

به نام خداوند بخشنده و مهربان

آموزش راه اندازی ماژول SIM808



مقدمه :

با پیشرفت روز افزون تکنولوژی و اهمیت فراوان مبحث کنترل از راه دور سیستم ها و تجهیزات الکترونیکی با استفاده از مودم های GSM ، همچنین اهمیت سیستم های موقعیت یاب جهانی GPS این آموزش مختصر با هدف آشنایی با نحوه راه اندازی GSM و GPS ماژول های SIM808 برای آن دسته از کاربرانی که علاقه مند کار با این ماژول ها می باشند ارائه گردیده ، امید است این آموزش مورد استفاده علاقه مندان عزیز قرار گیرد. در پایان از تمامی دوستان دعوت می کنم تا با ارائه نظرات و پیشنهادات خود به نشانی info@avaelectronic.ir ما را در هرچه بهتر شدن این آموزش یاری کنند.

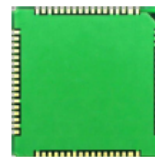
با امید فردایی بهتر

جواد زحمتی

بهار ۱۳۹۵

GSM/GPRS+GPS Module

SIM808



ماژول SIM808:

ماژول SIM808 یک ماژول GSM/GPRS/GPS می باشد نسخه جدید تر Sim908 بوده با این وجود پایه های یکسانی نسبت به SIM908 ندارد ولی از لحاظ عملکرد جایگزین مناسبی برای SIM908 می باشد، لذا کسانی که با ماژولهای قدیمی Sim900 و Sim908 کار کرده اند به راحتی می توانند ماژول Sim808 را راه اندازی کنند. این ماژول از چهار باند 850/900/1800/1900MHZ پشتیبانی می کند . برقراری ارتباط صوتی ، ارسال و دریافت SMS ، MMS ، دیتا ، اتصال به اینترنت و سیستم موقعیت یاب ماهواره ای GPS از جمله ویژگی های ماژول SIM808 میباشد.

خصوصیات عمومی:

چهار باند 850/900/1800/1900MHZ

GPRS کلاس 10/12

ایستگاه موبایل GPRS کلاس B

ابعاد : 24*24*2.6 میلی متر

وزن : 3.3 گرم

کنترل از طریق AT Command

محدوده ولتاژ کاری 3.4 تا 4.4 ولت

مصرف توان پایین

دمای کاری : -45 تا 85 درجه سانتیگراد

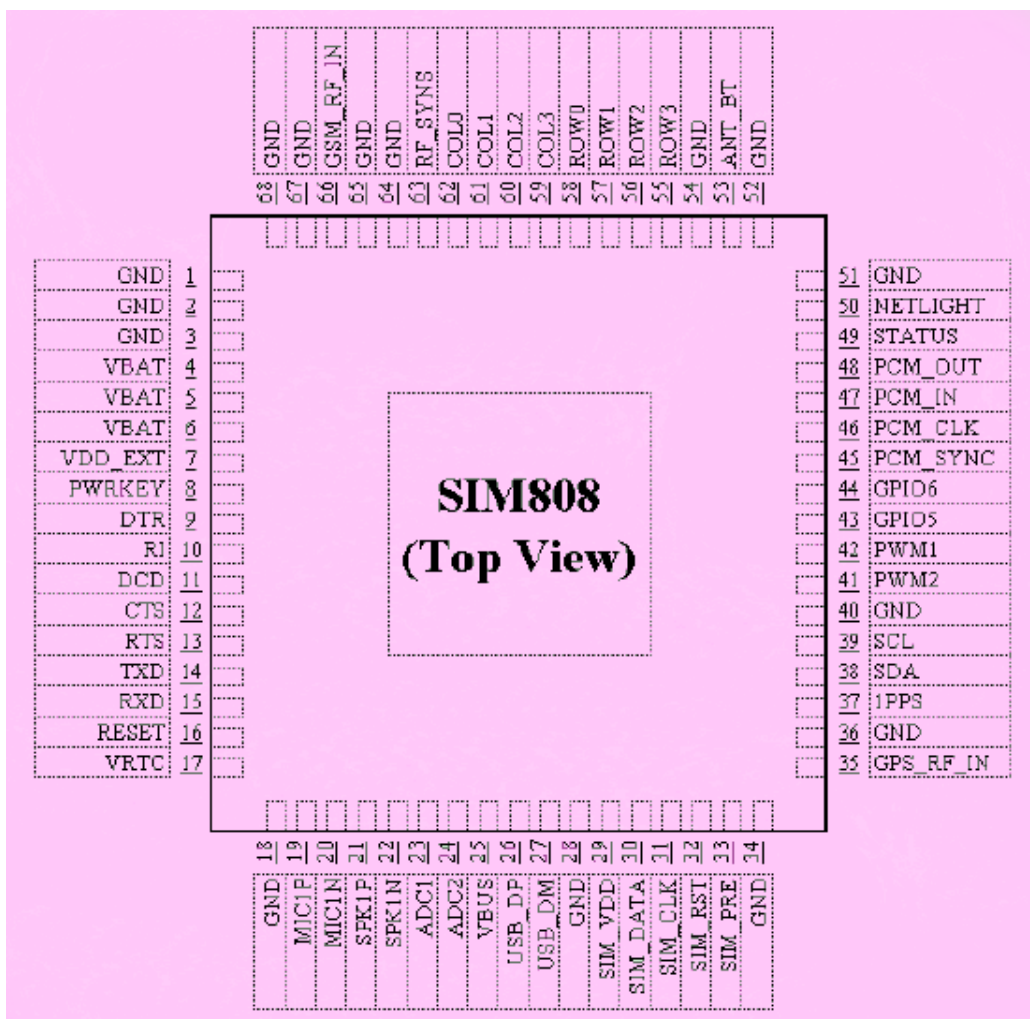
قابلیت ها:

ارسال اطلاعات هر دستگاه از راه دور

ارسال و دریافت اطلاعات از طریق GSM،GPRS و SMS

قابلیت کار کردن با تمامی اپراتورهای مخابراتی (همراه اول،ایرانسل،رایتل،تالیا و اسپادان)

مجهز کردن دستگاه های الکترونیکی به یک موبایل هوشمند





همان طور که مشاهده می کنید برد راه انداز شامل چیپ SIM808، سوکت سیم کارت، کلید Power سه عدد LED برای نمایش STAT، NET، POWER و سه عدد سوکت انتن برای متصل کردن انتن های GPS، GSM و Bluetooth می باشد.

ماژول SIM808 با متصل کردن تغذیه به پایه های (GND) و (VCC) و نگه داشتن کلید PWR روی هدربرد SIM808 به مدت 1 ثانیه راه اندازی میشود در ابتدا LED مربوط به NET با سرعت بیشتری شروع به چشمک زدن می کند وقتی به شبکه متصل شد این چشمک زدن با سرعت کمتری انجام می شود.

برای تست ماژول، با استفاده از مبدل USB به سریال پایه های شماره 1 (GND)، 14 (TX) و 15 (RX) برد راه انداز SIM808 را به کامپیوتر متصل می کنیم.



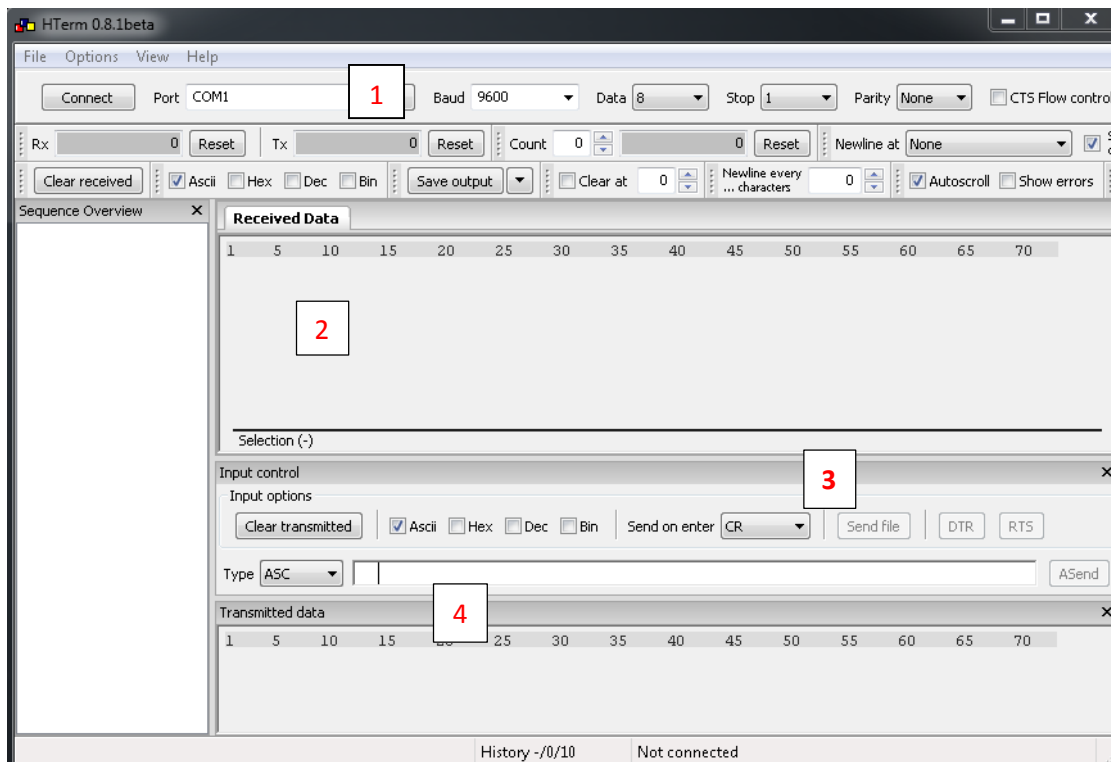
برای ارسال و دریافت اطلاعات بین کامپیوتر و ماژول SIM808 می توان از نرم افزارهای مختلفی مثل HyperTerminal، HTerm و غیره استفاده کرد.

نحوه ارتباط ماژول با کامپیوتر:

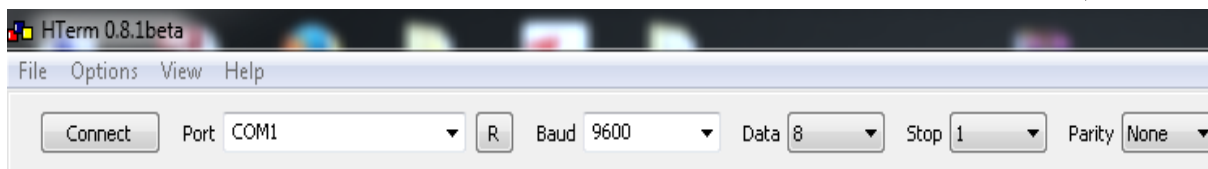


معرفی و تنظیم محیط نرم افزار HTerm:

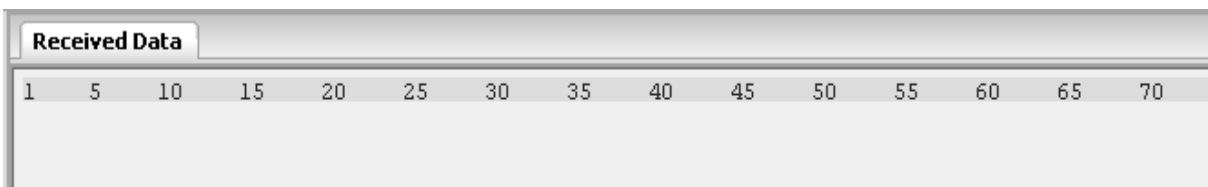
بعد از اینکه برد راه انداز SIM808 را از طریق کابل PL2303 به کامپیوتر متصل کردیم نرم افزار HTerm را اجرا می کنیم.



1: در این محیط و در قسمت port، پورتی که SIM808 به آن متصل است را انتخاب می کنیم. Baudrate را 9600، Data را 8 بیتی، stop bits را 1 و parity را None (بدون بیت توازن) انتخاب می کنیم.



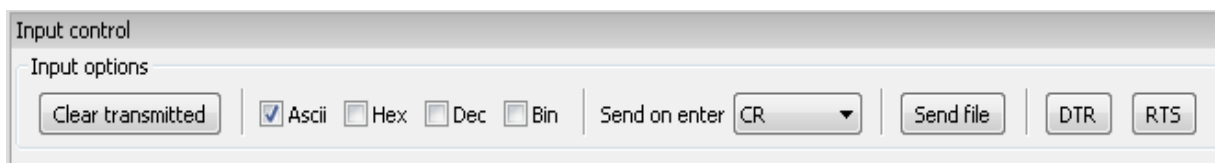
2: در این قسمت اطلاعات دریافت شده نمایش داده می شود.



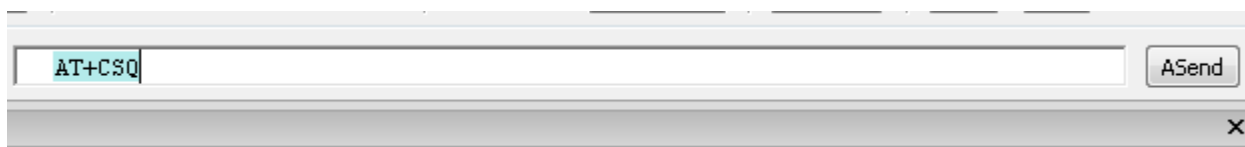
3: در این قسمت از نرم افزار می توان با زدن تیک هریک از گزینه ها ، اطلاعات رابه صورت هگز، دسیمال و باینری مشاهده کرد .

جهت ارسال هر دستور به ماژول SIM808 و پردازش آن توسط ماژول بایستی درانتهای هر دستور کد 0X0D به ماژول ارسال شود.

با انتخاب CR در زبانه Send on enter بعد از تایپ هر دستور در قسمت ارسال دیتا و زدن Enter این کد نیز به ماژول ارسال می شود.



4: در این قسمت دستورات مورد نظرمان را تایپ می کنیم.



معرفی برخی از AT Command :

توضیحات	AT Command
نمایش پیکربندی فعلی	AT&V
ذخیره کردن تنظیمات	AT&W
بازگرداندن به تنظیمات کارخانه ای	AT&F
نام کارخانه سازنده	AT&GMI
نمایش مدل ماژول	AT+GMM
دریافت قدرت سیگنال	AT+CSQ
تست سیم کارت	AT+CSMINS
وارد کردن پین کد	AT+CPIN="XXXX"
تماس گرفتن	ATD0937261XXXX;

تماس با آخرین شماره گرفته شده	ATDL
پاسخ دادن به تماس دریافتی	ATA
پایان تماس	ATH
فعال و غیر فعال کردن echo	ATE
نمایش همه شماره تلفن های ذخیره شده در SIM	AT+CPBF
دریافت موجودی " *141*1# " , AT+CUSD=1	AT+CUSD
انتخاب فرمت پیام PDU یا TEXT	AT+CMGF=(0,1)
نمایش لیست پیام ها	AT+CMGL= "ALL" "REC READ" "REC UNREAD"
نوشتن و ذخیره پیام کوتاه	AT+CMGW="09.." (Enter) >Message <ctrl+z>
ارسال پیام ذخیره شده	AT+CMSS
نوشتن و ارسال پیام کوتاه	AT+CMGS="09.." <Enter> >Message <ctrl+z>
خواندن پیام دریافت شده	AT+CMGR
پاک کردن پیام کوتاه	AT+CMGDA= "DEL ALL" "DEL READ" "DEL UNREAD"
نمایش Error	AT+CMEE
نمایش دمای ماژول	AT+CMTE
نمایش زمان ماژول	AT+CCLK?
تنظیم ساعت و تاریخ ماژول	AT+CCLK= "YY/MM/DD, HH:MM:SS+02"
تنظیم مقدار Buad rate	AT+IPR
تغذیه GPS	AT+CGNSPWR
دریافت مختصات جغرافیایی	AT+CGNSINF

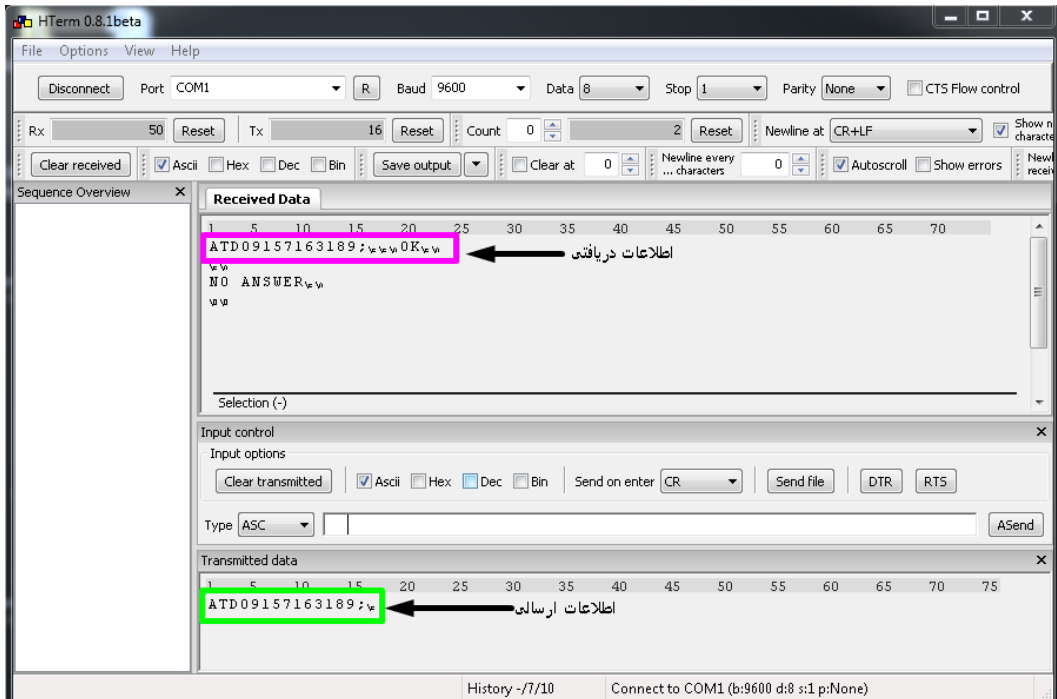
نمایش تمام گزینه ها پیش رو	AT+<X>=?	تست
خواندن و نمایش پارامترها و یا نمایش گزینه انتخاب شده	AT+<X>?	خواندن
انتخاب یک عملکرد از میان گزینه های پیش رو	AT+<X>=Value	نوشتن
عمل اجرایی کردن یک دستور را برعهده دارد	AT+<X>	اجرا

بعنوان مثال نحوه مشاهده انواع نرخ تبادل اطلاعات

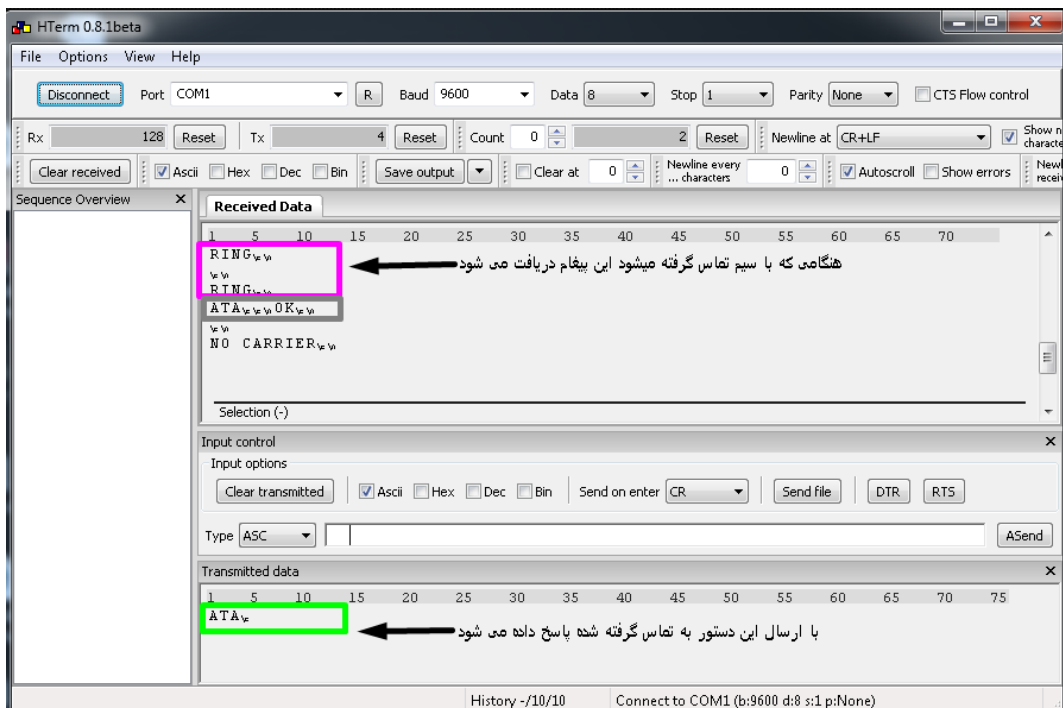
توضیحات	شکل دستور	نمایش وضعیت
+IPR: () , (0,1200,2400,4800,9600,..) OK	AT+IPR=?	تست
+IPR: 0 OK	AT+IPR?	خواندن
9600 OK	AT+IPR=9600	نوشتن

توضیحات	شکل دستور	نمایش وضعیت
با این دستور می توان با شماره دلخواه تماس گرفت	ATD0915716XXXX;	اجرا

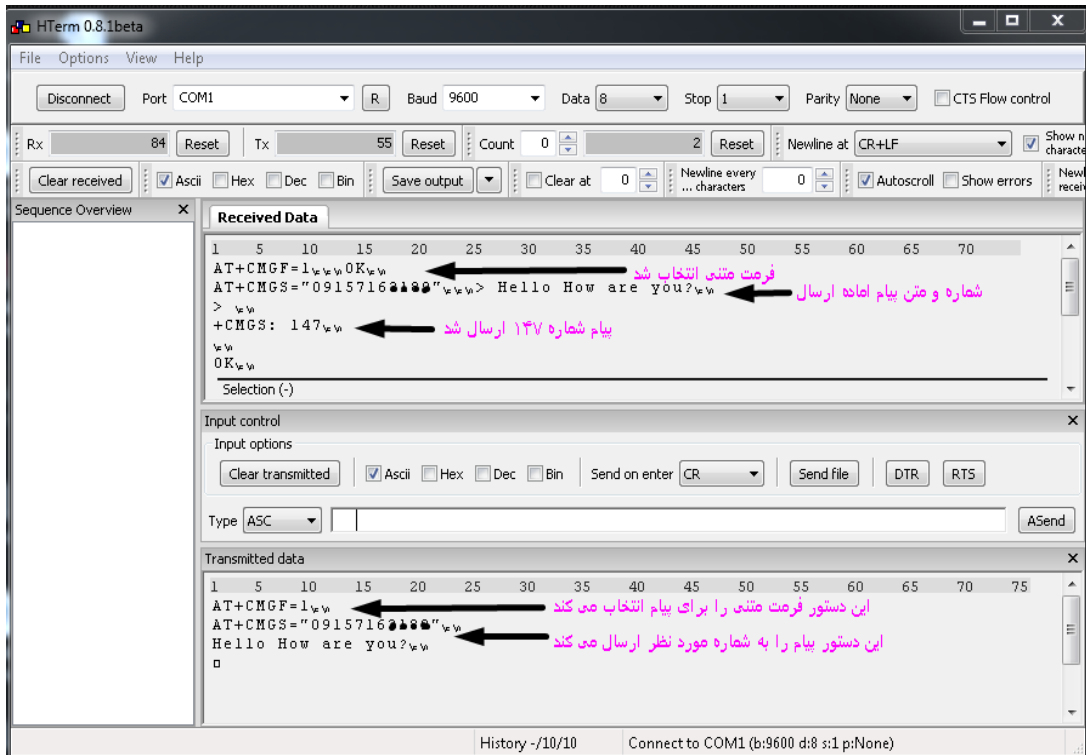
نحوه تماس گرفتن با استفاده از نرم افزار HTerm:



نحوه پاسخ به تماس دریافت شده:

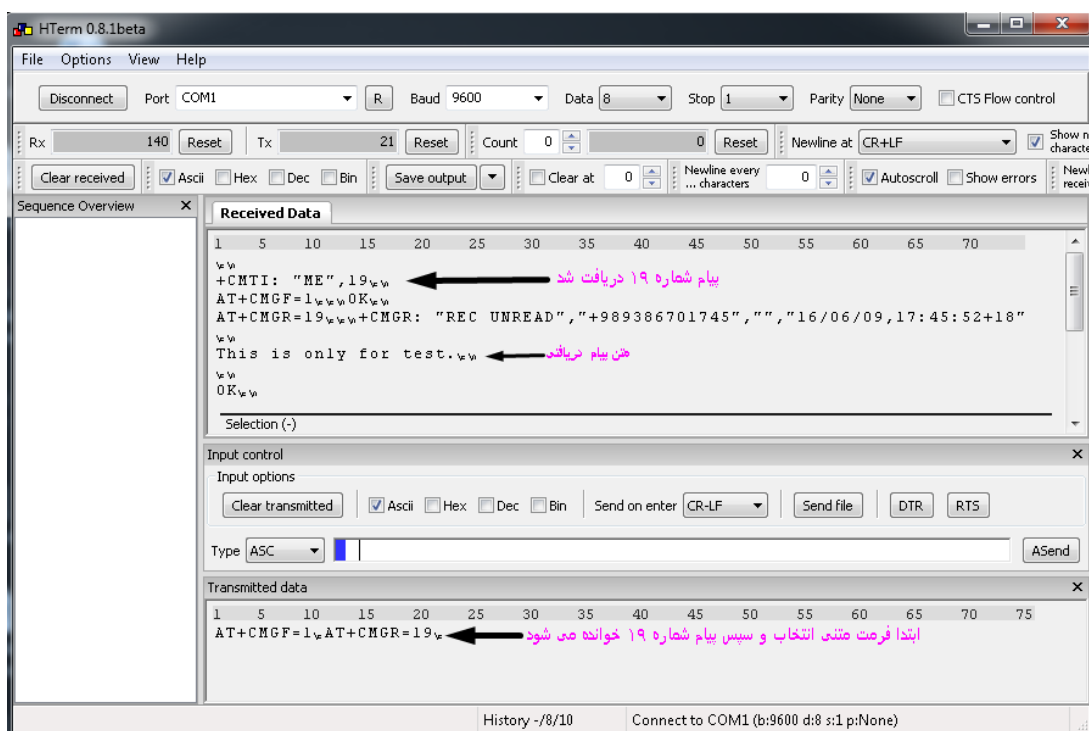


نحوه ارسال پیام کوتاه:



بعد از اینکه متن پیام ارسال شد باید کد هگز **ctrl+z** ارسال شود برای این کار در زبانه **send on enter** گزینه **None** و در زبانه **Type** گزینه **HEX** را انتخاب می کنیم و با تایپ **1A** در قسمت ارسال و زدن کلید **Enter** این کار انجام می شود.

نحوه دریافت پیام کوتاه:



نحوه دریافت مختصات جغرافیایی:

The screenshot shows the HTerm 0.8.1beta interface. The 'Received Data' pane contains the following text:

```

AT+CGNSPWR=1OK
AT+CGNSPWR=1OK
AT+CGNSINF: 1,1,20160611044133.000,36.354411,59.511111,1095.600
,0.48,221.7,1,,1.4,1.6,0.9,,,,,
OK
AT+CGNSPWR=0OK
    
```

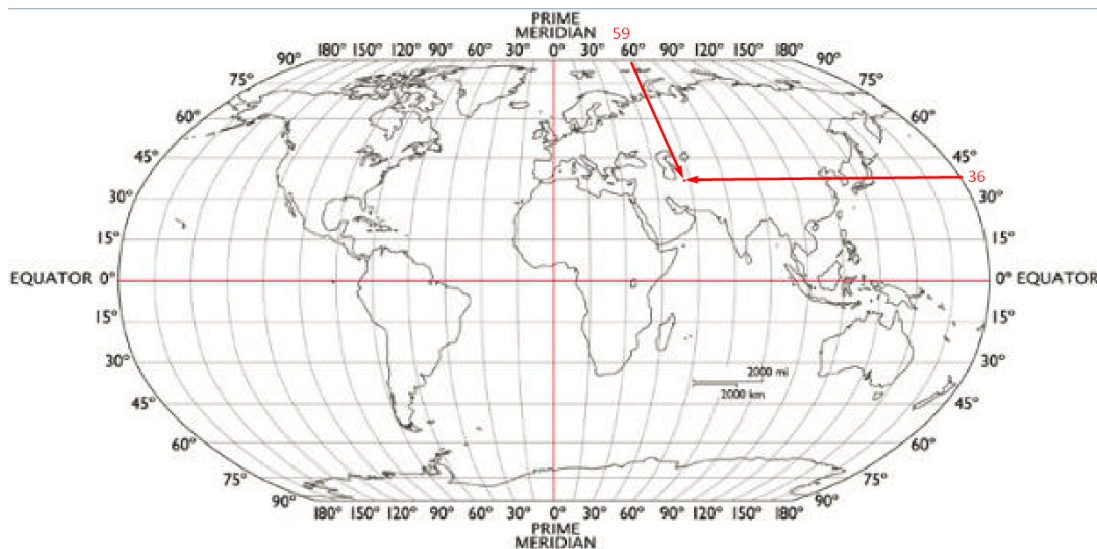
Annotations in Persian explain the data:

- ستور راه اندازی نغذ به GPS (GPS initialization command)
- طول جغرافیایی (Longitude)
- عرض جغرافیایی (Latitude)
- با این دستور تاریخ، ساعت، طول و عرض جغرافیایی و غیره را می توان از مازول دریافت کرد (With this command, you can receive date, time, longitude, and latitude from the module).
- دستور قطع نغذ به GPS (GPS power-off command)
- تاریخ (Date)
- ساعت (Time)

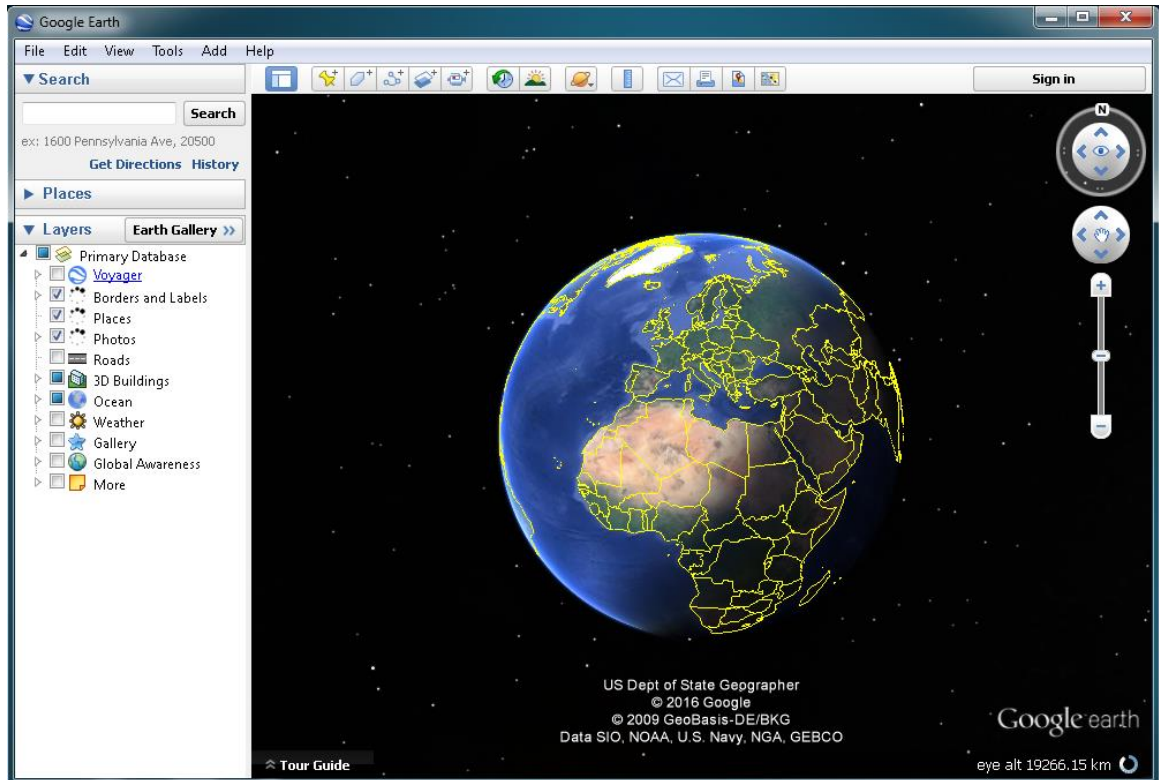
The 'Transmitted data' pane shows the commands sent:

```

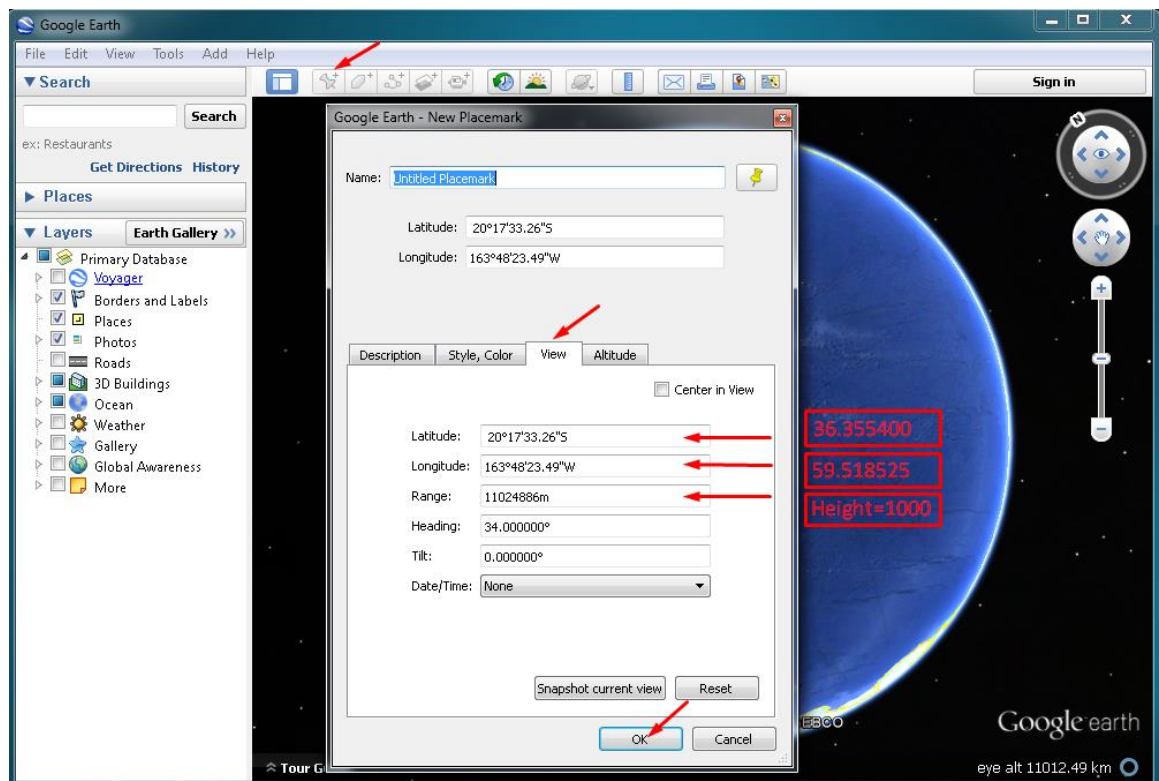
AT+CGNSPWR=1
AT+CGNSINF
AT+CGNSPWR=0
    
```



نرم افزار Google Earth را اجرا می کنیم.



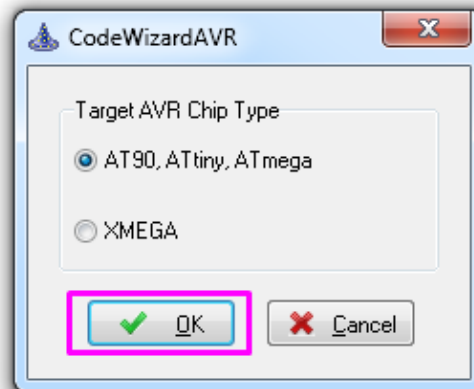
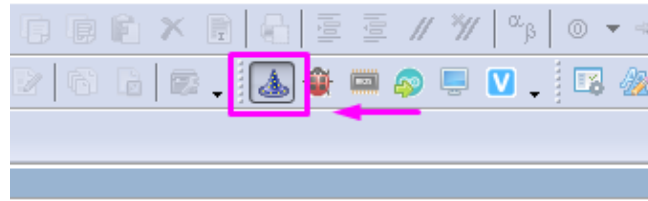
با وارد کردن طول و عرض جغرافیایی در نرم افزار Google Earth می توان موقعیت مکانی را مشاهده کرد.



ارتباط ماژول با میکروکنترلر ATmega32:

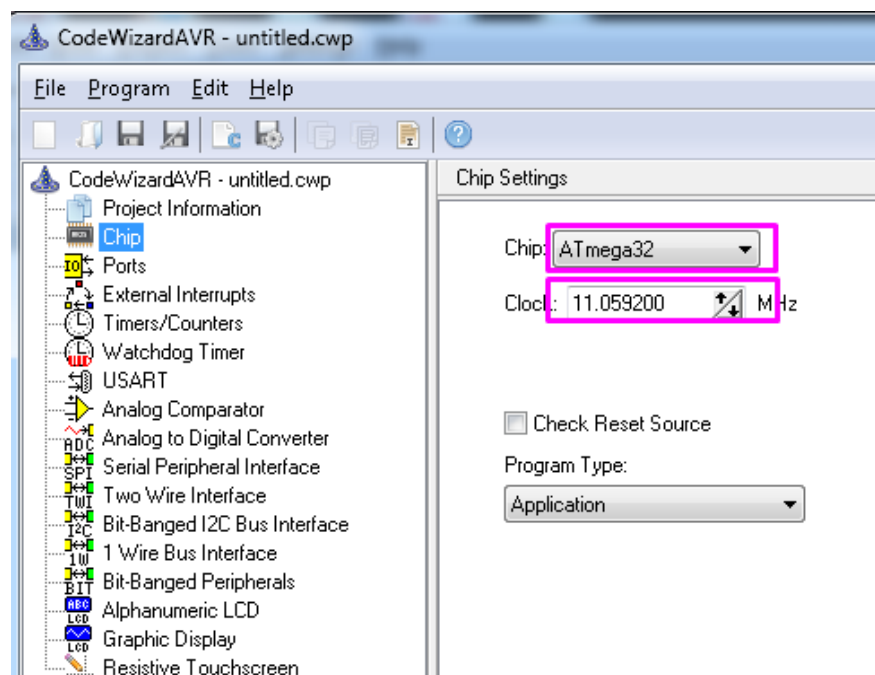
تنظیمات کدویژن:

نرم افزار کدویژن را اجرا کرده و کدویزارد را انتخاب می کنیم.



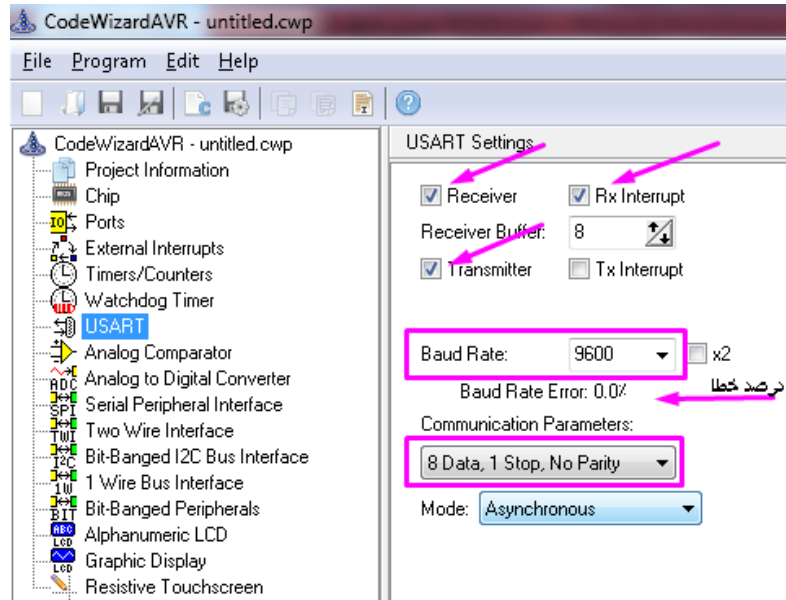
میکروکنترلر را ATMEGA32 و فرکانس را 11.059200MHz انتخاب می کنیم.

برای کاهش درصد خطای Baud Rate به صفر از کریستال خارجی 11.0592 مگاهرتز استفاده می شود.

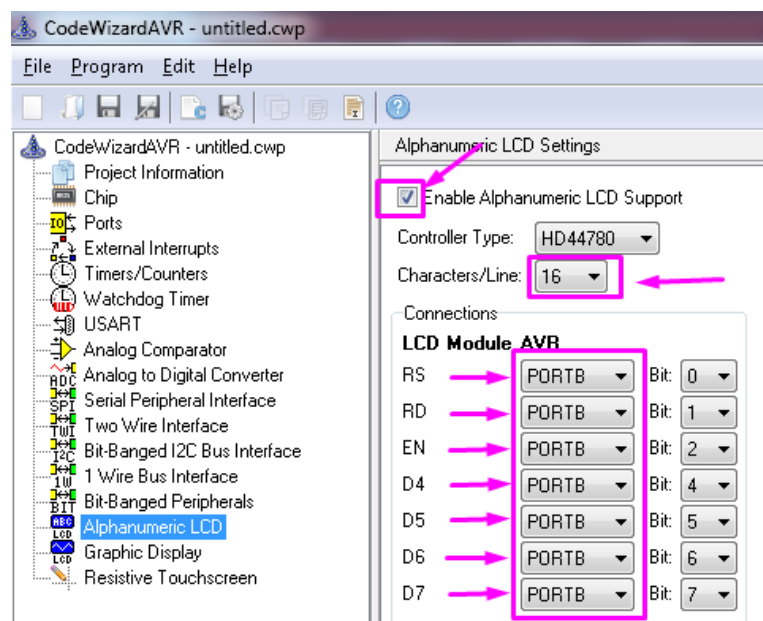


در قسمت USART Settings گیرنده ، وقفه گیرنده و فرستنده سریال را فعال می کنیم مقدار سرعت تبادل اطلاعات را 9600 ، دیتا را 8 بیتی ،

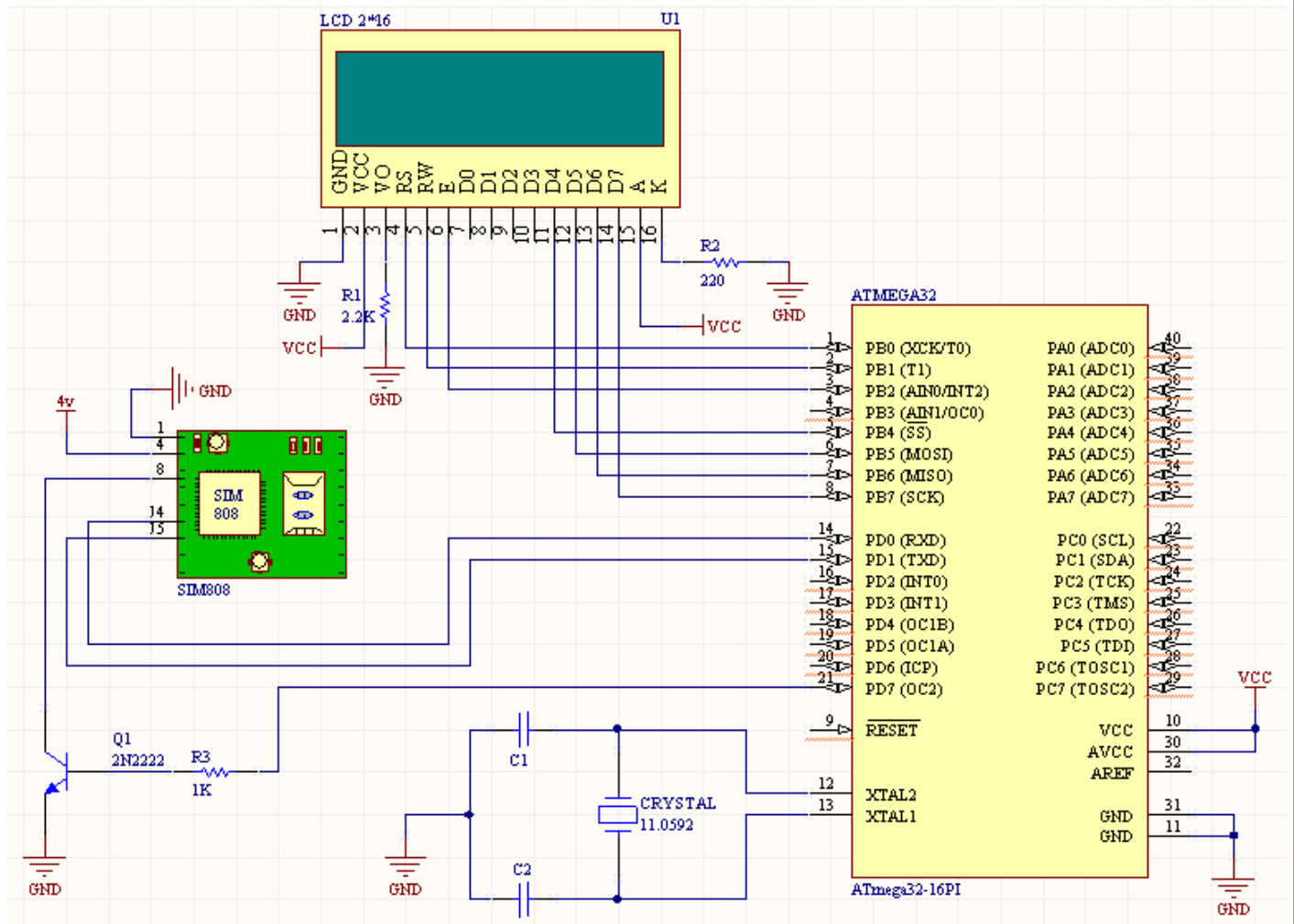
بیت پایان ارسال اطلاعات را 1 و بیت توازن None انتخاب می کنیم.



و در ادامه LCD را فعال می کنیم.



شماتیک :



کد برنامه دریافت پیام کوتاه در محیط کدویژن:
برنامه در محیط نرم افزار CodeVision 3.12:

```

CodeVisionAVR - D:\Electronic\AVR\SimCode\SIM808_3\SIM808_3.prj
File Edit Search View Project Tools Settings Help
D:\Electronic\AVR\SimCode\SIM808_3\SIM808_3.c
Notes SIM808_3.c
1 /*****
2 This program was created by the
3 CodeWizardAVR V3.12 Advanced
4 Automatic Program Generator
5 @ Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
6 http://www.hpinfotech.com
7
8 Project :
9 Version :
10 Date : 05/31/2016
11 Author :
12 Company :
13 Comments:
14
15 *****/
16
17 Chip type : ATmega32
18 Program type : Application
19 AVR Core Clock frequency: 11.059200 MHz
20 Memory model : Small
21 External RAM size : 0
22 Data Stack size : 256
23 *****/
24
25 #include <mega32.h>
26 #include <alcd.h>
27 #include <stdio.h>
28 #include <delay.h>
29
30 unsigned char Number[32];
31 unsigned char message[32];
32 unsigned char i=0,j=0,n=0;
33 unsigned char value=0,value1=0,value2=0,value3=0;
34 unsigned char m=0,i1=0;
35 char data=0;
36
37 interrupt [USART_RXC] void usart_rx_isr(void)
38 {
39 data=UDR;
40 if(value==0)
41 {
42 if(value1==1){m=data;value1=0;value=1;value3=1;i1=0;}
43 if(data==' ') i1=1;
44 if(data=='S' && i1==1 ) i1=2; چنانچه وقفه دریافت بورت سریال کارکترهای 'S','M',' ' را که بیانگر دریافت
45 if(data=='M' && i1==2 ) i1=3; یک پیام جدید است دریافت کرد متغیر value1 و به دنبال آن متغیر value3 برابر یک '1' شده و دستور
46 if(data==' ' && i1==3 ) i1=4; خواندن پیام دریافت شده ارسال می گردد.
47 if(data==',' && i1==4 ) i1=5;
48 if(i1==5){ value1=1; }
49 }
50 if(value==1)
51 {
52 if(data=='U') i1=1;
53 if(data=='N' && i1==1) i1=2;
54 if(data=='R' && i1==2) i1=3;
55 if(data=='E' && i1==3) i1=4;
56 if(data=='D' && i1==4) i1=5;
57 if(data==' ' && i1==5) i1=6;
58 if(data==',' && i1==6) i1=7;
59 if(data==' ' && i1==7) i1=8;
60 if(i1==8) value2=1;
61
62 if(value2==1 && i<=14)
63 { شماره تلفن ،پیام کوتاه دریافت شده را در سطر اول LCD نمایش می دهد.
64 lcd_gotoxy(i,0);
65 Number[i]=data;
66 lcd_putchar(Number[i]);
67 i++;
68 if(i==14){value=2;value2=0;}
69 }
70
71 }

```


متن پیام دریافت شده در سطر دوم LCD نمایش داده می شود

```
72
73 if(value==2)
74 {
75     if(data==' ')n++;
76
77     if(n>4)
78     {
79         lcd_gotoxy(j,1);
80         message[j]=data;
81         lcd_putchar(message[j]);
82         j++;
83     }
84
85 }
86
87
88 void main(void)
89 {
90
91
92 // Input/Output Ports initialization
93 // Port A initialization
94 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
95 DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
96 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
97 PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
98
99 // Port B initialization
100 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
101 DDRB=(0<<ddb7) | (0<<ddb6) | (0<<ddb5) | (0<<ddb4) | (0<<ddb3) | (0<<ddb2) | (0<<ddb1) | (0<<ddb0);
102 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
103 PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
104
105 // Port C initialization
106 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
107 DDRC=(0<<ddc7) | (0<<ddc6) | (0<<ddc5) | (0<<ddc4) | (0<<ddc3) | (0<<ddc2) | (0<<ddc1) | (0<<ddc0);
108 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
109 PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
110
111 // Port D initialization
112 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
113 DDRD=(0<<ddd7) | (0<<ddd6) | (0<<ddd5) | (0<<ddd4) | (0<<ddd3) | (0<<ddd2) | (0<<ddd1) | (0<<ddd0);
114 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
115 PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
116
117 // Timer/Counter 0 initialization
118 // Clock source: System Clock
119 // Clock value: Timer 0 Stopped
120 // Mode: Normal top=0xFF
121 // OCO output: Disconnected
122 TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
123 TCNT0=0x00;
124 OCR0=0x00;
125
126 // Timer/Counter 1 initialization
127 // Clock source: System Clock
128 // Clock value: Timer1 Stopped
129 // Mode: Normal top=0xFFFF
130 // OCA output: Disconnected
131 // OCB output: Disconnected
132 // Noise Canceler: Off
133 // Input Capture on Falling Edge
134 // Timer1 Overflow Interrupt: Off
135 // Input Capture Interrupt: Off
136 // Compare A Match Interrupt: Off
137 // Compare B Match Interrupt: Off
138 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
139 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
140 TCNT1H=0x00;
141 TCNT1L=0x00;
142 ICR1H=0x00;
143 ICR1L=0x00;
144 OCR1AH=0x00;
145 OCR1AL=0x00;
146 OCR1BH=0x00;
147 OCR1BL=0x00;
148
149 // Timer/Counter 2 initialization
150 // Clock source: System Clock
151 // Clock value: Timer2 Stopped
152 // Mode: Normal top=0xFF
153 // OC2 output: Disconnected
154 ASSR=0<<AS2;
155 TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
156 TCNT2=0x00;
157 OCR2=0x00;
158
159 // Timer(s)/Counter(s) Interrupt(s) initialization
160 TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
```

```

161
162 // External Interrupt(s) initialization
163 // INT0: Off
164 // INT1: Off
165 // INT2: Off
166 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
167 MCUCSR=(0<<ISC2);
168
169 // USART initialization
170 // Communication Parameters: 8 Data, 1 Stop, No Parity
171 // USART Receiver: On
172 // USART Transmitter: On
173 // USART Mode: Asynchronous
174 // USART Baud Rate: 9600
175 UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
176 UCSRB=(1<<RXCIEN) | (0<<TXCIEN) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
177 UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
178 UBRRH=0x00;
179 UBRRL=0x33;
180
181 // Analog Comparator initialization
182 // Analog Comparator: Off
183 // The Analog Comparator's positive input is
184 // connected to the AIN0 pin
185 // The Analog Comparator's negative input is
186 // connected to the AIN1 pin
161
162 // External Interrupt(s) initialization
163 // INT0: Off
164 // INT1: Off
165 // INT2: Off
166 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
167 MCUCSR=(0<<ISC2);
168
169 // USART initialization
170 // Communication Parameters: 8 Data, 1 Stop, No Parity
171 // USART Receiver: On
172 // USART Transmitter: On
173 // USART Mode: Asynchronous
174 // USART Baud Rate: 9600
175 UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
176 UCSRB=(1<<RXCIEN) | (0<<TXCIEN) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
177 UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
178 UBRRH=0x00;
179 UBRRL=0x33;
180
181 // Analog Comparator initialization
182 // Analog Comparator: Off
183 // The Analog Comparator's positive input is
184 // connected to the AIN0 pin
185 // The Analog Comparator's negative input is
186 // connected to the AIN1 pin
187 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
188 SFIOR=(0<<ACME);
189
190 // ADC initialization
191 // ADC disabled
192 ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
193
194 // SPI initialization
195 // SPI disabled
196 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
197
198 // TWI initialization
199 // TWI disabled
200 TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
201
202 // Alphanumeric LCD initialization
203 // Connections are specified in the
204 // Project/Configure/C Compiler/Libraries/Alphanumeric LCD menu:
205 // RS - PORTE Bit 0
206 // RD - PORTE Bit 1
207 // EN - PORTE Bit 2
208 // D4 - PORTE Bit 4
209 // D5 - PORTE Bit 5
210 // D6 - PORTE Bit 6
211 // D7 - PORTE Bit 7
212 // Characters/line: 16
213
214 delay_ms(1500);
215 PORTD.7=1;
216 delay_ms(1500);
217 PORTD.7=0;
218 delay_ms(5000);
219
220 lcd_init(16);
221
222 puts("AT\r");
223 delay_ms(1500);
224
225 puts("ATEO\r");
226 delay_ms(1500);

```

با یک کردن بین شماره ۷ پورت D با یک تاخیر ۱ ثانیه ای ماژول SIM808 را روشن می کنیم.

با ارسال دستور ATEO به ماژول، Echo را غیر فعال می کنیم.

با ارسال دستور AT+CMGF=1 ، فرمت متنی را برای پیام های دریافتی انتخاب می کنیم.

```
227 puts ("AT+CMGF=1\r");
228 delay_ms (1500);
229
230
231 #asm("sei")
232
233
234
235 while (1)
236 {
237     if(value3==1)
238     {
239         puts ("AT+CMGR=");
240         delay_ms (1500);
241
242         switch(m)
243         {
244
245             case '1':
246             {
247                 puts ("1\r");
248                 delay_ms (2000);
249             }
250             break;
251             case '2':
252             {
253                 puts ("2\r");
254                 delay_ms (2000);
255             }
256             break;
257             case '3':
258             {
259                 puts ("3\r");
260                 delay_ms (2000);
261             }
262             break;
263
264
265             case '4':
266             {
267                 puts ("4\r");
268                 delay_ms (2000);
269             }
270             break;
271             case '5':
272             {
273                 puts ("5\r");
274                 delay_ms (2000);
275             }
276             break;
277             case '6':
278             {
279                 puts ("6\r");
280                 delay_ms (2000);
281             }
282             break;
283
284
285             case '7':
286             {
287                 puts ("7\r");
288                 delay_ms (2000);
289             }
290             break;
291             case '8':
292             {
293                 puts ("8\r");
294                 delay_ms (2000);
295             }
296             break;
297             case '9':
298             {
299                 puts ("9\r");
300                 delay_ms (2000);
301             }
302             break;
303         }
304         value3=0;
305     }
306 }
307 }
```

خواندن پیام دریافت شده به واسطه ارسال دستور AT+CMGR به ماژول به همراه ارسال شماره پیام دریافتی انجام می شود.

کد برنامه دریافت طول و عرض جغرافیایی:

```
CodeVisionAVR - D:\Electronic\AVR\SimCode\SIM808GPS\SIM808GPS.prj
File Edit Search View Project Tools Settings Help
D:\Electronic\AVR\SimCode\SIM808GPS\SIM808GPS.c
Notes SIM808GPS.c
1 /*****
2 This program was created by the
3 CodeWizardAVR V3.12 Advanced
4 Automatic Program Generator
5 @ Copyright 1998-2014 Pavel Haiduc, HP InfoTech s.r.l.
6 http://www.hpinfotech.com
7
8 Project :
9 Version :
10 Date : 05/31/2016
11 Author :
12 Company :
13 Comments:
14
15
16 Chip type : ATmega32
17 Program type : Application
18 AVR Core Clock frequency: 11.059200 MHz
19 Memory model : Small
20 External RAM size : 0
21 Data Stack size : 256
22 *****/
23
24 #include <mega32.h>
25 #include <alcd.h>
26 #include <stdio.h>
27 #include <delay.h>
28
29 unsigned char i=0,i1=0,i2=0;
30 unsigned char length[10];
31 unsigned char latitude[10];
32
33 interrupt [USART_RXC] void usart_rx_isr(void)
34 {
35 char data;
36 data=UDR;
37
38 if(i==3 && i1<=8)
39 {
40 latitude[i1]=data; // با شماره تعداد کارکترهای " , " موجود در رشته دریافت شده از طریق پروتکل سریال ، می توانیم عرض جغرافیایی را
41 lcd_gotoxy(i1,0); // در ارایه latitude ذخیره کنیم و بروی LCD نمایش دهیم.
42 lcd_putchar(latitude[i1]);
43 i1++;
44 }
45
46 if(i==4 && i2<=8)
47 {
48 length[i2]=data; // چنانچه تعداد کارکترهای " , " دریافت شده برابر ۴ و i2<=8 گردید طول جغرافیایی را در ارایه length ذخیره و
49 lcd_gotoxy(i2,1); // در نهایت در سطر دوم LCD نمایش داده می شود.
50 lcd_putchar(length[i2]);
51 i2++;
52 }
53
54 if(i2>9){puts("AT+CGNSPWR=0\r");} // هنگامی که مختصات جغرافیایی دریافت شد با ارسال دستور AT+CGNSPWR=0 به ماژول ، تغذیه GPS را غیر
55 // فعال می کنیم.
56 if(data=='\r')i++;
57
58 }
59 }
60
61 void main(void)
62 {
63 // Input/Output Ports initialization
64 // Port A initialization
65 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
```

```

66 DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
67 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
68 PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
69
70 // Port B initialization
71 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
72 DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
73 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
74 PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
75
76 // Port C initialization
77 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
78 DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
79 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
80 PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
81
82 // Port D initialization
83 // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
84 DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
85 // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
86 PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
87
88 // Timer/Counter 0 initialization
89 // Clock source: System Clock
90 // Clock value: Timer 0 Stopped
91 // Mode: Normal top=0xFF
92 // OCO output: Disconnected
93 TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
94 TCNT0=0x00;
95 OCR0=0x00;
96
97 // Timer/Counter 1 initialization
98 // Clock source: System Clock
99 // Clock value: Timer1 Stopped
100 // Mode: Normal top=0xFFFF
101 // OCA output: Disconnected
102 // OCB output: Disconnected
103 // Noise Canceler: Off
104 // Input Capture on Falling Edge
105 // Timer1 Overflow Interrupt: Off
106 // Input Capture Interrupt: Off
107 // Compare A Match Interrupt: Off
108 // Compare B Match Interrupt: Off
109 TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
110 TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
111 TCNT1H=0x00;
112 TCNT1L=0x00;
113 ICR1H=0x00;
114 ICR1L=0x00;
115 OCR1AH=0x00;
116 OCR1AL=0x00;
117 OCR1BH=0x00;
118 OCR1BL=0x00;
119
120 // Timer/Counter 2 initialization
121 // Clock source: System Clock
122 // Clock value: Timer2 Stopped
123 // Mode: Normal top=0xFF
124 // OC2 output: Disconnected
125 ASSR=0<<AS2;
126 TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) | (0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
127 TCNT2=0x00;
128 OCR2=0x00;
129
130 // Timer(s)/Counter(s) Interrupt(s) initialization
131 TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
132
133 // External Interrupt(s) initialization
134 // INT0: Off
135 // INT1: Off
136 // INT2: Off
137 MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
138 MCUCSR=(0<<ISC2);
139
140 // USART initialization
141 // Communication Parameters: 8 Data, 1 Stop, No Parity
142 // USART Receiver: On
143 // USART Transmitter: On
144 // USART Mode: Asynchronous
145 // USART Baud Rate: 9600
146 UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
147 UCSRB=(1<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCS22) | (0<<RXB8) | (0<<TXB8);
148 UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<U8BS) | (1<<UCS21) | (1<<UCS20) | (0<<UCPOL);
149 UBRRH=0x00;
150 UBRL=0x33;
151
152 // Analog Comparator initialization
153 // Analog Comparator: Off
154 // The Analog Comparator's positive input is
155 // connected to the AINO pin

```

```

156 // The Analog Comparator's negative input is
157 // connected to the AIN1 pin
158 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
159 SFIOR=(0<<ACME);
160
161 // ADC initialization
162 // ADC disabled
163 ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
164
165 // SPI initialization
166 // SPI disabled
167 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
168
169 // TWI initialization
170 // TWI disabled
171 TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
172
173 // Alphanumeric LCD initialization
174 // Connections are specified in the
175 // Project/Configure/C Compiler/Libraries/Alphanumeric LCD menu:
176 // RS - PORTE Bit 0
177 // RD - PORTE Bit 1
178 // EN - PORTE Bit 2
179 // D4 - PORTE Bit 4
180 // D5 - PORTE Bit 5
181 // D6 - PORTE Bit 6
182 // D7 - PORTE Bit 7
183 // Characters/line: 16
184
185 lcd_init(16);
186
187 #asm("sei")
188
189 delay_ms(1500);
190 PORTD.7=1;
191 delay_ms(1500);
192 PORTD.7=0;
193 delay_ms(5000);
194
195 puts("AT\r");
196 delay_ms(2000);
197
198 puts("AT+CGNSPWR=1\r"); // با ارسال دستور AT+CGNSPWR=1 ماژول SIM808 را فعال می کنیم.
199 delay_ms(1500);
200
201 for(i=0;i<6;i++) // بعد از فعال شدن GPS مدت زمانی باید سیری شود.
202 delay_ms(2000);
203 i=0;
204
205 puts("AT+CGNSINF=1\r"); // سپس با ارسال دستور AT+CGNSINF=1 می توان مختصات جغرافیایی را از ماژول دریافت کرد.
206 delay_ms(2000);
207
208
209 while (1)
210 {
211
212 }
213

```

پایان