحیح باشد اپتوکوپلر به مدت ۵۰۰ میلی تحریک شده و درنتیجه برق ۱۲ ولت **AC** بوبین دربازکن تحریک می شود. (نکته: ولتاژی که اپتوکوپلر سوئیچ می کند باید AC باشد)

در هر مرحله که درخواست رمز می شود اگر هنگام وارد کردن رمز اشتباهی کردید می توانید از کلید * برای پاک کردن اطلاعات وارد شده استفاده نمائید. برای تغیر رمز سیستم وقتی که در منوی اصلی هستید کلید * را فشار دهید که در این حالت باید ابتدا رمز قبلی وارد کنید و بعد هم رمز جدید در صورتی که رمز قبلی صحیح باشد پیغامی مبنی بر ذخیره رمز جدید به شما می دهد که در صورت تایید کردن (#) رمز جدید جایگزین خواهد شد.

برای تنظیم کردن تاریخ و ساعت سیستم در منوی اصلی کلید ۱ را فشار دهید، در اینجا از شما رمز عبور درخواست می شود در صورتی که رمز صحیح باشد به منوی تنظیم ساعت خواهید رفت که باید به این فرم اعداد را وارد کنید، مثلا برای وارد کردن عدد ۸ دقیقه باید ۰۸ وارد کنید و نکته مهم این است که تاریخ را باید برحسب **میلادی** وارد کنید. بعد از وارد کردن این ۶ پارامتر از شما سوال می شود که ذخیره انجام شود یا نه. (# بله * خیر)

در صورتی که تا ۱۵ ثانیه هیچ کلیدی را فشار ندهید LED پس زمینه LCD خاموش خواهد شد و بعد از آن با فشار دادن هر کلیدی مجدد روشن خواهد شد. پین PD.7 که alarm نام دارد در صورت نیاز می توانید با آن یک رله را تحریک کنید و خروجی های رله را به شاسی زنگ آیفون وصل کنید. در این صورت اگر کسی به دنبال پیدا کردن رمز عبور باشد در صورتی که ۴ بار رمز اشتباه وارد کند قبل از قفل شدن سیستم ابتدا با ریتم خاصی سه مرتبه بوق آیفون منزل شما را به صدار می آورد و شما متوجه این امر خواهید شد.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳ برنامه نویسی C

۳-۱ معرفی متغیر ها و کاربرد آنها

در ابتدا یک سری تعریف ها و فراخوانی توابع مختلف نوشته شده است و بعد هم متغیر های ما که همه از نوع سراسری هستند یعنی در هر تابعی می توان از محتوای آنها استفاده کرد که به شرح زیر می‌باشد:

```
eeprom unsigned char prog, pass[5];
 char buffer[32];
 unsigned char len_pass=5, pass_in[5], old_pass[5], key, op=0;
 signed temp;
 unsigned char err_cunt, err_cun_num=4, wait_scan_keypad=15; //secound
 unsigned int lcd_bl;
 unsigned char scan[4]={0XFE,0XFD,0XFB,0XF7};
 unsigned char arrkey[16]={
  1,2,3,15,
  4,5,6,14,
  7,8,9,13,
  10,0,11,12};
```

متغیر prog برای ساختن اولین رمز توسط Admin است و بعد مقدار آن به ۸۰ تغییر میکند تا این تابع دیگر هرگز اجرا نشود. آرایه پنج عضوی pass رمز ۵ رقمی مورد نظر در آن قرار میگیرد و هر دوی این متغیرها در حافظه EEPROM میکرو تعریف شده اند. متغیر buffer برای نگهداری اطلاعات ارسالی به نمایشگر است. len_pass طول رمز عبور را مشخص می‌کند که در اینجا ۵ رقم می باشد که باید برابر با دو آرایه زیر باشد. pass_in و old_pass آرایه ای است ۵ عضوی برای نگهداری رمزهای وارد شده توسط کاربر است. Key ارزش کلید فشار داده شده در آن قرار میگیرد. op که مقدار پیشفرض آن صفر می باشد برای عملیات خاص در نظر گرفته شده که بیشتر در تابع تغییر رمز مورد استفاده قرار گرفته است. Temp متغیر علامتدار برای نگهداری مقدار دمای خوانده شده می باشد. err_cunt تعداد دفعات رمزهایی که اشتباه وارد می شود را می شمارد. err_cun_num مجوز تعداد دفعات وارد کردن رمز توسط کاربر است که بصورت پیش فرض ۴ بار است. wait_scan_keypad مدت زمانی است که اگر کاربر در این زمان تعیین شده (پیش فرض ۱۵ ثانیه) رمز خود را وارد نکند سیستم تقاضای او را نادید گرفته و به منوی اصلی می رود. lcd_bl یک شمارنده می باشد و در صورتی که تا زمان مشخص شده ای کلیدی فشار داده نشود LED پس زمینه LCD خاموش می‌شود. آرایه scan و arrkey برای اسکن کیبورد و یافتن مقدار کلید فشار داده شده است.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳-۲ توابع موجود در فایل اصلی

۳-۲-۱ تابع خواندن دما به صورت وقفه

```
unsigned int read_adc(unsigned char adc_input){
  ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
  delay_us(10);
  ADCSRA|=0x40;
  while ((ADCSRA & 0x10)==0);
  ADCSRA|=0x10;
  return ADCW;
}
```

این تابع به صورت وقفه کار می‌کند و دارای یک ورودی adc_input است که تعین می کند از کدام کانال مقدار را دریافت کند بعد از اتمام زمان تبدیل مقدار تبدیل شده به دیجیتال که در ADCW است برگشت داده می شود تا سایر عملیات روی آن انجام شود.

۳-۲-۲ تابع دریافت دما و تبدیل به سانتیگراد

```
void get_temp(){
 signed int a,b;
 float t1,t2;
 b=read_adc(0);
 a=read_adc(1);
 t1=((b*2.56)/1023)*100;
 t2=((a*2.56)/1023)*100;
 temp=t1-t2;
}
```

در این تابع دو مقدار آنالوگ از دو کانال ۰ و ۱ دریافت می‌شود که کانال صفر مقدار دمای سنسور را می خواند و از کانال یک هم مقدار ولتاژ دو سر زنر ها را می خوانیم و با کم کردن این دو مقدار از یکدیگر دمای اصلی در temp قرار می گیرد. عمل تبدیل به سانتیگراد هم خیلی واضح است با ضرب و تقسیم مقادیر خوانده شده بدست می آید، مقدار ۲,۵۶ به این علت است که ولتاژ مرجع داخلی ۲,۵۶ ولت انتخاب شده است. همانگونه که می‌دانیم رفتار یک دیود زنر در دما محیط با ولتاژ دو سر آن رابطه دارد بنابراین از این خاصیت همراه با سنسور LM35 برای اندازه گیری دمای منفی استفاده شده.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳-۲-۳ تابع ارسال اطلاعات به نمایشگر

```
  void display(){
    get_temp();
    sprintf(buffer,"%d\xdfC
%d:%d:%d\n%d/%d/%d",temp,hour,minute,second,jy,jm,jd);
    lcd_clear();
    lcd_puts(buffer);
    week_name();
    cunter_err_pass(err_cunt);
  }
```

این تابع متغیرهای مهم و سراسری را با فرمت مشخص شده در متغیر buffer آماده سازی کرده و سپس برای نمایشگر ارسال می‌شود. توابع week_name و cunter_err_pass به ترتیب روز هفته را به صورت فارسی آماده کرده و تابع بعدی تعداد دفعات باقی مانده برای وارد کردن رمز عبور را آماده و روی نمایشگر چاپ می کند.

۴-۲-۳ تابع کنترل نور نمایشگر

```
  void lcd_back_light(){
    lcd_bl++;
    if(lcd_bl>=250) lcd=0;
      else lcd=1;
  }
```

با هر بار فراخوانی این تابع یک واحد به متغیر lcd_bl افزوده می شود، زمانی که به عدد ۲۵۰ رسید LED مربوطه خاموش می‌شود و غیر این صورت روشن می‌شود. این تابع، خود در تابع keypad هر ۱۰۰ میلی ثانیه اجرا می شود بنابراین ۱۰۰×۲۵۰ حدودا ۲۵ ثانیه خواهد شد یعنی اگر تا ۲۵ ثانیه هیچ کلیدی فشار داده نشود LED مربوطه خاموش می‌شود. نکته: متغیر lcd_bl با هر بار فشار دادن یک کلید مقدار آن در تابع keypad صفر خواهد و به اصلاح زمان شارژ می شود.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳-۲-۵ تابع اسکن صفحه کلید

```
unsigned char keypad(unsigned char wait){
unsigned char r,c,k=0,j;
unsigned int i,l;
 DDRB=0X0F;
 PORTB=0XFF;
 l=(wait*1000)+1000;

    for (i=0; i<l; i++){
      for (r=0; r<4; r++){
       c=4;
       PORTB=scan[r];
       delay_us(10);
       if(c1==0) c=0;
       if(c2==0) c=1;
       if(c3==0) c=2;
       if(c4==0) c=3;

        if (!(c==4)){
         k=arrkey[(r*4)+c];
         while(c1==0);
         while(c2==0);
         while(c3==0);
         while(c4==0);
         delay_ms(100);
         lcd_bl=0;
         return k;
        }
      }
      delay_ms(1);
      j++;
      if(j==100) {lcd_back_light(); j=0;}
    }

 k=100; //time out  "Not press any key"
 return k;
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

وظیفه این تایع اسکن صفحه کلید به صورت ماتریسی است که دارای یک پارامتر ورودی به نام wait است. تابع صورت پیش فرض به مدت ۱ ثانیه صفحه کلید را اسکن می‌کند و بر می‌گردد حالا اگر هنگام فراخوانی این تابع مثلا عدد ۱۵ را بفرستیم آنگاه ۱۵+۱ برابر ۱۶ ثانیه صفحه کلید را اسکن خواهد کرد. نکته مهم این است که اگر در این مدت زمان هیچ کلیدی فشرده نشود تابع مقدار ۱۰۰ را برگشت خواهد داد که در ادامه برنامه ما با عدد ۱۰۰ متوجه خواهیم شد که زمان وارد کردن اطلاعات پایان یافته و کلیدی فشرده نشده بنابراین از عملیات مربوطه صرف نظر می‌کنیم.

در اینجا فقط به این نکته اشاره خواهیم کرد تابع از دو حلقه تو در تو تشکیل شده است که حلقه داخلی وظیفه اسکن پورت را بر عهد دارد و حلقه بیرونی مدت زمان اسکن را مشخص می کند، به این صورت که حلقه بیرونی حلقه داخلی را هر ۱ میلی

ثانیه یک بار اجرا می‌کند. در صورتی که کلیدی فشرده شود شرط داخل حلقه داخلی اجرا می‌شود و مقدار واقعی کلید از آرایه انتخاب می‌شود، همچنین lcd_bl صفر شده تا زمان روشن بودن LED شارژ شود و همان جا مقدار k برگشت داد می شود و CPU به ادامه برنامه بازگشت داده خواهد شد.

۳-۲-۶ تابع بازنشانی متغیر های مهم

```
void reset_var(){
  unsigned char i;
  for (i=0; i<len_pass; i++){
    pass_in[i]=255;
    old_pass[i]=255;
  }
}
```

وظیفه این تابع بازنشانی و پر کردن کردن متغیر مهم سیستم با عدد FF می‌باشد.

۳-۲-۷ تابع نمایش پیغام ذخیره سازی

```
unsigned char masage(){
  lcd_clear();
  lcd_gotoxy(5,0);
  lcd_putsf("SAVE?");
  lcd_gotoxy(0,1);
  lcd_putsf("NO(*)");
  lcd_gotoxy(10,1);
  lcd_putsf("YES(#)");
  key = keypad(wait_scan_keypad);
  return key;
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

در مکان های مختلفی از برنامه نیاز است که از کاربر سوالی مبنی بر انجام ذخیره سازی پرسیده شود، این تابع این عمل را انجام می‌دهد و شروع به اسکن صفحه کلید می‌کند و مقدار کلید فشرده شده را بازگشت خواهد داد که در ادامه برنامه در صورتی که کلید # فشرده شده بود عملیات ذخیره سازی انجام می‌شود و اگر کلید * بود عملیات متوقف خواهد شد.

۳-۲-۸ تابع دریافت رمز عبور

```
void get_pass(){
  unsigned char i,x;
res_get_pass:
  lcd_clear();
  _lcd_write_data(0X0F);
  if(op==0) lcd_putsf("Input Password");
  if(op==1) lcd_putsf("Old Password");
  if(op==2) lcd_putsf("New Password");
  for (i=0; i<len_pass; i++){
      x=i+5;
      lcd_gotoxy(x,1);
      key = keypad(wait_scan_keypad);
      if(key==10) goto res_get_pass;
      if(key==100 || key==11) break;
      pass_in[i]=key;
      sprintf(buffer,"%d",pass_in[i]);
      lcd_puts(buffer);
      delay_ms(150);
      lcd_gotoxy(x,1);
      lcd_putsf("*");
  }
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

در این تابع ابتدا مکان نمای نمایشگر به صورت چشمک زن خواهد شد و در ادامه با توجه به مقدار متغیر op پیغام مربوطه روی نمایشگر نشان داده خواهد شد و بعد از آن به تعداد len_pass از کاربر رمز عبور را دریافت خواهد کرد. همچنین مقدار متغیر key برای اعداد (*)۱۰ و (#) ۱۱ و (Time out) ۱۰۰ چک خواهد شد. در صورتی که مقدار ۱۰ باشد رمز وارد شده پاک شده و به برچسب res_get_pass پرش خواهد کرد و مجددا شما می توانید رمز عبور را وارد کنید. و در صورتی که مقادیر ۱۱ یا ۱۰۰ باشد حلقه شکسته (break) خواهد شد تابع به اتمام می‌رسد. با هر بار وارد کردن یکی از ارقام به مدت ۱۵۰ میلی ثانیه عدد نمایش داده می‌شود و سپس به جای آن کاراکتر "*" نمایش داده خواهد شد.

۳-۲-۹ تابع ساختن رمز اولیه بعد از پروگرم کردن

```
void new_pass(){
    unsigned char i;
    lcd=1;
    if(prog!=80){ // Create a first password
        sprintf(buffer,"Admin: Create a password");
        lcd_clear();
        lcd_puts(buffer);
        delay_ms(3000);
        get_pass();
        i=masage();

      if(i==11){
        for (i=0; i<len_pass; i++){
            pass[i]=pass_in[i];
        }
        prog=80; // burn fuse for a first get password
      }
    }
    reset_var();
  }
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

این تابع تنها یک بار اجرا می‌شود و بعد از ساختن اولین رمز عبور دیگر اجرا نخواهد شد. بعد از پروگرم کردن میکرو محتوای EEPROM عموما FF است، در هر صورت عدد ۸۰ نیست پس این نشان میدهد که هنوز رمز اولیه ساخته نشده است که بعد از ساختن آن این متغیر برابر ۸۰ خواهد شد و در راه اندازی بعدی سیستم این تابع دیگر اجرا نخواهد شد. اصول کار این تابع نیز شبیه تابع دریافت رمز عبور می باشد که در بالا توضیح داده شد.

۳-۲-۱۰ تابع چک کردن رمزعبور

```
void check_pass(unsigned char open){ //open: 1=> rel=1
    unsigned char i,c=1;
    for (i=0; i<len_pass; i++){
        if(pass_in[i]!=pass[i]){
            c++;
            break;
        }
    }

    if(c==1){
      if(open==1){
        rel=1; //cunter for wait invalid password
        lcd_clear();
        lcd_putsf("Opened the door");
        lcd_gotoxy(4,1);
        lcd_putsf("WELCOME");
        delay_ms(500);
        rel=0;
        delay_ms(2000);
      }
      err_cunt=err_cun_num;
      op=50; //op 50 ==> password is true and chane time date
    }else{
       lcd_clear();
       lcd_putsf("INVALID PASSWORD");
       delay_ms(1000);
       err_cunt--;
    }
   reset_var();
  }
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

این تابع دارای یک پارامتر ورودی به نام open می‌باشد در صورتی که مقدار آن ۱ باشد بعد از چک کردن رمز عبور عملیات مربوط به تحریک اپتوکوپلر انجام شده و درب باز می‌شود. اصول کار تابع به این صورت می‌باشد که رمز عبور وارد شده با رمز عبور ذخیره شده مقایسه می‌شود و در صورتی که یکی از آنها مطابقت نداشته باشد یک واحد به متغیر c اضافه می‌شود، بعد از اتمام این مرحله محتوای c تست می‌شود و اگر مخالف مقدار پیش فرض ۱ بود متوجه خواهیم شد که یکی از ارقام رمز عبور اشتباه است بنابراین پیغام INVALID صادر می‌شود و یک واحد از err_cunt کم می‌شود اما در صورتی که مقدار c تغییر نکند یعنی رمز ورودی صحیح بود و درب با توجه به شرط گفته شده باز می‌شود و متغییر err_cunt مجددا با مقدار پیش فرض برای کاربر بعدی بازنشانی می‌شود همچنین op=50 می‌شود تا اگر این تابع برای تغییر ساعت و تاریخ فراخوانی شده بود مجوز تغییر داده شود. و در آخر هم متغییر های مهم بازنشانی می‌شوند.

۳-۲-۱۱ تابع تغییر رمز عبور

```
void change_pass(){
  unsigned char i,c=1;
  op=1; //get old password
  get_pass();
  for (i=0; i<len_pass; i++){
      old_pass[i]=pass_in[i];
  }
  op=2; //get new password
  get_pass();

  for (i=0; i<len_pass; i++){
      if(old_pass[i]!=pass[i]){
          c++;
          break;
      }
  }

  key = masage();
  if(key==11){//# key==11 save is OK Go to cheking...

     if(c==1){
        for(i=0; i<len_pass; i++){
          pass[i]=pass_in[i];
        }
        lcd_clear();
        lcd_putsf("Successfully !");
        delay_ms(3000);
        err_cunt=err_cun_num; //cunter for wait invalid password
      }else{
         lcd_clear();
         lcd_putsf("ERROR!   Wait...");
         delay_ms(5000);
         err_cunt--;
      }

  }
  reset_var(); key=0;
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

در تابع ابتدا op=1 می‌شود و رمز قبلی را از کاربر دریافت کرده سپس مجددا op=2 می‌شود و رمز جدید را از کاربر دریافت می کند، سپس رمز قبلی را با رمز ذخیره شده مقایسه می کند و پیغام ذخیره سازی را نمایش می‌دهد در صورتی که # فشرده شود نتیجه مقایسه رمز را برسی می‌کند اگر c برابر ۱ باشد رمز صحیح بوده و در نتیجه رمز جدید جایگزین رمز ذخیره شده خواهد و پیغام Successfully نمایش داده می‌شود. اما در صورتی که c مخالف ۱ باشد یعنی رمز قدیمی اشتباه است و پیغام ERROR نمایش داده می شود و یک واحد از err_cunt کم می شود. و در آخر هم متغیر های مهم بازنشانی شده و سیستم وارد منوی اصلی می‌شود.

۳-۲-۱۲ تابع تغییر ساعت و تاریخ

```
  void change_time(){
    unsigned char digit[2],newtime[6];    //0h,1m,2s  3y,4mo,5d
    unsigned char x,y=0,j,locate[6]={5,8,11,5,8,13};
    get_pass();
    check_pass(0);
    if(op==50){ //op 50 ==> password is true and chane time is alow
      sprintf(buffer,"Time --:--:--\nDate --/--/20--");
      lcd_clear();
      lcd_puts(buffer);
      _lcd_write_data(0X0F);

      for(x=0; x<=5; x++){ //get 6 number
        digit[0]=digit[1]=0;
        if(x>2) y=1; // Go to next row
        for(j=0; j<=1; j++){ //get 2 digit from keypad
          lcd_gotoxy(locate[x],y);
          digit[j]=keypad(wait_scan_keypad);
          if(j==0) digit[0]*=10; //Digit format xx => digit[0]x
          newtime[x]=digit[0]+digit[1]; //چسپاندن دو عدد به هم و تشکیل عدد دورقمی
          if(digit[j]!=100){ //chek for time out keypad
            sprintf(buffer,"%d",newtime[x]);
            lcd_puts(buffer);
          }else goto endTime; //esc
        }
      }
      key=masage();
      if(key==11){
        gregorian_week_day(newtime[3],newtime[4],(newtime[5]+2000));
//calculate week day
        rtc_set_time(newtime[0],newtime[1],newtime[2]);   //hour, min,
sec  //0h,1m,2s
        rtc_set_date(week,newtime[3],newtime[4],newtime[5]);
//week_day, day ,month, year //3y,4mo,5d
      }else if(key==10) goto endTime;
    }
   endTime:
   key=op=0;
  }
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

این تابع نسبت به تابع های دیگر کمی پیچیده تر است و نیاز به تمرکز بیشتری دارد. کلا وظیفه این تابع دریافت اعداد دورقمی مربوط به ساعت، ماه و سال ... و ست کردن آنها در DS1307 می باشد. متغیر digit یک آرایه دو عضوی است که عدد دو رقمی ورودی در آن قرار می‌گیرد. newtime آرایه ۶ عضوی برای نگهداری اطلاعات ساعت، دقیقه، ثانیه، روز، ماه و سال می باشد. locat نیز مکان های نوشتن اطلاعات ورودی روی LCD می باشد.

ابتدا check_pass فراخوانی می شود تا هرکسی که رمز عبور را دارد بتواند ساعت تنظیم کند، بنابراین اگر رمز عبور صحیح باشد op=50 و مجوز تغییرات صادر می‌شود، ابتدا کاراکتر هایی روی نمایشگر چاپ می شود سپس در حلقه داخلی ۶ بار اطلاعات دو رقمی دریافت کرده و در نهایت سوال ذخیره سازی مطرح می‌شود و با تائید آن اطلاعات جدید جایگزین خواهد شد. اما در حلقه for چه اتفاقاتی رخ میدهد؟ ابتدا هر دو عضو digit صفر می‌شود، و بعد از آن شرطی برای x>2 گذاشته شده است که در صورت برقرار شدن این شرط مکان به سطر دوم انتقال می یابد. حال وارد حلقه for داخلی می‌شویم و مکان نما به سطری که y دارد و مقداری که x دارد از آرایه locate[x] محل مکان نما در مختصات x انتخاب شده و به آنجا منتقل می‌شود، بعد از آن کیبورد فراخوانی می شود و مقدار کلید فشار داده شده در آرایه digit قرار می‌گیرد حال اگر j=0 بود اولین رقم در ۱۰ ضرب شده و در عضو اولی آریه قرار می‌گیرد و در صورتی که j=1 بود عمل ضرب انجام نمی‌گیرد و مستقیما در عضو دومی آرایه قرار خواهد گرفت بنابراین به این صورت ما یک عدد صحیح دور رقمی دریافت کردیم، حال اگر time out از سوی کیبورد نداشته باشیم عدد دریافت شده روی نمایشگر چاپ می‌شود. اما اگر time out داشته باشیم قسمت دوم این شرط یعنی else goto endTime اجرا شده و تابع به کار خود پایان می‌دهد بنابراین هیچ عمل بروزرسانی در DS1307 انجام نخواهد گرفت. با فرض اینکه اطلاعات صحیح وارد شده و time out نیز نداریم در ادامه برنامه سوال ذخیره سازی مطرح شده و با تائید آن مقادیر جدید در حافظه DS1307 RAM جایگزین شده و به اصطلاح آپدیت می‌شود. نکته آخر اینکه اطلاعات جدید سال، ماه و روز را وارد تابع gregorian_week_day کرده تا روز هفته محاسبه شود و متغیر week هم در حافظه RAM آپدیت شود.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳-۲-۱۳ تابع اصلی برنامه

```
void main(void){
  unsigned char a,b;
  lcd=1;
  lcd_init(16);

  PORTC=0x00;
  DDRC.2=1;
  DDRC.3=1;
  PORTD=0x00;
  DDRD.7=1;

  err_cunt=err_cun_num;
  new_pass();

  i2c_init();
  rtc_init(0,0,0);

  ACSR=0x80;
  SFIOR=0x00;
  ADMUX=ADC_VREF_TYPE & 0xff;
  ADCSRA=0x87;
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

تابع main که تمامی توابع نوشته شده در اینجا فراخوانی و کنترل می‌شوند، در ابتدای این تابع تا قبل از شروع حلقه بینهایت پیکره بندی ها و تنظیمات اولیه صورت می‌گیرد. که ابتدا LED نمایشگر روشن می‌شود و بعد از آن نمایشگر به صورت ستون های ۱۶ تایی پیکره بندی می‌شود. در ادامه پین هایی از پورت ها به صورت خروجی تعریف می‌شود و بعد از آن نیز شمارنده فرصت های باقی مانده برای وارد کردن رمز با مقدار پیش فرض err_cun_num بارگذاری شده و تابع new_pass برای چک کردن اینکه آیا رمز اولیه ساخته شده یا نه فراخوانی می‌شود. در قسمت بعد پروتکل I2C و خود آیسی DS1307 جهت راه اندازی پیکره بندی اولیه می‌شوند. و در آخر هم مبدل آنالوگ به دیجیتال با تنظیمات اولیه آماده راه اندازی و نمونه برداری می‌شود.

از اینجا به بعد CPU وارد حلقه بینهایت شده و توابع را به صورت مداوم فراخوانی می‌کند.

```
    while (1){
     lcd_back_light();
     get_rtc();
     key=0xFF;op=0; //restr this var
         if(err_cunt==0){ //wait for error
            lcd_clear();
            lcd_gotoxy(3,0);
            lcd_putsf("ERROR ...");
            lcd_gotoxy(0,1);
            lcd_putsf("You Don't Input");
            alarm=1; delay_ms(500); alarm=0; delay_ms(300);
            alarm=1; delay_ms(500); alarm=0; delay_ms(300);
            alarm=1; delay_ms(2000); alarm=0;

            for (a=0; a<15; a++){
               for (b=0; b<60; b++){ // 1 min
                 lcd_gotoxy(15,0);
                 lcd_putsf("|");
                 delay_ms(300);
                 lcd_gotoxy(15,0);
                 lcd_putsf("/");
                 delay_ms(300);
                 lcd_gotoxy(15,0);
                 lcd_putsf("-");
                 delay_ms(300);
               }
            }
          err_cunt=err_cun_num;
         }
     display();
     key = keypad(0);

      if(key==1) change_time();
      if(key==10) change_pass();
      if(key==11){
        get_pass();
        if(key!=100) check_pass(1); //if: no time out then open=1
      }
    }
 }
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

در حلقه بینهایت تابع lcd_back_light برای کنترل نور نمایشگر به صورت مداوم فراخوانی می‌شود. و بعد هم تابع get_rtc برای بروز رسانی کردن متغیر های ساعت و تاریخ فراخوانی می‌شود. متغیر های key و op را ریست میکنیم، متغیر مهم ما یعنی err_cunt ابتدا چک می شود و در صورتی که برابر با صفر شده بود سیستم به مدت ۱۵ دقیقه قفل شده و هیچ عملی انجام نخواهد داد، این قسمت برای امنیت بیشتر قرار داده شده تا که اگر کسی رمز عبور را نمی‌داند و بخواهد با امتحان کردن تعدادی رمز عبور تصادفی به رمز عبور اصلی دست یابد، نتواند بیشتر از ۴ مورد را امتحان کند و پس از ۴ مرتبه سیستم با ریتم خاصی بوق آیفون شما را به صدا در خواهد آورد و قفل خواهد کرد که البته بعد از سپری شدن این زمان err_cunt دوباره با مقدار پیش فرض بارگیری خواهد شد.

بعد از این شرط تابع display فراخوانی شده تا نمایگشر را تازه کند. بعد از آن تابع keypad(0) فراخوانی می‌شود که در اینجا چون صفر ورودی تابع است تنها به مدت یک ثانیه کیبورد اسکن شده و بر می‌گردد حالا مقدار key را مورد مقایسه قرار می‌دهد و هر کدام یک از شرط ها که برقرار شود توابع خاص و مربوط به آنها فراخوانی می‌شوند.

۳-۳ توابع موجود در فایل هدر

۳-۳-۱ متغیرهای سراسری این هدر

```
typedef unsigned char byte;
 unsigned char g_days_in_month[] = {31, 28, 31, 30, 31, 30, 31, 31, 30,
31, 30, 31};
 unsigned char j_days_in_month[] = {31, 31, 31, 31, 31, 31, 30, 30, 30,
30, 30, 29};
 char buffer[32],week;
 unsigned char hour=0,minute=0,second=0;
 unsigned char year=0,month=0,day=0;
 long int y,m,d;
 long int jy,jm,jd;
 long int gy,gm,gd;
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

byte همان unsigned char می باشد که توسط دستور typedef تغییر نام داده شده است. g_days_in_month و j_days_in_month آرایه هایی هستند که تعداد روزهای ماه های شمسی و میلادی در آن قرار گرفته اند. سایر متغیر ها هم مربوط به ساعت و تاریخ می باشند، آنهایی که با حرف j شروع می شوند شمسی اند و آنهایی که با حرف g شروع می شوند میلادی اند. برخی از متغیر های سراسری که می خواهیم به صورت مشترک در اینجا هم از آنها استفاده نمائیم باید مجددا به صورت سراسری تعریف شان کنیم.

۳-۳-۲ تابع تقسیم ساده

```
   long int div(long int a, long int b){
    return (int) (a / b);
   }
```

این تابع دو عدد a و b را گرفته و آنها را برهم تقسیم می‌کند، سپس نتیجه را برگشت می‌دهد.

۳-۳-۳ تابع تبدیل تاریخ میلادی به شمسی

```
void gregorian_to_jalali(long int g_y, long int g_m, long int g_d){
  long int g_day_no,j_day_no,j_np;
  unsigned int i;
 gy = g_y-1600;
 gm = g_m-1;
 gd = g_d-1;
 g_day_no = 365*gy+div(gy+3,4)-div(gy+99,100)+div(gy+399,400);

    for (i=0; i < gm; i++)
       g_day_no += g_days_in_month[i];
    if (gm>1 && ((gy%4==0 && gy%100!=0) || (gy%400==0)))
       /* leap and after Feb */
       g_day_no++;
    g_day_no += gd;
     j_day_no = g_day_no-79;
     j_np = div(j_day_no, 12053); /* 12053 = 365*33 + 32/4 */
     j_day_no = j_day_no % 12053;
     jy = 979+33*j_np+4*div(j_day_no,1461); /* 1461 = 365*4 + 4/4 */
     j_day_no %= 1461;

     if (j_day_no >= 366) {
       jy += div(j_day_no-1, 365);
       j_day_no = (j_day_no-1)%365;
     }

     for (i = 0; i < 11 && j_day_no >= j_days_in_month[i]; i++)
     j_day_no -= j_days_in_month[i];
     jm = i+1;
     jd = j_day_no+1;
}
```

این تابع دارای سه پارامتر ورودی شامل سال، ماه و روز است که برحسب میلادی می‌باشد. با توجه به قوانین تبدیل سال میلادی به سال شمسی این تابع یک سری ضرب و تقسیم ها را انجام می دهد تا این عمل تبدیل صورت گیرد. تابع تبدیل خیلی دقیق می باشد حتی سالهای کبیسه را نیز محاسبه می‌کند. این تابع را خود بنده نه نوشته ام، بلکه از سورس سیستم مدیریت محتوای نیوک Nuke که به زبان PHP نوشته شده بود استفاده کردم و آن را به زبان سی C میکروکنترلر تبدیل کردم. کلا از لحاظ دستوری چیز خاصی ندارد، همانطور که مشاهده می‌کنید از ضرب و تقسیم استفاده شده، تنها سوالی که پیش می آید این است که این اعداد ثابت از کجا آمده اند؟ همانطور که اشاره شده بر طبق قوانین ریاضی تبدیل سالها به یکدیگر است.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳-۳-۴ تابع بدست آوردن روز هفته

```
void gregorian_week_day(long int g_y, long int g_m, long int g_d){
 long int gy,gm,gd,g_day_no;
 unsigned int i;
     gy = g_y-1600;
     gm = g_m-1;
     gd = g_d-1;
     g_day_no = 365*gy+div(gy+3,4)-div(gy+99,100)+div(gy+399,400);
     for (i=0; i < gm; i++)
         g_day_no += g_days_in_month[i];
     if (gm>1 && ((gy%4==0 && gy%100!=0) || (gy%400==0)))
     /* leap and after Feb */
       g_day_no++;
     g_day_no += gd;
     week = (g_day_no + 5) % 7 + 1;
}
```

این تابع نیز از سورس همان سیستم نیوک گرفته شده است، اصول کار آن به این صورت است که سال و ماه و روز میلادی را دریافت کرده و پس از انجام یک سری عملیات ریاضی روز هفته را مشخص می کند. مثلا شما یک تاریخی مربوط به ۱۰۰ سال پیش یا آینده را به این تابع بدهید به شما خواهد گفت که آن روز چند شنبه است. (به نظر من کار این تابع خیلی جالب است)

۳-۳-۵ تابع تعریف کاراکتر خاص

```
void define_char(byte flash *pc,byte char_code){
 byte i,a;
 a=(char_code<<3) | 0x40;
 for (i=0; i<8; i++) lcd_write_byte(a++,*pc++);
 }
```

این تابع که یکی از توابع معروف Help نرم افزار کدویژن است، برای شما کاراکترهای خاص را که روی حافظه فلش به صورت آرایه ۸ عضوی تعریف کرده اید را به کمک دستور lcd_write_byte در LCD برای شما به نقش می‌کشد. شما می توانید با نرم افزار هایی که برای اینکار طراحی شده اند اشکال خاص خود را به صورت پیکسلی رسم کرده و سپس کدهای ۸ بایتی آن را در حافظه فلش میکرو تعریف کرده و با استفاده از توابع زیر آنها را روی LCD رسم کنید. که در اینجا ما کلمه فارسی "شنبه" و "جمعه" را تعریف کرده ایم.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۳-۳-۶ تابع کاراکتر فرصت های باقی مانده

```
void cunter_err_pass(unsigned char err_cunt){
  if(err_cunt==4){
     define_char(char_4err,6);
     lcd_gotoxy(15,0);
     lcd_putchar(6);
  }

  if(err_cunt==3){
     define_char(char_3err,6);
     lcd_gotoxy(15,0);
     lcd_putchar(6);
  }

  if(err_cunt==2){
     define_char(char_2err,6);
     lcd_gotoxy(15,0);
     lcd_putchar(6);
  }

  if(err_cunt==1){
     define_char(char_1err,6);
     lcd_gotoxy(15,0);
     lcd_putchar(6);
  }
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

این تابع هم به این صورت عمل می کند که ابتدا متغیر err_cunt را مورد مقایسه قرار می دهد و با توجه به مقدار آن یکی از شرط ها برقرار شده و آن کاراکتر خاص را در مکان مربوطه رسم می‌کند. ساده تر بگویم چوب خط های باقی مانده کاربر را می‌کشد.

۳-۳-۷ تابع نمایش فارسی روز هفته

```
void week_name_FA(){
   if(week!=5){
     define_char(char_sh,0);
     define_char(char_nb,1);
     define_char(char_h,2);
     lcd_gotoxy(12,1);
     lcd_putchar(2);
     lcd_putchar(1);
     lcd_putchar(0);
     lcd_gotoxy(15,1);
   }else{
         define_char(char_j,4);
         define_char(char_m,5);
         define_char(char_h,2);
         lcd_gotoxy(12,1);
         lcd_putchar(2);
         lcd_putchar(5);
         lcd_putchar(4);
   }
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

در اینجا نیز عدد روز هفته که در week قرار دارد را تست می کند که آیا امروز جمعه است یا روزهای دیگر. همانطور که می‌دانید همه روزهای هفته ما با کلمه شنبه شروع می‌شود به جزء روز جمعه، پس این تابع روز را شناسایی کرده و یکی از کلمات شنبه یا جمعه را به صورت فارسی در مکان های مشخص شده نمایشگر رسم می‌کند.

۳-۳-۸ تابع انتخاب روز هفته و نمایش آن

```
void week_name(){
  week_name_FA();
   switch(week){
      case 1:
          lcd_putchar('2'); //"دوشنبه"
          break;
      case 2:
          lcd_putchar('3'); //"سه شنبه"
          break;
      case 3:
          lcd_putchar('4'); //"چهارشنبه"
          break;
      case 4:
          lcd_putchar('5'); //"پنج شنبه"
          break;
      case 5:
          //"جمعه"
          break;
      case 6:
          lcd_putchar(' '); //"شنبه"
          break;
      case 7:
          lcd_putchar('1'); //"یکشنبه"
          break;
   }
 }
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

بعد از نوشتن روز شنبه یا جمعه توسط تابع week_name_FA حال باید عدد روز قبل از کلمه شنبه نوشته شود که این کار هم توسط این تابع انجام می‌گیرد. که البته برای روز شنبه کاراکتر خالی ارسال می‌کند و برای بقیه روزها عدد آن روز را چاپ می‌کند.

۳-۳-۹ تابع دریافت تاریخ و ساعت

```
void get_rtc(){
   rtc_get_time(&hour,&minute,&second);
   rtc_get_date(&week,&day,&month,&year);
   y=year+2000; m=month; d=day;
   gregorian_to_jalali(y,m,d);
   gregorian_week_day(y,m,d);
   //در صورتی که باتری بک آپ وجود نداشته باشد یا تغذیه ساعت قطع شود، مقدار دهی اولیه خواهد شد
   if(jy<1390){ //if not battery backup ==> set&reset RTC
     rtc_set_time(12,0,0); /* set time 12:00:00 */
     rtc_set_date(2,29,11,11); /* set date Tuesday 29/11/2011 */
   }
}
```

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

وظیفه این تابع نیز دریافت اطلاعات از DS1307 می باشد، که ابتدا اطلاعات را به صورت میلادی دریافت کرده سپس با افزودن عدد ۲۰۰۰ به متغیر سال آنها را برای تبدیل به شمسی ارسال می نماید و بعد هم روز هفته را محاسبه می‌کند. و در آخر هم شرطی برای سال شمسی گذاشته ایم که در صورتی که از ۱۳۹۰ کوچکتر باشد این احتمال دارد که آیسی هنوز تنظیم نشده است و یا تغذیه آن و باتری پشتیبان آن خراب شده است بنابراین یک ساعت و تاریخ پیش فرض در آن ست می شود تا شروع به کار کند.

نکته: در صورتی که برای اولین بار DS1307 را به مدار وصل می‌کنید برای راه اندازی و شروع بکار آن ابتدا باید حتما، تاریخ و ساعت را تنظیم کنیم و در آن ست کنیم تا شروع بکار کند. سال پیش فرض اغلب آیسی ها به شمسی معادل ۱۳۷۸ می‌باشد.

sbargh.ir
@sbargh
sbargh.ir@yahoo.com

۴ مراجع

Help CodeVisionAVR V2.05.3

PHP NUKE Farsi

sbargh.ir
@sbargh
sbargh.ir@yahoo.com