ارسال اطلاعات محیطی از طریق شبکه موبایل

استاد راهنما

جناب آقای دکتر باصری

استاد دفاع

جناب آقای دکتر کاشفی

پژوهنده

احسان صفامنش

تابستان 90

فهرست

دستورات AT Command

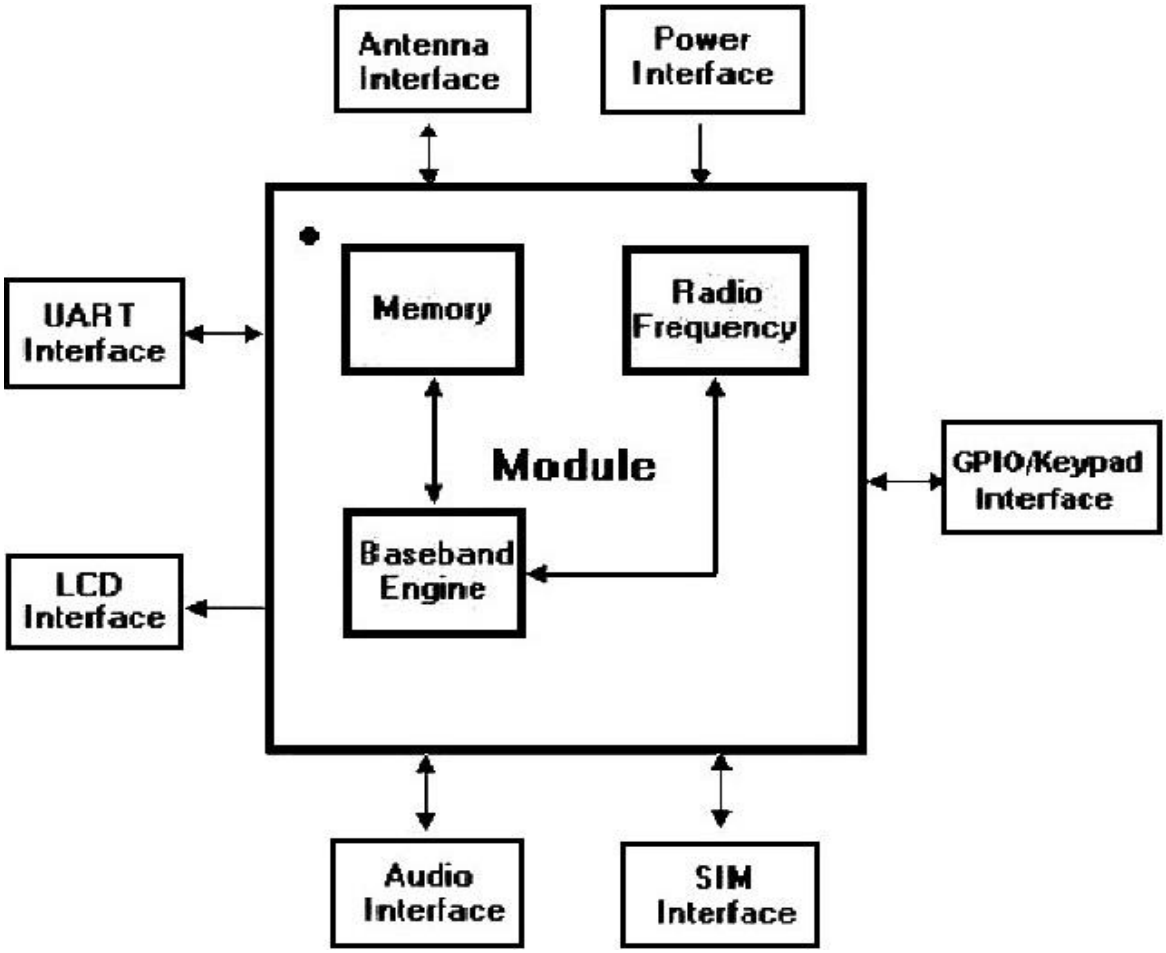
شرح پروژه

بررسی ماژول SIM900

شرکت simcom ماژول بسیار فشرده و قابل اطمینان sim900 را ارائه کرد. این یک ماژول GSM/GPRS کامل در نوع SMT و با یک هسته ی پردازشی بسیار قوی AMR926EJ-S طراحی شده است. و به شما اجازه می دهد تا از قیمت مناسب و ابعاد کوچک آن استفاده کنید.

انتقال صدا ، پیام کوتاه ، فکس و دیتا در یک پک کوچک با توان کصرفی کم از امکانات آن است.

بخش های اصلی SIM900 در شکل زیر نمایش داده شده است.



- **Memory** : شامل حافظه های RAM ، ROM و flash
- **بخش فرکانس رادیویی** : که در چهار باند GSM 850 / 900 / 1800 / 1900 عمل میکند و ارتباطات رادیویی بر عهده ی این بخش است.
- **پردازنده** : این بخش از یک هسته ی پردازشی ARM تشکیل شده و کنترل ماژول ، رابط ها و بخش رادیویی را بر عهده دارد.

-رابط ها ی ماژول

امکانات عمومی

چهاز باند 850 / 900 / 1800 / 1900

GPRS کلاس 10/8

ایستگاه مبایل GPRS کلاس B

مطابق با GSM phase 2/2+

Class 4 (2 W @850/ 900 MHz)
Class 1 (1 W @ 1800/1900MHz)

ابعاد : 24 در 24 در 3 میلی متر

وزن : 3.4 گرم

کنترل از طریق ATcommands (GSM 07.07,07.05
و SIMCOM enhanced ATCommands)

محدوده ی ولتاژ کاری : 3.1 تا 4.8 ولت

برنامه ی ابزار سیم کارت

امکانات توان مصرفی پایین : 1/5 میلی آمپر در حالت خواب

دمای کارکرد : 45- تا 85 درجه ی سانتیگراد

مشخصات برای پیام کوتاه

نقطه به نقطه ی MO و MT

سلول منتشر کننده ی پیامک

حالت text و PDU

مشخصات برای صدا

کد کننده های

Half rate (HR)
Full rate (FR)
Enhanced Full rate (EFR)
عملکرد هندس فری (Echo suppression)

AMR
Half rate (HR)
Full rate (FR)

مشخصات برای دیتا

GPRS کلاس 10 : دریافت تا 85.6 kbps

رابط ها (Interfaces)

رابط سیم کارت خارجی SIM 3V/ 1.8V

رابط صدای آنالوگ

RTC backup

رابط ارتباط سریال برای کنترل و دیباگ ماژول

رابط های SPI ، I2C

پایه های ورودی خروجی قابل برنامه ریزی

PWM و ADC

امکانات نرم افزاری

پروتکل MUX 0710

پروتکل TCP/UDP

FTP/HTTP

FOTA

MMS

پایه های SIM900

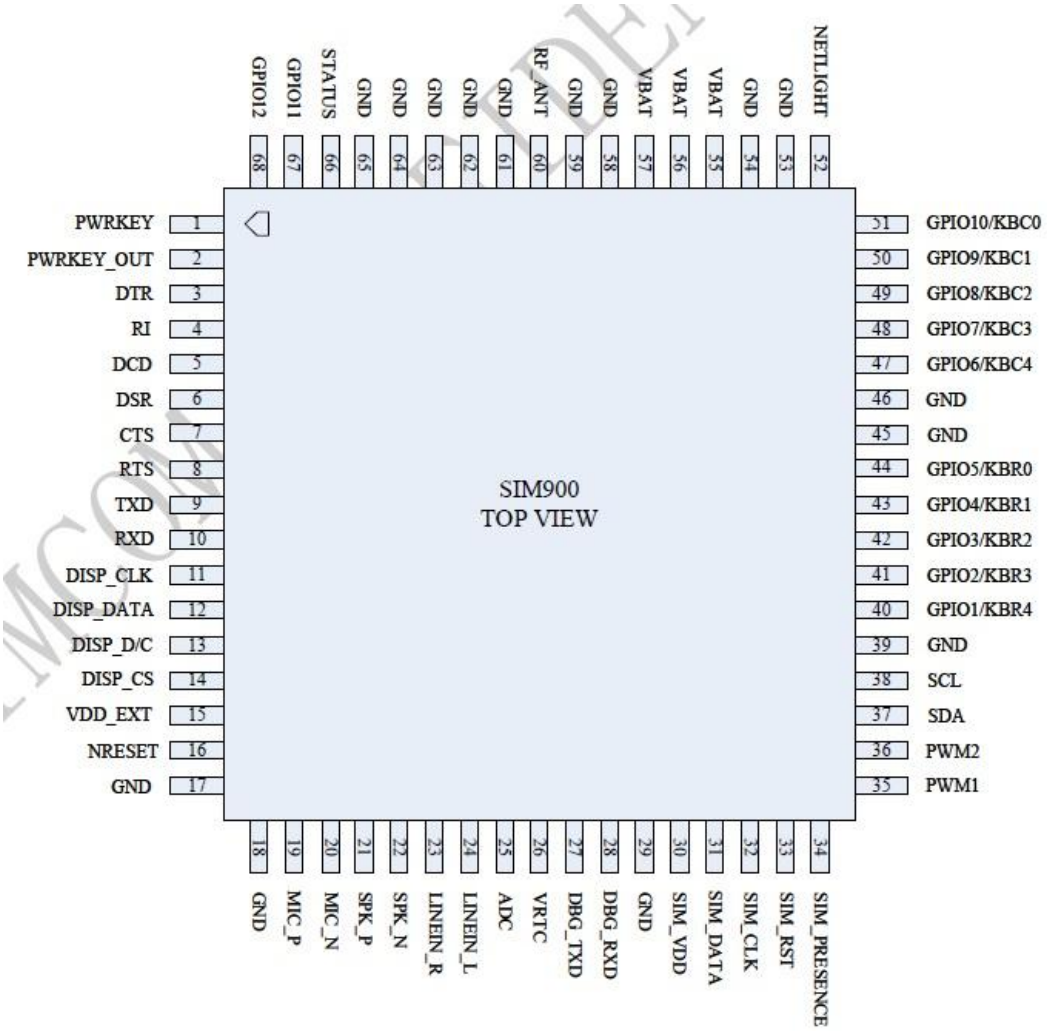
با توجه به شکل بعد که نمای بالایی پایه های SIM900 را نمایش می دهد:

با وصل کردن پایه های 1 و 2 به یکدیگر تا 1 ثانیه ماژول خاموش یا روشن می شود.

پایه های 3 تا 10 برای ارتباط سریال و تبادل داده و دستورات ATC به کار می رود.

پایه های 11 تا 14 برای اتصال LCD به ماژول استفاده می شود.

پایه ی 15 ولتاژ خروجی سطح منطقی ماژول تا 10 میلی آمپر را سورس می کند. ولتاژ سطح منتطقی 1 برابر 2.8 تا 3 ولت و منطق 0 ، صفر ولت است.



پایه ی 26 برای اتصال باطری بکاپ ماژول استفاده می شود

پایه های 27 و 28 ارتباط سریال برای دیباگ کردن و بروز کردن ماژول استفاده می شود.

پایه های 30 تا 34 رابط اتصال به سیم کارت هستند.

پایه های 40 تا 51 و 67 و 68 ، ورودی خروجی های قابل برنامه ریزی هستند که قابلیت اتصال صفحه کلید ماتریسی را دارند.

پایه ی 52 طبق جدول زیر وضعیت سیستم را نمایش می دهد.

NETLIGHT

State	SIM900 function
Off	SIM900 is not running
64ms On/ 800ms Off	SIM900 does not find the network
64ms On/ 3000ms Off	SIM900 find the network
64ms On/ 300ms Off	GPRS communication

اگر پایه خاموش باشد (صفر): SIM900 فعال نیست.

64 م ث روشن و 800 م ث خاموش : شبکه یافت نشده است.

64 م ث روشن و 3 ثانیه خاموش : شبکه پیدا شده است.

64 م ث روشن و 300 م ث خاموش : در حال ارتباط GPRS

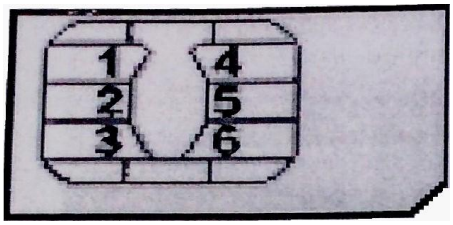
پایه ی 60 برای اتصال به آنتن

پایه های 55 تا 57 ولتاژ تغذیه ی ماژول است که در محدوده ای بین 3.2 تا 4.8 ولت کار می کند.

پایه ی 66 وضعیت ماژول را با تغییرات سطح منطقی معین می کند.

نحوه ی اتصال سیم کارت با ماژول و مشخصات سیم کارت در زیر نشان داده شده است:

مشخصات پایه های سیم کارت



1 – Vcc : پایه ی تغذیه ی سیم کارت

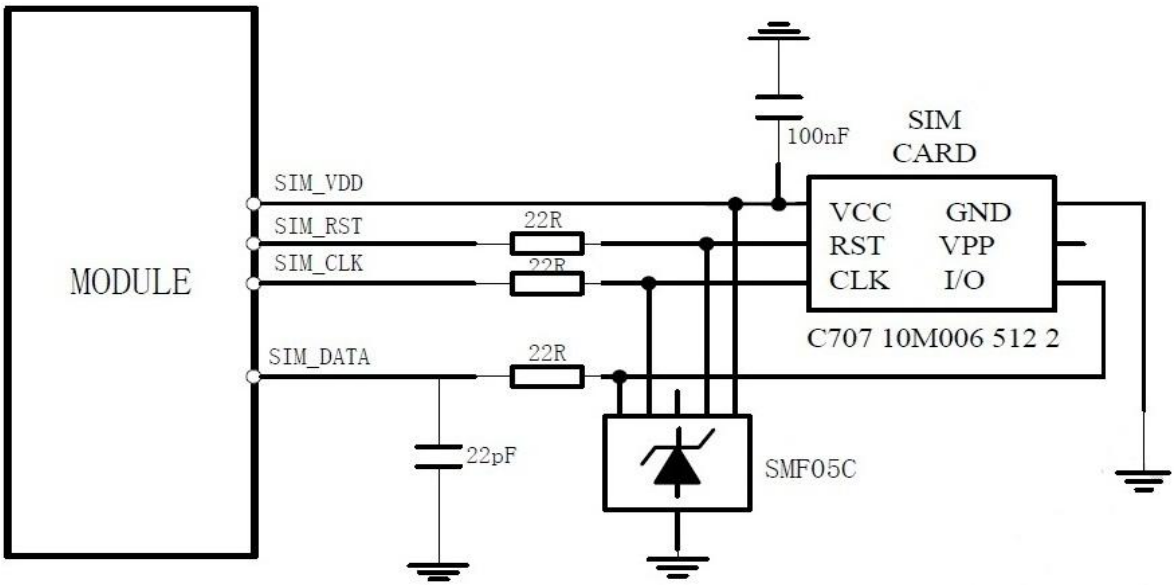
2 – Rst : پایه ی ریست کردن سیم کارت

3 – Clk : پایه ی کلاک سیم کارت

4 – GND

5 – Vpp : برای برنامه ریزی سیم کارت (دلخواه)

6 – Data : پایه ی دیتای سیم کارت



دستورات AT Command

دستورات عمومی

AT COMMAND	عملکرد
AT	چک کردن دستورات
ATI	شناسایی ورژن firmware
AT+GMR	شناسایی ورژن کامل firmware
AT+GMM	شناسایی مدل ماژول
AT+CSCA	چک کردن شماره ی مرکز SMS
AT+CREG	مشخص شدن اتصال به شبکه
AT+CSQ	قدرت سیگنال
AT&F	تنظیمات کارخانه
AT&W	ذخیره ی تنظیمات
AT&V	چک کردن تنظیمات ماژول
AT+CPIN=“XXXX”	باز کردن شماره ی PIN
AT+IPR=x	تنظیم baud rate (مثال: x=9600)
AT+CUSD=1,”*140*1#”,15	درخواست اعلام اعتبار از ایرانسل

تنظیم زمان

AT COMMAND	عملکرد
AT+CCLK?	خواندن زمان ماژول
AT+CCLK= “YY/MM/DD,HH:MM:SS+02”	تنظیم زمان

ذخیره ی شماره ی تلفن

AT COMMAND	عملکرد
AT+CPBS="ON"	آماده سازی برای ذخیره ی شماره در سیم کارت
AT+CPBW=1,“+98*”,145,“NAME”	ذخیره ی شماره در خانه ی اول سیم کارت 145 شماره گیری بین الملل
AT+CNUM	اگر فعال باشد شماره نشان داده می شود
AT+CPBS="SM"	بازگشت به ذخایر سیم کارت
AT+CPBR=1,5	خواندن از مکان 1 تا 5

ایجاد تماس صوتی

AT COMMAND	عملکرد
AT+CHFA?	چک کردن تنظیمات کانال صوتی
AT+CHFA: 0	صفر هندست ، یک aux
ATL3	تنظیم بلندی صدا 1 تا 9
ATD+2783xxxx;	شماره گیری
ATDL	شماره گیری مجدد
ATA	جواب به تماس دریافتی
ATH	قطع تماس

تنظیمات برای SMS

AT COMMAND	عملکرد
AT+CMGF=1	حالت متنی(اگر صفر باشد حالت Packet Data)
AT+CSCS="GSM"	حالت متنی کاراکتر های GSM
AT+CNMI=2,1,0,0	نشان دادن SMS جدید (پیشفرض فعال)
AT+CSCS?	چک کردن شماره ی مرکز sms
AT+CSAS=0	ذخیره ی تنظیمات SMS در پروفایل 0 یا 1
AT+CMGS = " +27.."(Enter) >your message < ctrl-z>	فرستادن SMS
AT+CMGW >your message < ctrl-z> +CMGW: 1	ذخیره ی SMS در SIM متن مورد نظر بازگرداندن مکان ذخیره شده(دراینجا 1)
AT+CMSS=1,"+27...",145	فرستادن SMS ذخیره شده
AT+CMGDA= "DEL READ" "DEL UNREAD" "DEL SENT" "DEL UNSENT" "DEL INBOX" "DEL ALL"	پاک کردن یک دسته از SMS های مشخص مثلا : AT+CMGDA="DEL ALL" همه را پاک می کند

تنظیمات ماژول برای GPRS

AT COMMAND	عملکرد
	Irancell APN name = internet
ATE1	فعال کردن echo در حالت تماس و SMS باید صفر باشد
AT+CMGF=1	حالت متنی (text mode)
AT+CSCS="GSM"	حالت متنی کاراکتر های GSM
AT+CGATT?	باید یک برگردد. در غیر این صورت AT+CGATT=1
AT+CIPMUX=0	غیر فعال کردن Multiplex mode 1 فعال می کند Server در این حالت کار نمی کند.
AT+CIPMODE=0	تنظیم به command mode

متصل شدن با GPRS به عنوان client

AT COMMAND	عملکرد
AT+CIPCSGP=1,"internet"	پیدا کردن اتصال APN
AT+CLPORT="TCP","2020"	پیدا کردن اتصال TCP و پورت برای وسیله
AT+CSTT="internet","",""	متصل شدن به MTN "APN","user name","pass"
AT+CIPSRIP=1	نمایش IP وسیله ی اتصال دهنده
AT+CIICR	متصل شدن
AT+CIFSR	گرفتن IP ماژول محلی (باید استفاده شود، یا error باز گردانده می شود)
AT+CIPSTART= "TCP","xxx.xxxx.xxx","xxxx"	تنظیم domain , IP , port صبر کنیم تا "CONNECT OK" باز گردانده شود.
AT+CIPSTATUS	باید باشد: "STATE: IP STATUS"

AT+CIPSEND	فرستادن داده (از ctrl+z استفاده کنیم) مثال: AT+CIPSEND Xxxxxxxxxx(ctrl-z) "SEND OK"
AT+CIPSHUT	قطع کردن اتصال GPRS

متصل شدن با GPRS به عنوان server

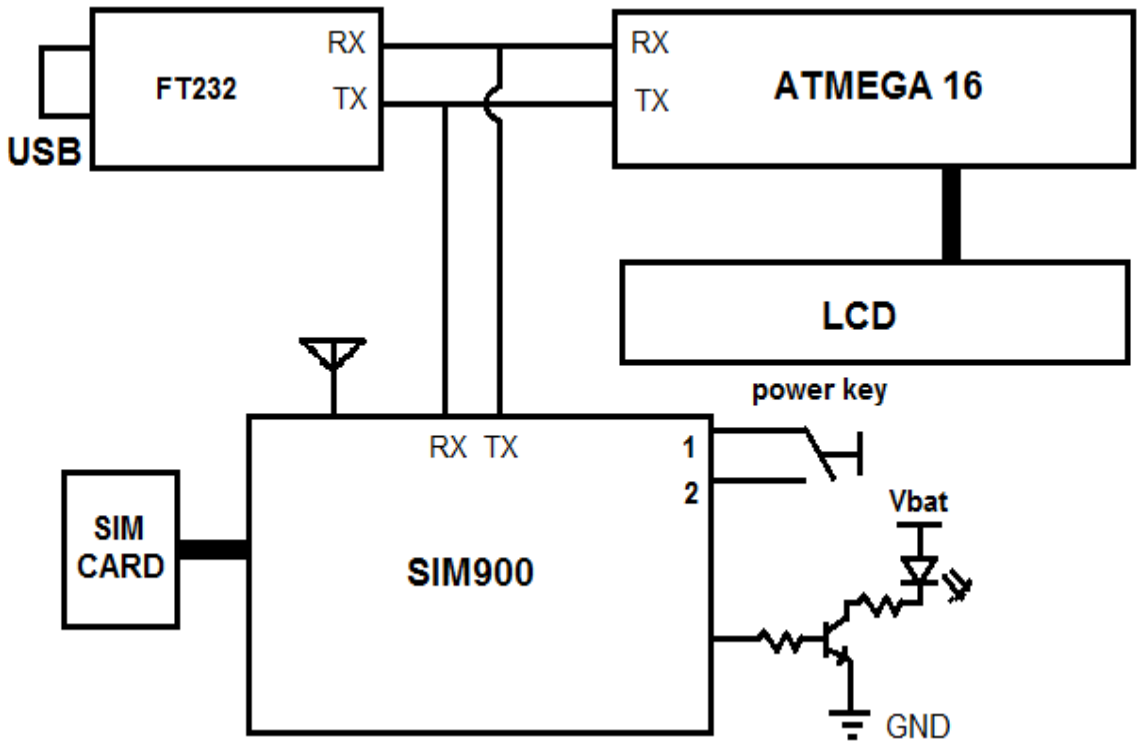
AT COMMAND	عملکرد
AT+CIPCSGP= 1,"internet","",""	پیدا کردن اتصال GPRS "APN","user name","pass"
AT+CIPSRIP=1	نمایش IP و port
AT+CIPSERVER=1,2020	پیدا کردن اتصال و پورت (0 = USD)
AT+CIPSTATUS	بررسی وضعیت اتصال
AT+CIPSEND	فرستادن داده (از ctrl+z استفاده کنیم) مثال: AT+CIPSEND Xxxxxxxxxx(ctrl-z) "SEND OK"
AT+CIPSHUT	قطع کردن اتصال GPRS

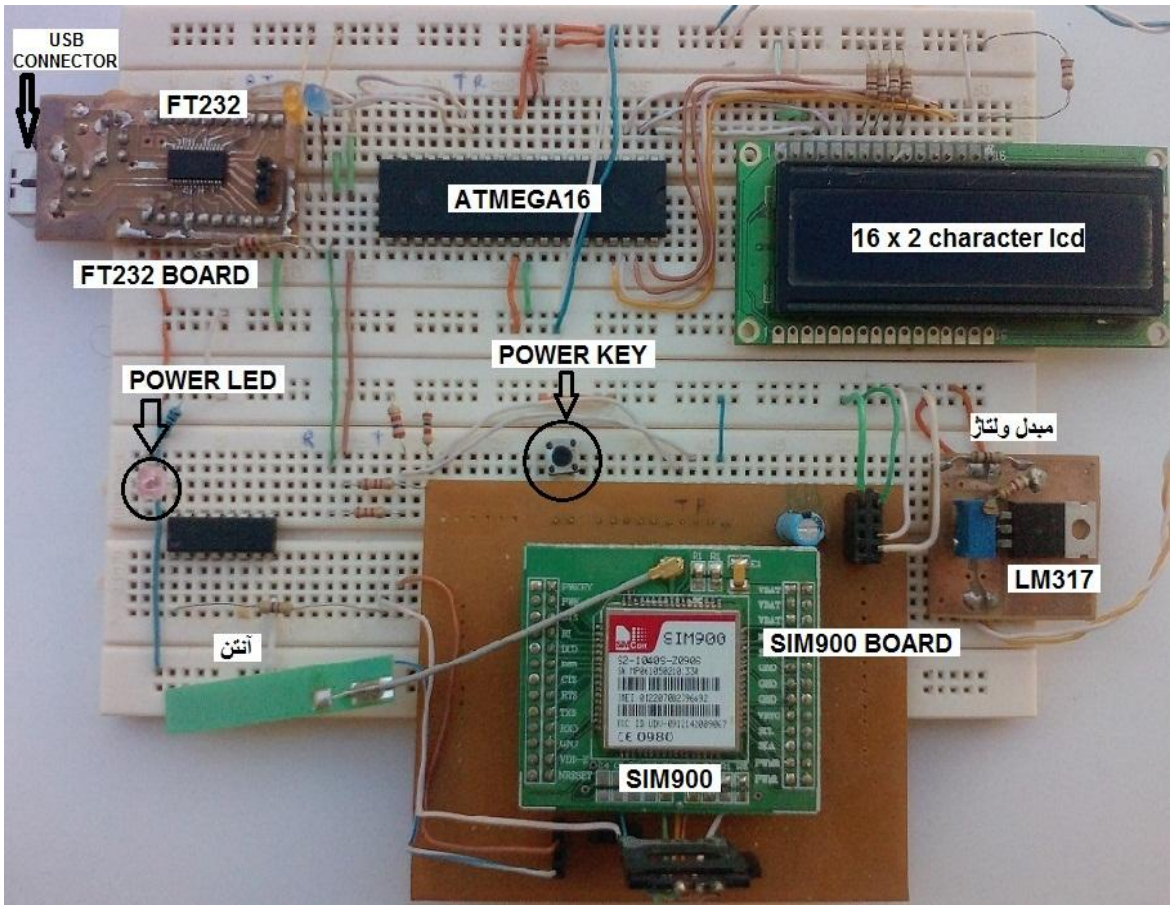
فصل چهارم : شرح پروژه

سخت افزار و عملکرد مدار

دراین قسمت با نحوه اتصالات آی سی ها و ارتباط آنها باهم آشنا می شویم. و روند انجام کار به صورت مرحله به مرحله توضیح داده می شود.

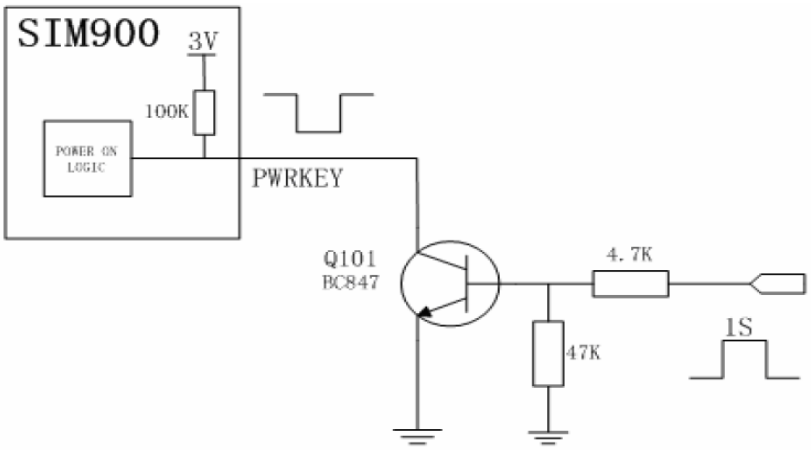
بخش های اصلی مدار در شکل زیر نشان داده شده است.



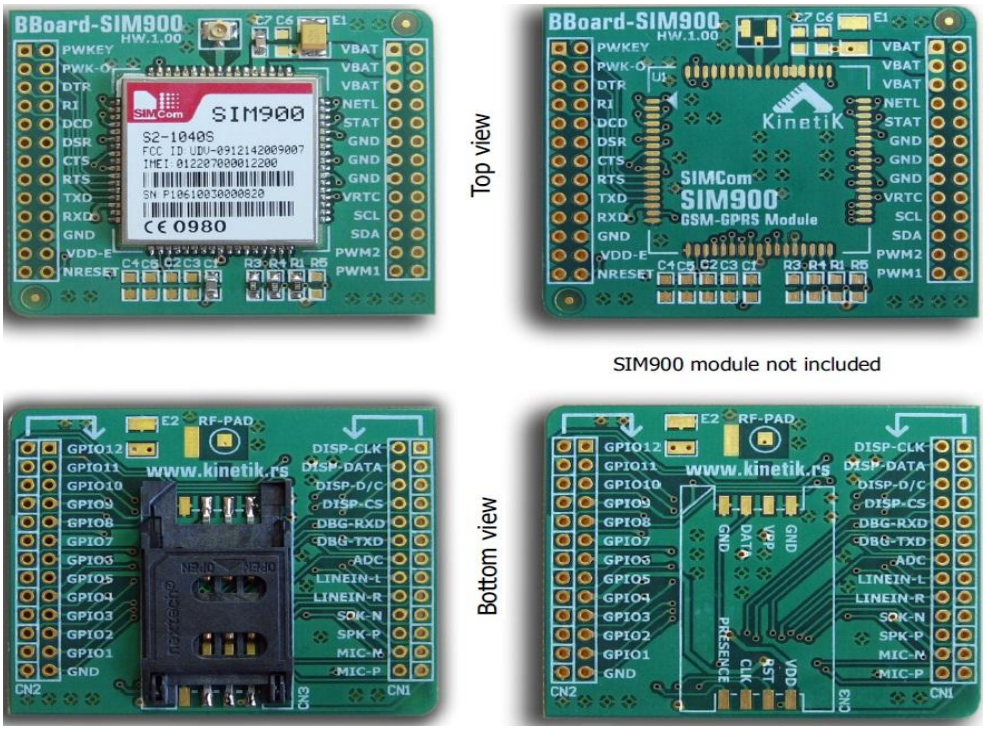


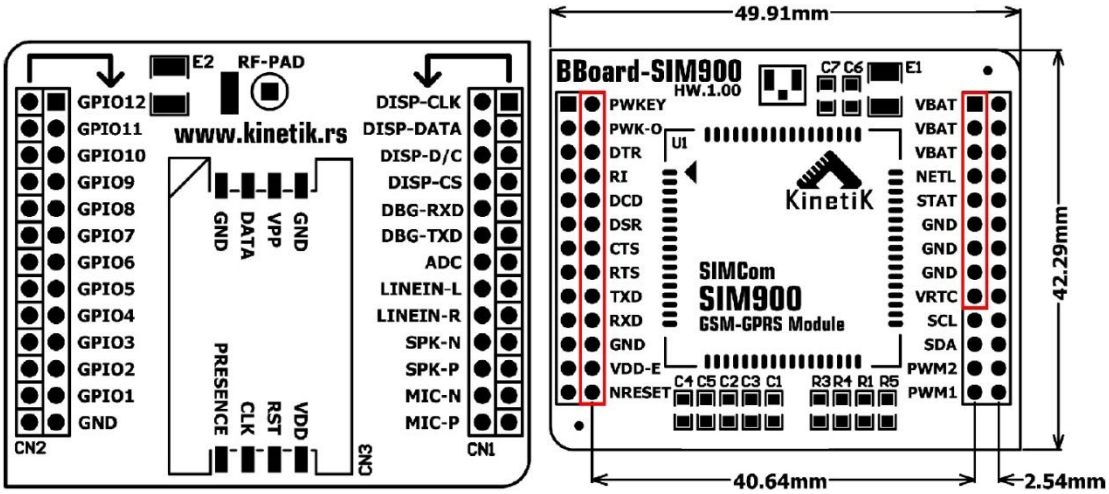
میکروکنترلر ATMEGA16 وظیفه ی پردازش اطلاعات و کنترل سیستم را بر عهده دارد. دریافت اطلاعات محیط مورد نظر مانند گلخانه و کنترل ابزار موجود نیز بر عهده این میکروکنترلر است.

SIM900 وظیفه ی ایجاد ارتباط بین مدار و شبکه ی GSM را بر عهده دارد. با فشردن کلید power key به مدت حدودا یک ثانیه ماژول SIM900 روشن یا خاموش می شود. با میکرو کنترلر نیز می توان SIM900 را خاموش یا روشن کرد که در شکل زیر ملاحظه می کنیم:



بر روی بورد زیر ماژول SIM900 ، SIM card و آنتن قرار گرفته اند.



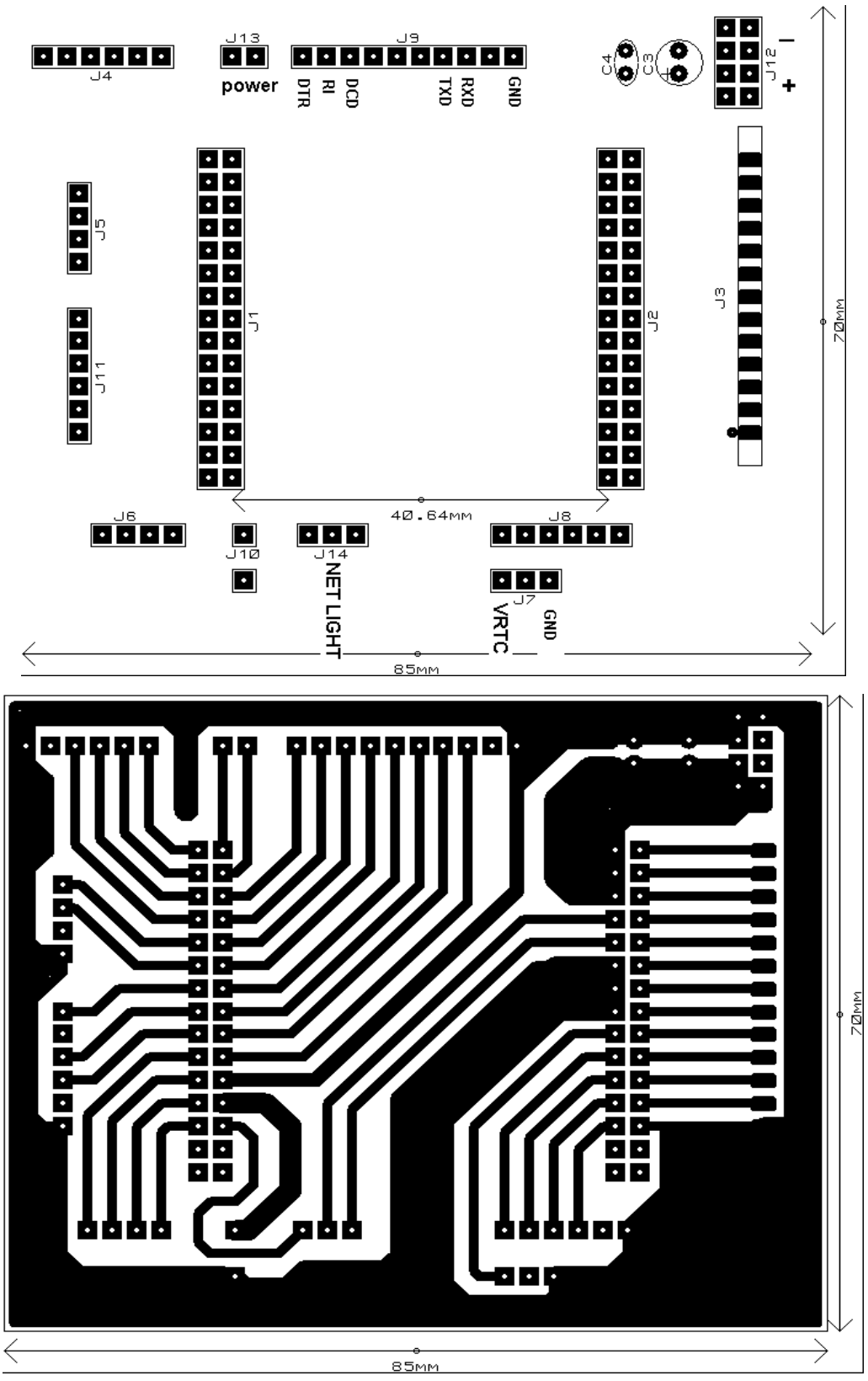


این برد قابلیت قرار گرفتن بر روی برد بورد را ندارد، بتابراین برد زیر طراحی شد تا بورد بالا (Bbord) بر روی آن قرار بگیرد و بتوان از امکانات SIM900 بر روی برد بورد استفاده نمود.

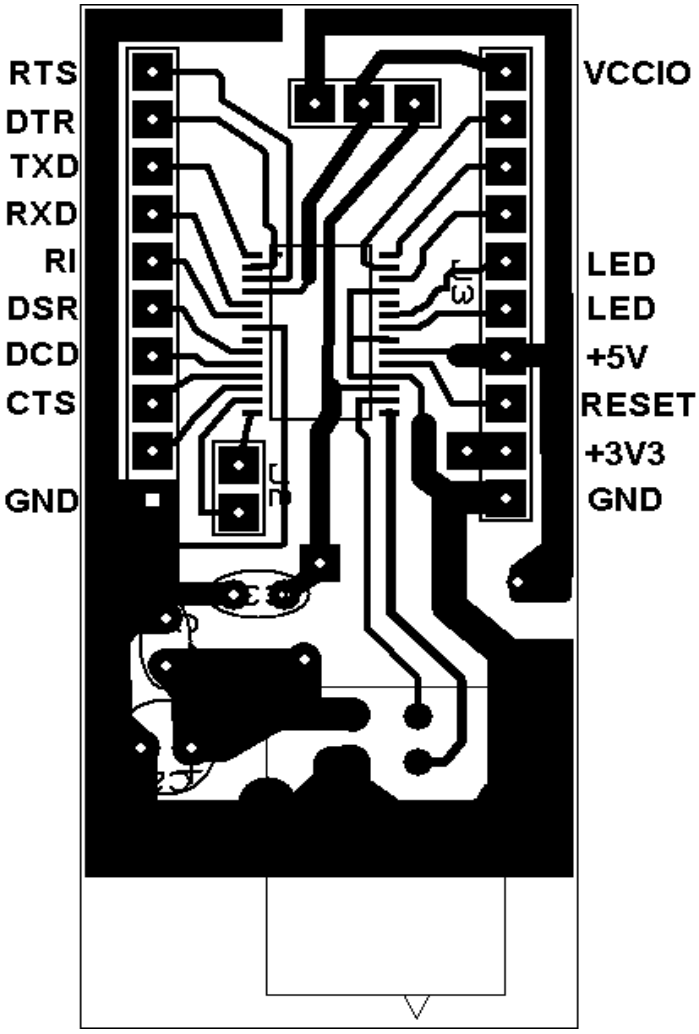
ابعاد این برد بر روی شکل نشان داده شده است.

SIM CARD برای شناسایی در شبکه به کار می رود که به همراه SIM900 ، بخش کوچکی از شبکه GSM را تشکیل می دهد.

توسط این دو بورد ،می توان ماژول SIM900 را در مدار های دیگر نیز راه اندازی و استفاده نمود.



وضعیت سیستم نیز به وسیله ی led ای که به پایه ی netlight متصل شده مشخص می شود که قبلا در بخش معرفی SIM900 توضیح داده شد. همچنین اگر از ماژول در مکان هایی استفاده می شود که قدرت سیگنال کم است، میتوان از آنتن های بزرگتر و بهتر به صورت خارجی استفاده نمود.



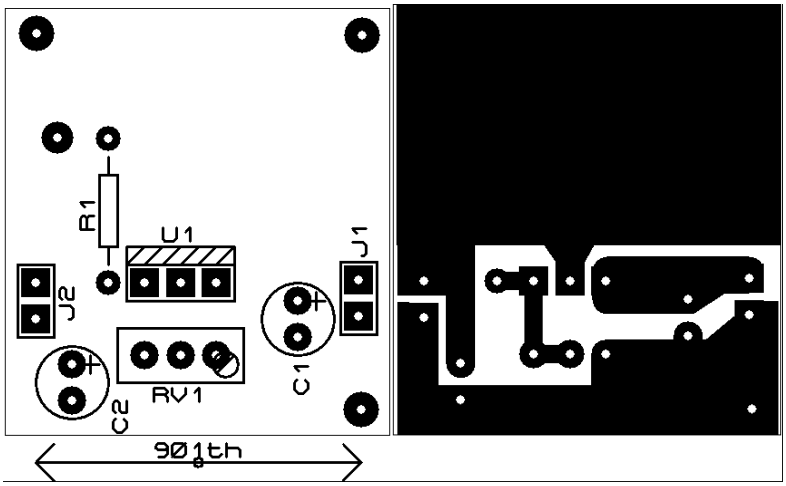
FT232 برای ایجاد ارتباط بین کامپیوتر و ماژول برای به روز کردن برنامه ی میکروکنترلر و SIM900 و همچنین تست کردن عملکرد و خطایابی استفاده می شود.

از آنجایی که این آی سی از نوع پکیج SSOP است و نمی توان به طور مستقیم بر روی برد بورد نصب شود، برای استفاده ی راحت تر از آن بر روی برد مقابل متصل می شود.

با این برد می توان از تمام امکانات FT232 بر روی برد بورد و موارد دیگر به راحتی استفاده نمود.

LCD برای نمایش ساعت و عملیات در حال اجرا به کار می رود.

به این دلیل که SIM900 از ولتاژ استانداردی استفاده نمی کند، بنابراین ولتاژ آن به وسیله ی آی سی رگولاتور متغیر LM317 تامین می شود. برای استفاده ی راحت تر از آن بر روی برد بورد، از بورد زیر استفاده شده است.



ارسال و دریافت SMS و درخواست از اپراتور

ماژول SIM900 دارای دو حالت برای SMS هست

1 – text mode : در این حالت می توان 140 حرف با کد اسکی 8 بیتی فرستاد.

2 – PDU mode : در این روش فشرده سازی انجام شده و از آنجا که هر کد 7 بیت دارد در نتیجه می توان تا 160 کاراکتر را در 140 بایت ارسال نمود.

برای فرستادن پیام فارسی هر کاراکتر دو بایت را اشکال می کند که می توان 70 کاراکتر را ارسال نمود.

در اینجا می خواهیم رفتار قطعات را زمانی که می خواهیم شارژ باقیمانده سیم کارت را درخواست می کنیم، بررسی کنیم.

از طریق گوشی موبایلی که شماره ی آن برای ماژول معین شده یک sms با متن "#1111" برای ماژول می فرستیم.

با رسیدن sms به SIM900 از طریق پورت سریال پیام زیر به میکرو کنترلر ارسال می شود

+CMTI: "SM",3

با رسیدن این پیام به میکرو مشخص می شود که یک پیام جدید در خانه ی شماره ی 3 ی سیم کارت ذخیره شده است که اکنون باید خوانده شود که برای این کار میکرو دستور زیر را از طریق ارتباط سریال برای SIM900 ارسال می کند

AT+CMGR=3

اکنون ماژول متن زیر را برای میکرو ارسال می کند

+CMGR: "REC UNREAD","+98915xxxxxxx","MODOULE NO","11/06/03,11:36:09+18"

#1111

OK

این پیام شامل اطلاعاتی از قبیل زیر است:

دستور خوانده شدن پیام مورد نظر

پیام قبلا خوانده نشده است

شماره ای که پیام را ارسال کرده است

زمان رسیدن پیام

کد کلید enter

متن پیام

OK

اکنون وظیفه ی میکرو کنترلر درک و بررسی موارد بالاست. ابتدا شماره ی تلفن فرستنده بررسی می شود تا از شماره ی مشخصی باشد. سپس متن پیام بررسی می شود که با یافتن متن 1111 مشخص می شود درخواست اعلام باقیمانده ی اعتبار از اپراتور شده است. پس میکرو دستور زیر را برای ماژول ارسال می کند .

AT+CUSD=1,”*140*1#”,15

این دستور همان دستور درخواست شارژ از اپراتور ایرانسل است . (البته می توانیم با دستور AT+CUSD=0 این درخواست را پایان دهیم.) اکنون باید منتظر جاب اپراتور بمانیم. بعد از دریافت پاسخ SIM900 متن زیر را برای میکرو ارسال می کند

+CUSD: 0,"متن شامل تاریخ و میزان شارژ و اطلاعات دیگر"

از درون متن، میکرو اطلاعات لازم را ذخیره می کند تا در مرحله ی بعد برای کاربر ارسال نماید. دستور زیر این متن را برای کاربر ارسال می کند که میکرو به ماژول می فرستد

AT+CMGS = " +98915xxxxxxx"(Enter)

میکرو منتظر میماند تا > بازگردد

>your message < ctrl-z>

سپس متن را برای SIM900 ارسال می کند و سپس با ارسال کاراکتر متناظر ctrl-z (0x1a)، متن ارسال می شود و در انتها SIM900 بعد از ارسال با موفقیت پیام، متن زیر را برای میکروکنترلر می فرستد

AT+CMGS: xxx

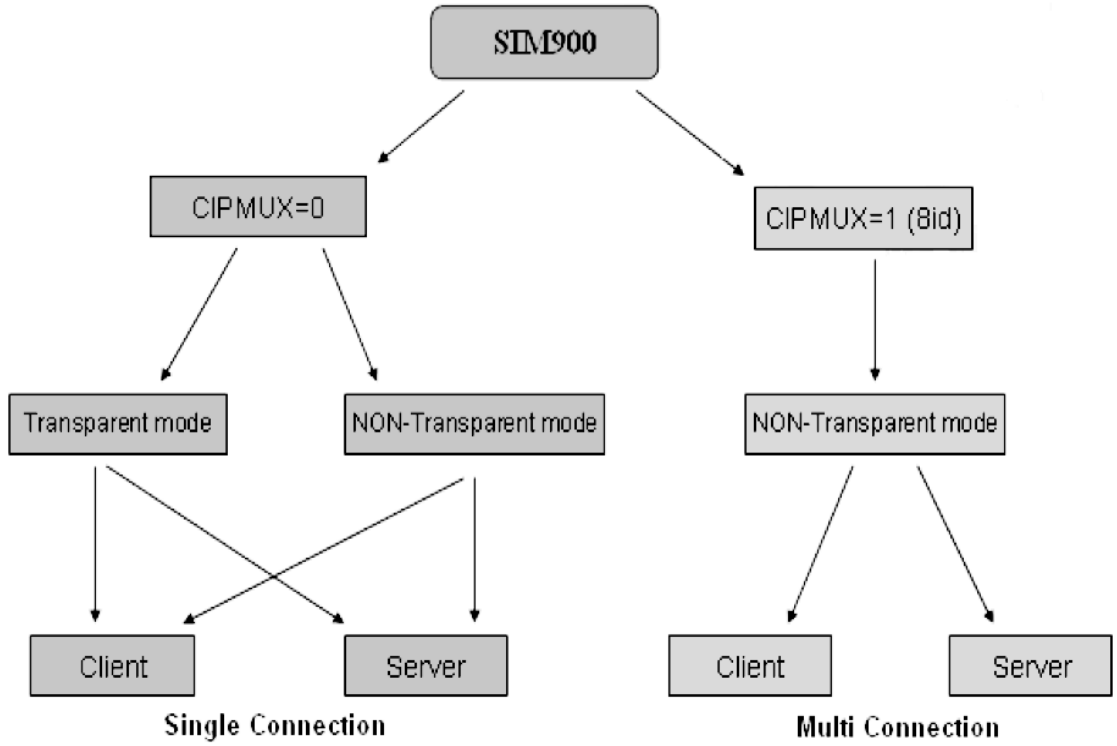
OK

بعد از این یک پیام برای تلفن همراه کاربر شامل متن میزان اعتبار باقیمانده می رسد .

تبادل داده به وسیله ی GPRS

به وسیله ی ماژول SIM900 امکان ارتباط با اینترنت ازطریق GPRS وجود دارد. SIM900 اطلاعات را از طریق پروتکل http ارسال و دریافت می کند که برای این منظور به سرور و کلاینت نیاز است. در صورت وجود سرور ، SIM900 از طریق GPRS و شبکه ی اینترنت اطلاعات را برای سرور مورد نظر ارسال می کند و از آن دریافت می نماید. همچنین می توان ارتباط را بین یک ماژول SIM900 که به عنوان سرور تنظیم شده و دیگری که به عنوان کلاینت تنظیم شده برقرار نمود.

در شکل زیر نحوه ی این ارتباطات نشان داده شده است:



SIM900 در دو حالت چند کاربره و یک کاربره می تواند کارکند (multiplex) البته زمانی که به عنوان کلاینت عمل نماید و در حالتی که به عنوان سرور عمل کند فقط به صورت تکی تنظیم می شود که در مجموع 8 ماژول می توانند به یکدیگر متصل شوند و یکی می تواند به عنوان سرور عمل کند.در زیر نمونه هایی از دستورات تنظیم ماژول و ارسال داده را مشاهده می کنیم:

AT+CGATT?
+CGATT: 1

OK

AT+CSTT=”CMNET” تنظیم APN

OK

AT+CIICR آماده کردن ارتباط

OK

AT+CIFSR دادن آی پی محلی

10.78.245.128

AT+CIPSTART=”TCP”,”116.228.221.51”, “8500” شروع اتصال

OK
CONNECT OK

AT+CIPSEND
> hello TCP server ارسال داده به سرور , CTRL+Z (0x1a) to send.

SEND OK اطلاعات توسط سرور دریافت شد

hello sim900 متن بازگشتی از سرور

CLOSED

از این طریق می توان اطلاعات را از طریق شبکه ی اینترنت به هر جایی منتقل کرد و محیط مورد نظر را کنترل نمود.

ارتباط صوتی

با متصل کردن یک گوشی شامل میکروفن و بلندگو و متصل کردن یک صفحه کلید، می توان یک تماس صوتی برقرار نمود.

ATD+2783xxxx;	شماره گیری
ATDL	شماره گیری مجدد
ATA	جواب به تماس دریافتی
ATH	قطع تماس

برنامه نویسی

مترجم زبان C مورد استفاده در این پروژه، نرم افزار Code vision 2.05 است.

تحلیل برنامه ی میکروکنترلر

قسمت هایی که توضیح داده نشده توسط نرم افزار به صورت اتوماتیک و با انجام تنظیمات تکمیل شده است.

```
/*****************************************************
Chip type           : ATmega16
AVR Core Clock frequency: 8.000000 MHz
Data Stack size        : 256
*****************************************************/

#include <mega16.h>
#include <stdlib.h>

// Alphanumeric LCD Module functions
#include <alcd.h>

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
```

```
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
```

در این محل کلید enter و کاراکتر ترکیب کلیدهای control + z با دستور #define مشخص می شود و تابع wait_loop اعلان اولیه می شود.

```
//////////////////////////////////////////////////////////////////////////
#define enter 0x0d
#define ctrl_z 0x1a
void wait_loop();
//////////////////////////////////////////////////////////////////////////

// USART Receiver buffer
#define RX_BUFFER_SIZE 128
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
```

```
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
  {
  rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
  // special case for receiver buffer size=256
  if (++rx_counter == 0)
    {
#else
  if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
  if (++rx_counter == RX_BUFFER_SIZE)
    {
    rx_counter=0;
#endif
    rx_buffer_overflow=1;
    }
  }
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
```

تابع wait_loop در این قسمت استفاده می شود

این تابع در مواقعی که میکرو برای رسیدن کاراکتری توسط ارتباط سریال منتظر مانده اجرا می شود و بسیاری از کارهای کنترلی در این تابع (حلقه) انجام می شود.

```
while (rx_counter==0)  wait_loop();
data=rx_buffer[rx_rd_index++];
#if RX_BUFFER_SIZE != 256
if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#endif
#asm("cli")
```

```
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Timer1 output compare A interrupt service routine
```

تعریف متغیرهای ساعت و نمایش آن

```
unsigned char second=0;
unsigned char minute=0;
unsigned char hour=0;
char time[15];
bit send_cond=0;

interrupt [TIM1_COMPA] void timer1_compa_isr(void)
```

این تابع، تابع وقفه ی تایمر – کانتر یک است که هر ثانیه یکبار اجرا می شود و تغییرات لازم را در ساعت انجام می دهد.

```
{
  second++;
  if(second>=60)
  {
    minute++;
    second = 0;
    if(minute>=60)
    {
      hour++;
      minute = 0;
      if(hour>=24) hour=0;
    }
  }
```

در این قسمت زمانی که sms حاوی اطلاعات محیط ارسال می شود، تنظیم می گردد.

```
   if((hour==0)&&(minute==5)&&(second==30))send_cond=1;
}

// Declare your global variables here
```

تعریف رشته ها برای ارسال با ارتباط سریال

ماژول را آماده ی ارسال sms به شماره ی مورد نظر می کند

```
flash char at_cmgs[]=
{'A','T','+','C','M','G','S','=','"','+','9','8','9','1','5','3','8','2','5','6','7','5','"','\r'};
```

درخواست از اپراتور برای اعلام شارژ باقی مانده.

```
flash char at_cusd[]=
{'A','T','+','C','U','S','D','=','1',',','"','*','1','4','0','*','1','#','"',',','1','5','\r'};
```

تعریف رشته ی پاک کردن sms ها از حافظه ی سیم کارت

```
flash char at_cmgda[]=
{'A','T','+','C','M','G','D','A','=','"','D','E','L',' ','A','L','L','"','\r'};
```

تعریف متغیر های نگه داری شرایط گلخانه

```
int currenttemp,currentdamp,mintemp,maxtemp,mindamp,maxdamp;
```

انتظار تا دریافت کاراکتر مورد نظر ch

```
void wait_to_get(char ch)
{
   while(ch != getchar());
}
```

تابع فرستادن شرایط محیط به کاربر

```
void send_condition()
{
   putsf(at_cmgs);
   wait_to_get('>');

   printf("condition: currenttemp:%d;currentdamp:%d;
         mintemp:%d;maxtemp:%d;mindamp:%d;maxdamp:%d;",
         currenttemp,currentdamp,mintemp,maxtemp,mindamp,maxdamp);

   putchar(ctrl_z);
}
```

در این تابع کدی که با فرمت #xxxx توسط کاربر به ماژول فرستاده و بازیابی شده، درخواست آن اجرا می شود.

```
void check_request(int request)
{
                                        ارسال درخواست به اپراتور برای اعلام شارژ باقیمانده
  if(request==1111)
  {
    putsf(at_cusd);
    wait_to_get('K');
  }
                                        ارسال شرایط محیط
  else if(request==2222)
  {
    send_condition();
  }
      در صورتی که درخواست مورد نظر اشتباه باشد پاسخ Answer of xxxx فرستاده می شود.
  else
  {
    putsf(at_cmgs);
    wait_to_get('>');
    printf("Answer of %d",request);
    putchar(ctrl_z);
  }
}

int sms_location;
```

زمانی که پیامی به ماژول برسد این تابع اجرا می شود. در داخل این تابع مکان پیام دریافتی شناسایی شده و در آخر دستوری برای خواندن آن ارسال می شود.

```
void cmti(void)
{
  char c[1];

  wait_to_get(',');
  c[0]=getchar();
  sms_location=atoi(c);
  if(sms_location>=10)sms_location/=10;
  printf("AT+CMGR=%d\r",sms_location);
}
```

وقتی دستور خواندن پیام ارسال شد توسط این تابع پیام خوانده می شود.

```
void cmgr(void)
{
  unsigned char x;
  int code;
  char number[4];

  for(x=0;x<3;x++)wait_to_get('"');
  for(x=0;x<9;x++)getchar();
  for(x=0;x<4;x++)number[x]=getchar();
```

در اینجا چهار رقم آخر شماره ی تلفن بدست می آید

```
  code=atoi(number);
```

اگر چهار رقم آخر 5675 باشد بعد از این کد ارسالی توسط کاربر بدست می آید و سپس تابع check_request اجرا می شود.

در صورت لزوم اگر حافظه sms سیم کارت پر شود پیام ها پاک می شوند.

```
  if(code==5675)
  {
    wait_to_get('#');
    for(x=0;x<4;x++)number[x]=getchar();

    wait_to_get('K');
    code=atoi(number);
    if(code > 9999)code/=10;

    if(sms_location > 5)
    {
      putsf(at_cmgda);
      wait_to_get('K');
    }

    check_request(code);
  }
```

اگر پیام دریافتی از کابر مورد نظر نباشد، این دستوات اجرا می شود.

```
  else
  {
    lcd_clear();
```

```
      lcd_gotoxy(0,0);
      lcd_puts(number);
      wait_to_get('K');

      if(sms_location > 5)
      {
         putsf(at_cmgda);
         wait_to_get('K');
      }
   }

}
```

وقتی sms با موفقیت ارسال شد این تابع اجرا می شود.

```
void cmgs(void)
{
   wait_to_get('K');
}
```

دریافت درخواستی که از اپراتور برای اعلام شارژ شده بود و ارسال آن به کاربر

```
void cusd(void)
{
   unsigned char x=0;
   char buf[64];
   char c=0;

   wait_to_get('"');

   do
   {
      c=getchar();
      if(x<63)buf[x++]=c;
   }
   while(c != '"');

   wait_to_get(enter);

   putsf(at_cmgs);
   wait_to_get('>');
   puts(buf);
   putchar(ctrl_z);
```

```
}
```

تابعی که در زمان بیکاری میکرو از دریافت پیامها، کارهای کنترلی را انجام می دهد.

```
void wait_loop()
{
```

در ایجا به دلیل نبود شرایط محیط برای اندازه گیری مقادیری به متغیر ها داده شده است

```
   currenttemp=30;
   currentdamp=50;
   mintemp=20;
   maxtemp=40;
   mindamp=40;
   maxdamp=70;
```

در این مکان بعد از دریافت اطلاعات واقعی می توان داده ها را پردازش نمود

در زمان مورد نظر (ساعت مورد نظر یا در شرایط بحرانی یا ...) تابع ارسال شرایط به کاربر را اجرا می کند.

```
   If(send_cond)
   {
      send_cond =0;
      send_condition();
   }
```

نمایش بر روی lcd

```
   sprintf(time,"time: %d:%d:%d",hour,minute,second);
   lcd_clear();
   lcd_gotoxy(0,0);
   lcd_puts(time);
}

void main(void)
{
// Declare your local variables here

char buffer;

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
DDRA=0x00;
PORTB=0x00;
DDRB=0x00;
PORTC=0x00;
DDRC=0x00;
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 7.813 kHz
// Mode: CTC top=OCR1A
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x1E;
OCR1AL=0x85;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;

// Analog Comparator initialization
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
ADCSRA=0x00;
// SPI initialization
SPCR=0x00;
// TWI initialization
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTB Bit 0
// RD - PORTB Bit 1
// EN - PORTB Bit 2
// D4 - PORTA Bit 0
// D5 - PORTA Bit 1
// D6 - PORTA Bit 2
// D7 - PORTA Bit 3
// Characters/line: 16
lcd_init(16);
```

```
// Global enable interrupts
#asm("sei")

lcd_putsf("hello");
```

در این حلقه دستوراتی که از SIM900 به میکرو کنترلر فرستاده می شوند بررسی می شوند و در صورت دریافت یک دستور مشخص عملیات آن انجام می شود.

با دریافت هر دستور تابعی با همان نام اجرا می شود که این توابع در بالا تعریف شده اند

تابع getchar() منتظر می ماند تا کاراکتر جدیدی به پورت سریال برسد

```
while (1)
 {

      buffer=getchar();
      if(buffer=='+')
      {
        buffer=getchar();
        if(buffer=='C')
        {
          buffer=getchar();
          if(buffer=='M')
          {
            buffer=getchar();
            if(buffer=='T')
            {
              buffer=getchar();
              if(buffer=='I')
              {
                buffer=getchar();
                if(buffer==':')    cmti();
              }
            }
            else if(buffer=='G')
            {
              buffer=getchar();
              if(buffer=='R')
              {
                buffer=getchar();
                if(buffer==':')    cmgr();
              }
```

وقتی پیامی برسد

وقتی پیامی خوانده شود

```
                else if(buffer=='S')
                {
                  buffer=getchar();
                  if(buffer==':')   cmgs();                    وقتی پیامی ارسال شود
                }
              }
            }
            else if(buffer=='U')
            {
              buffer=getchar();
              if(buffer=='S')
              {
                buffer=getchar();
                if(buffer=='D')
                {
                  buffer=getchar();
                  if(buffer==':')   cusd();                    وقتی درخواست انجام شود
                }
              }
            }
          }
        }
      lcd_putsf("!");                                          وقتی هیچ یک از موارد بالا نباشد

    }
  }
```

پیوست

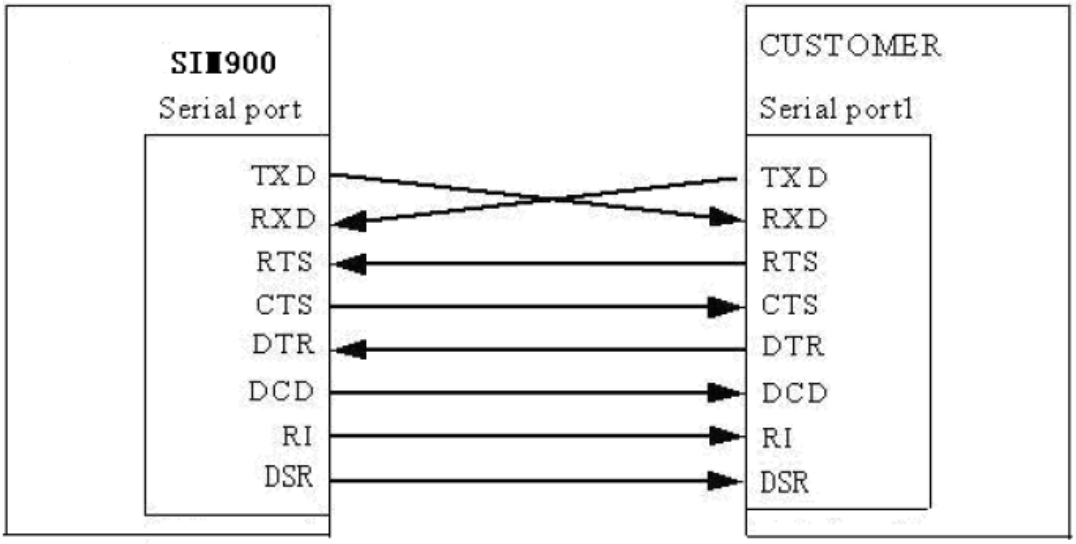
اطلاعات بیشتر از SIM900

PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
VBAT	I	3 VBAT pins are dedicated to connect the supply voltage. The power supply of SIM900 has to be a single voltage source of VBAT= 3.4V...4.5V. It must be able to provide sufficient current in a transmit burst which typically rises to 2A	Vmax= 4.5V Vmin=3.4V Vnorm=4.0V	
VRTC	I/O	Current input for RTC when the battery is not supplied for the system. Current output for backup battery when the main battery is present and the backup battery is in low voltage state.	Vmax=3.15V Vmin=2.0V Vnorm=3.0V Iout(max)= 300uA Iin=2 uA	If the RTC function is enabled, a battery or capacitor should be connected with the VRTC pin. Otherwise the VRTC pin can be keep open.
VDD_EXT	O	2.8V output power supply	Vmax=2.95V Vmin=2.70V Vnorm=2.80V	If unused, keep open.

I_{VBAT}	Average supply current	POWER DOWN mode SLEEP mode		30 1.5		uA mA
		IDLE mode GSM 850 EGSM 900 DCS1800 PCS1900		 22 22 22 22		 mA
		TALK mode GSM 850 EGSM 900 DCS1800 PCS1900		 235 241 158 166		 mA
		DATA mode, GPRS (3 Rx,2Tx) GSM 850 EGSM 900 DCS1800 PCS1900		 435 444 287 299		 mA
		DATA mode, GPRS (4 Rx,1Tx) GSM 850 EGSM 900 DCS1800 PCS1900		 266 270 191 202		 mA

Serial Port

TXD, RXD, CTS, RTS, DSR, DTR, DCD, RI



TXD, RXD --- Data Line
CTS, RTS --- Hardware Flow Control Line
DTR --- Sleep Mode Control Line
DCD --- Data Mode
RI --- Incoming Call, SMS, Arouse host.
DSR --- Reserve

Table 18: Behaviours of the RI

State	RI respond
Standby	HIGH
Voice calling	Change LOW, then: （1）Change to HIGH when establish calling. （2）Use AT command ATH, the RI pin changes to HIGH. （3）Sender hangs up, change to HIGH.
Data calling	Change LOW, then: （1）Change to HIGH when establish calling. （2）Use AT command ATH, the RI changes to HIGH.
SMS	When receive SMS, The RI will change to LOW and hold low level about 1200 ms, then change to HIGH.
URC	Some URCs triggers 1200ms low level on RI. ***For more details, please refer to document [10]***

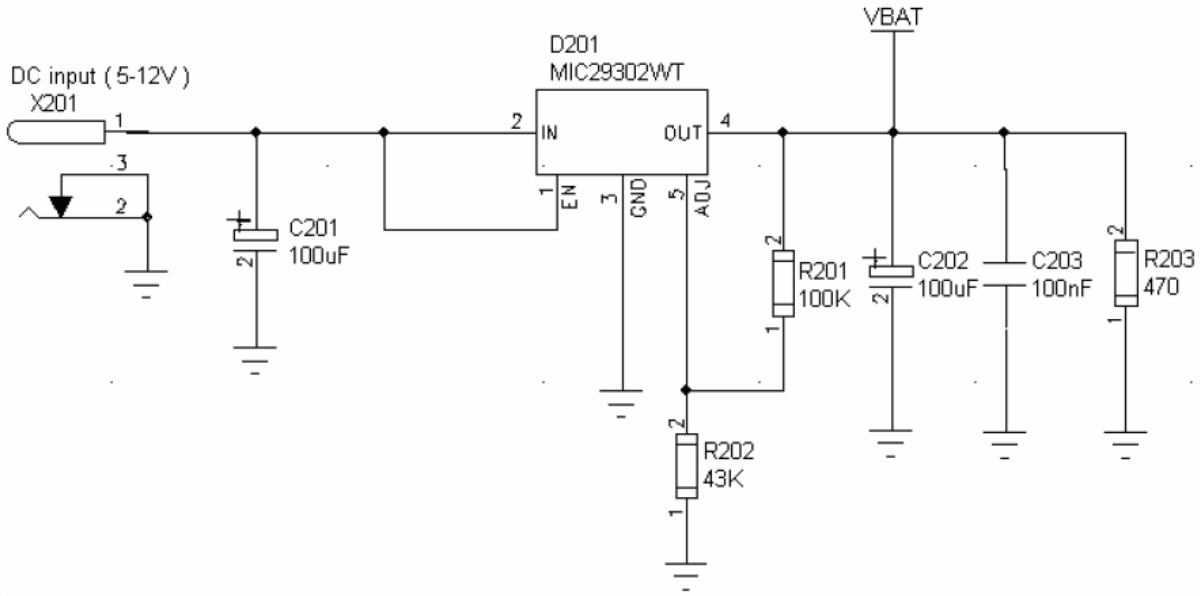


Figure 4: Reference circuit of the source power supply input

3.3.3 Monitoring Power Supply

To monitor the supply voltage, you can use the "AT+CBC" command which include a parameter: voltage value (in mV).

The voltage is continuously measured at intervals depending on the operating mode. The displayed voltage (in mV) is averaged over the last measuring period before the AT+CBC command is executed.

The following figures show various sample circuits for RTC backup.

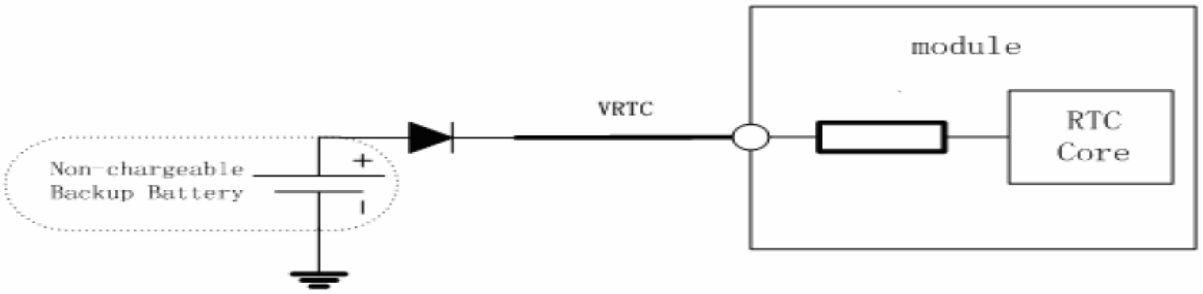
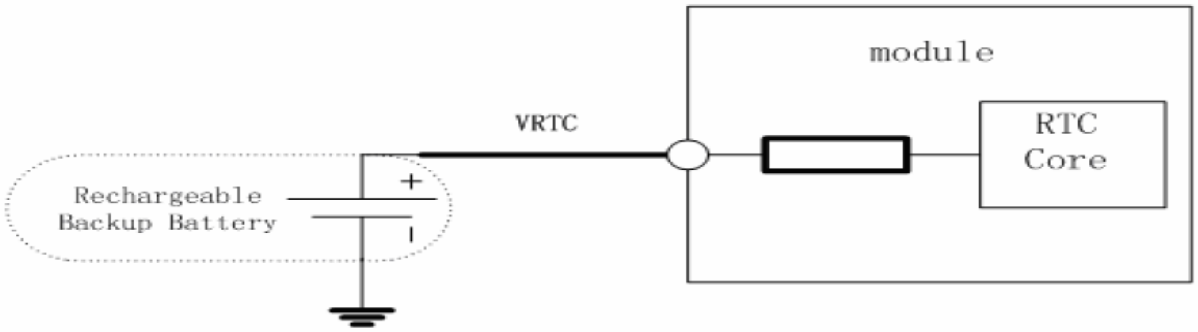


Figure 16: RTC supply from non-chargeable battery



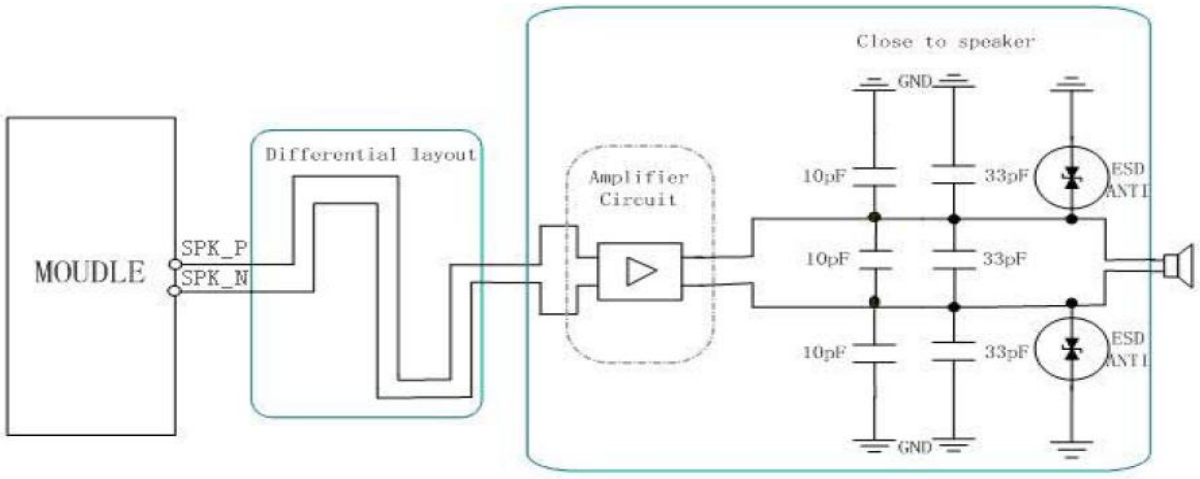


Figure 24: Speaker interface with amplifier configuration

3.9.2 Microphone Interfaces Configuration

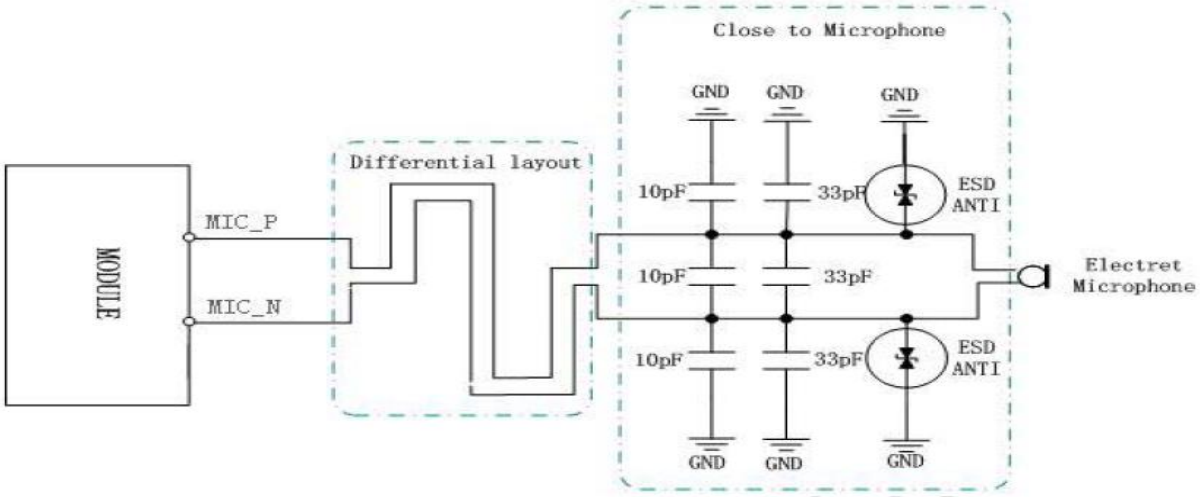


Figure 25: Microphone interface configuration

3.9.3 Earphone Interface Configuration

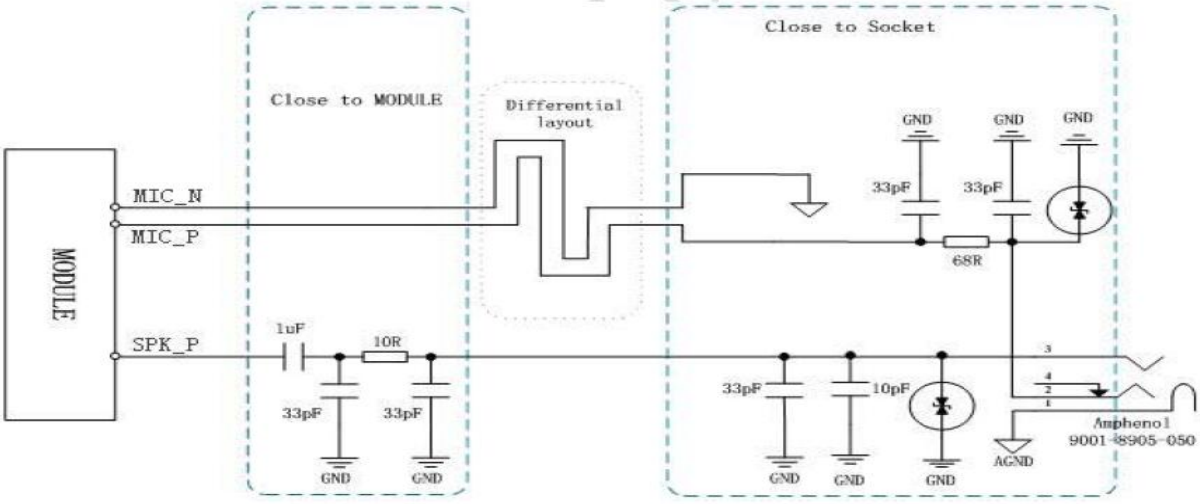
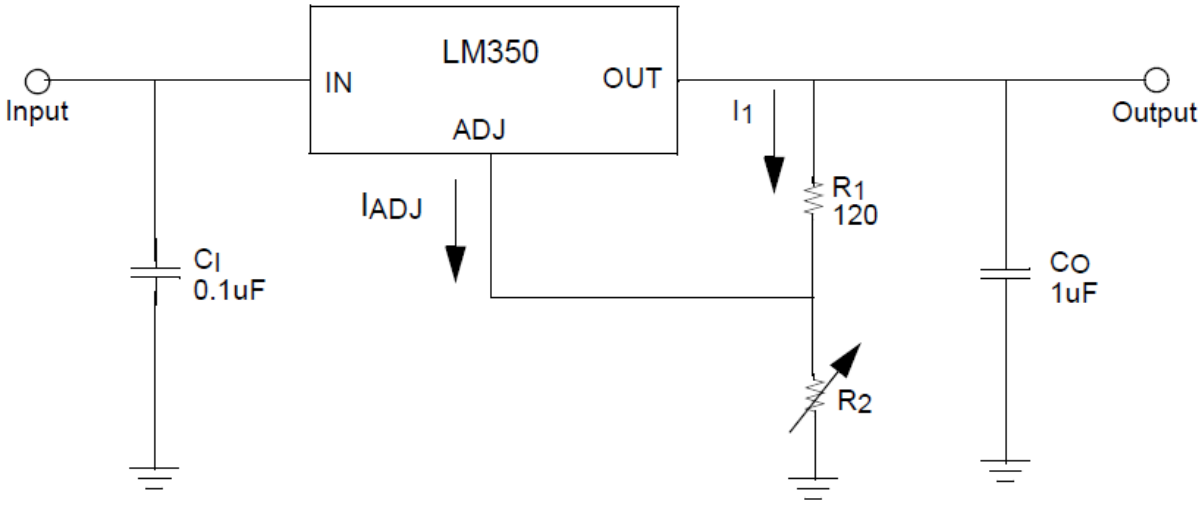


Figure 26: Earphone interface configuration

مدار نمونه برای منبع تغذیه ی متغییر با آی سی های lm317 و lm350

Lm350 = 3A max

Lm317 = 1.5A max



$V_O = 1.25V(1+R_2/R_1) + I_{ADJ}R_2$ **~0**

منابع

سایت های اینترنتی:

سایت سازنده ی SIM900 :

http://wm.sim.com

سایت سازنده ی FT232 :

http://www.ftdichip.com

سایت سازنده ی ATMEGA16 :

http://www2.atmel.com

سایت گفتگو در مورد الکترونیک (انگلیسی) :

http://www.edaboard.com

سایت های فارسی درباره ی الکترونیک :

http://www.kavirelectronic.ir
http://www.iranmicro.ir