

به نام خدا

راهنمای استفاده از کتابخانه ماژول GSM (اتمل استودیو-ایکسمگا)

تعاریف اولیه:

- 1- تعاریف پایه های متصل شده به ماژول را تعیین کنید. در قسمت : EDIT THIS DEFINE's
- 2- واحد سریال متصل شده به ماژول را تعیین کنید. در قسمت : EDIT THIS DEFINE's
- 3- دو تایمر در کتابخانه نیاز است. یک تایمر برای TIMEOUT واحد سریال و یک تایمر برای ساخت زمان ۲ ثانیه استفاده شده. توجه داشته باشید که تنظیمات تایمرها با فرض کلاک ۳۲ مگاهرتز میباشند. این تایمرها در قسمت EDIT THIS DEFINE's تعیین کنید.
- 4- در صورتی که از LCD کارکتری استفاده کرده اید؛ تعاریف LCD را فعال کنید. در این صورت اگر پیامک دریافت شود شماره موبایل ارسال کننده روی LCD نمایش داده میشود. در هنگام ارسال پیامک هم شماره ای که به آن پیام ارسال میشود نمایش داده میشود.
- 5- تعاریف زیر DONT EDIT THIS DEFINE's را تغییر ندهید.

متغیرها:

- متغیرها به صورت استراکچر(ساختار) تعریف شده اند. تحت نام gsm.
- متن دریافت شده در متغیر آرایه ای tex_recive ذخیره میشود.
- شماره ای که از آن پیامک دریافت شده در متغیر آرایه ای mobile_number ذخیره میشود.
- نام اپراتور در متغیر آرایه ای name_operator ذخیره میشود. Irancell یا IR-MCI یا NO KNOW ویا در صورت نبودن سیم کارت: NO SIM

توابع:

- تابع gsm_handl(): تابع اصلی که باید در MAIN برنامه قرار بگیرد، پیام های دریافت شده اتومات خوانده میشوند. در صورت دریافت پیامک مقدار ۱ (apply_gsm) برگردانده میشود.

مثال کاربردی:

```
handl = gsm_handl();
if(handl==apply_gsm)
{
    handl=DISABLE;
    application(gsm.tex_recive);
}
Else
{
    gsm.dell=ENABLE;
```

```

}

void application(char *recv_buffer)
{
    if (strstr_P(recv_buffer,PSTR("ON"))!=0)
    {
        PORTA.OUT = 1;
    }
}

```

توضیح: در صورتی که تابع (`gsm_handl()`) مقدار یک¹ (`apply_gsm`) برگرداند متن پیامک دریافت شده به تابع `application` ارسال میشود، این تابع در برنامه اصلی قرار دارد و یک تابع دلخواه است. داخل این تابع میتوان متن دریافت شده را تحلیل کرد.

۱- `apply_gsm` ابتدای کتابخانه با مقدار یک `define` شده است.

پاک کردن پیام دریافت شده :

اگر به پیامک دریافت شده جواب داده شود، بعد از ارسال جواب؛ پیامک دریافت شده اتومات پاک خواهد شد. در غیر این صورت (به پیام دریافت شده جواب داده نشود) میتوان با یک `dell` متغیر در `gsm` آخرین پیامک خوانده شده را با یک بار اجرای تابع (`gsm_handl()`) پاک کنید.

• تابع (`gsm_send_sms(char *mobile,char *tex)` :

تابع ارسال پیامک میباشد. ورودی آن شماره موبایل مقصد و متن پیامک ارسالی میباشد. برای پاسخ دادن به پیامک دریافت شده به جای شماره موبایل، متغیر `gsm. mobile_number` را قرار دهید. متن پیام را میتوان در متغیر `gsm. tex_send` آماده کنید و به عنوان ورودی تابع در نظر بگیرید.

• تابع (`send2gsm(char select)` :

برای ارسال یک سری دستورات و تعیین مدهای ماژول استفاده میشود. لیست کارهایی که انجام میدهد(تعاریف زیر به جای `select` مینشیند) :

<code>off_gsm</code>		خاموش کردن ماژول
<code>get_imei</code>		دریافت سریال ماژول
<code>gsm_mod</code>	GSM	قرار دادن در مد
<code>ucs2_mod</code>	UCS2	قرار دادن در مد
<code>del_all</code>		حذف تمام پیامک ها
<code>get_charj</code>		ارسال دستور دریافت شارژ
<code>sim_name</code>		ارسال دستور دریافت نام سیم کارت
<code>net_status</code>		ارسال دستور دریافت وضعیت شبکه
<code>gsm_rst</code>		ریست کردن ماژول
<code>dell_sms</code>		پاک کردن آخرین پیامک خوانده شده
<code>signal_q</code>		ارسال دستور دریافت قدرت آنتن شبکه

hang_up

قطع تماس

answer_ring

پاسخ دادن به تماس

• تابع `gsm_start(void)` :

این تابع برای راه اندازی اولیه ماژول میباشد. و یک بار قبل از حلقه اصلی برنامه فراخوانی میشود تا ماژول را روشن و کانفیگ کند. در این تابع تایمر ها، وقفه های لازم و پورت سریال متصل به ماژول و پورت سریال دبیگ (در صورت فعال کردن تعریف `DEBUGE` در ابتدای برنامه) فعال میشوند.

۱. در صورتی که سیم کارت قرار داده نشده باشد دستور خاموش شدن به ماژول ارسال میشود.
۲. بعد از کانفیگ شدن ماژول متغیر `gsm.stat` به معنای قرار داشتن و روشن بودن ماژول `ENABLE` یا یک میشود.
۳. میزان سیگنال در متغیر `signal_value` در استراچر `gsm` به صورت یک عدد ذخیره میشود.
۴. نام سیم کارت در متغیر `sim_operator` از استراچر `gsm` با یکی از تعاریف زیر ذخیره میشود:

```
#define irancell 1
#define ir_mci 2
#define no_know 4
#define no_sim 3
۵. همچنین نام سیم کارت در متغیر رشته ای name_operator از استراچر gsm به صورت یک رشته برای نمایش روی lcd ذخیره میشود.
۶. وضعیت شبکه در متغیر net_state از استراچر gsm با یکی از تعاریف زیر ذخیره میشود:
#define net_registred 10
#define net_not_registred 0
```

• تابع `gsm_dialing(char *tex)` :

برای شماره گیری شماره موبایل و یا کد `ussd` استفاده میشود. ورودی `tex` بعد از دستور `ATD` شماره گیری میشود.

• دریافت شارژ سیم کارت:

شارژ سیم کارت های ایرانسل و همراه اول قابل دریافت میباشد. میزان باقی مانده شارژ در متغیر `charge` از استراچر `gsm` به صورت رشته ذخیره میشود.

تابع `read_charj()` متن دریافت شده بعد از اجرای کد دریافت شارژ را تحلیل و میزان شارژ باقی مانده را از متن خارج میکند.

مطابق زیر عمل کنید:

```
send2gsm(gsm_mod);_delay_ms(200);
send2gsm(get_charj);
int timeout = 0;
while(gsm.flag!=gsm_read_charj && gsm.charj==DISABLE && timeout<20){_delay_ms(500);}
if(gsm.flag==gsm_read_charj)
{
    read_charj();
    sprintf(gsm.tex_send,"%s:مانده باقی شارژ",gsm.charge);
}
else
```

```
{strcpy(gsm.tex_send, "شارژ دریافت در خط");}
```

فلوچارت عملکرد برنامه:

