UART GPS NEO-6M-C 模块使用教程

产品特性：

使用 NEO-6M 模组，自带高增益有源天线
TTL 电平，兼容 3.3V/5V 系统
波特率默认为 9600，可以通过 u-center 修改
自带 IPX 接口，可以连接各种有源天线
自带可充电后备电池，可以掉电保存星历数据，实现热启动
自带 EEPROM，保存配置信息

参数：

接口特性：	TTL 电平，兼容 3.3V/5V 单片机系统
接收特性：	50 通道， GPS L1(1575.42Mhz) C/A 码，SBAS:WAAS/EGNOS/MSAS
定位精度：	2.5mCEP (SBAS:2.0mCEP)
更新速率：	最大 5Hz(默认 1HZ)
捕获时间：	冷启动：27S（最快）
	热启动：1S
捕获追踪灵敏度：	-161dBm
通信协议：	NMEA(默认)/UBX Binary
串口通信波特率：	4800、9600(默认)、19200、38400、57600、115200、230400
工作温度：	-40 摄氏度 ~85 摄氏度
工作电压：	2.7V-5.0V(VCC 引脚输入)
工作电流：	45mA
TXD/RXD 阻抗：	510 欧

应用案例：

导航仪，四轴飞行器定位

使用说明：（以接入 MCU 为例）

VCC ： 接 3.3V/5V
GND： 接 GND
TXD ： 接 MCU.RX
RXD ： 接 MCU.TX
PPS ： 接 MCU.IO 时钟脉冲输出（可不接）

在拿到模块后，可以先将模块与个人电脑通过串口相连(须经过电平转换芯片如 MAX232)。
打开串口调试助手，进行如下设置：

波特率为 9600（默认），8 位数据位，1 位停止位，无校验位，无流控制。

设置完成后，打开相应的串口，可以看到串口调试助手中有相应的数据输出。

如下图：

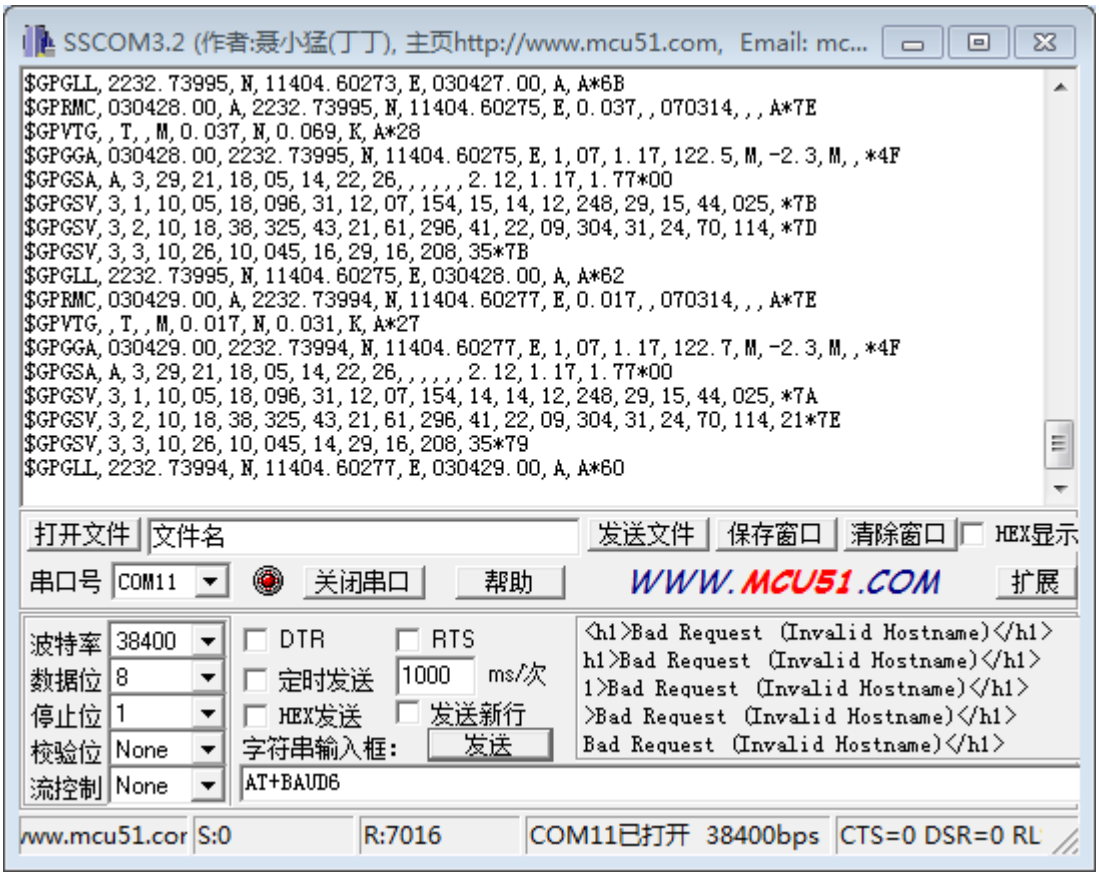


出现上图类似的数据说明硬件连接正常，但未定位。

将 GPS 模块放到阳台或窗户旁，或者直接在户外进行实验。

经过一段时间后，当串口输出类似以下数据时，说明 GPS 已经定位成功，LED 将秒闪（未定位时 LED 常亮）。

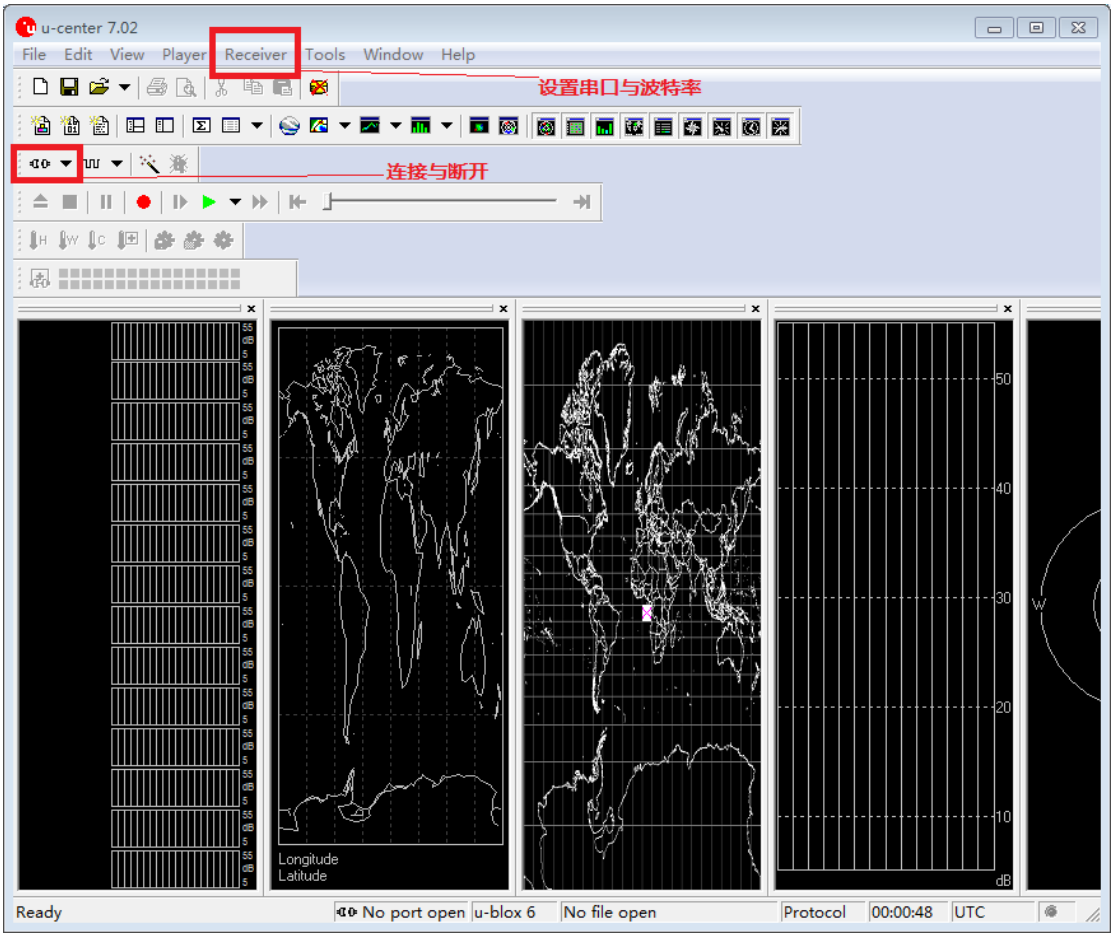
如下图：



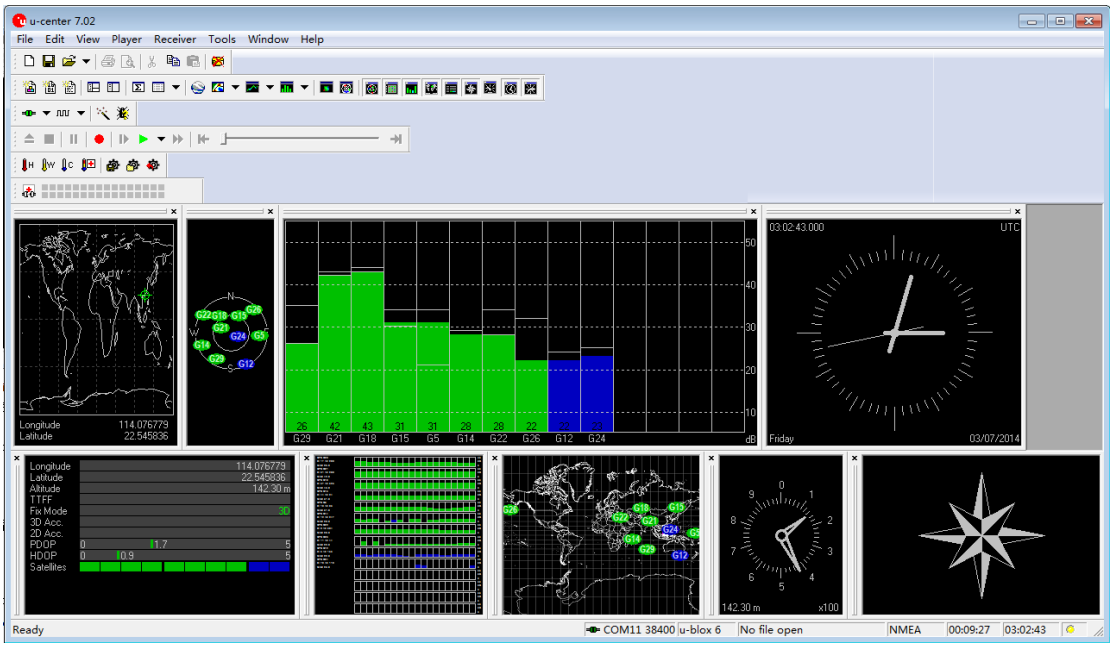
注意：GPS 模块首次定位（冷启动），在正常情况下（户外，天气良好），需要 1-3 分钟左右的时间才能定位。请耐心等待。如果天气条件不好，可能需要更长的定位时间，或者无法定位。

安装 software\u-centersetup-7.0.2.1 目录下的 u-centerSetup-7.0.2.1.exe （需联网）。

打开后如下图：



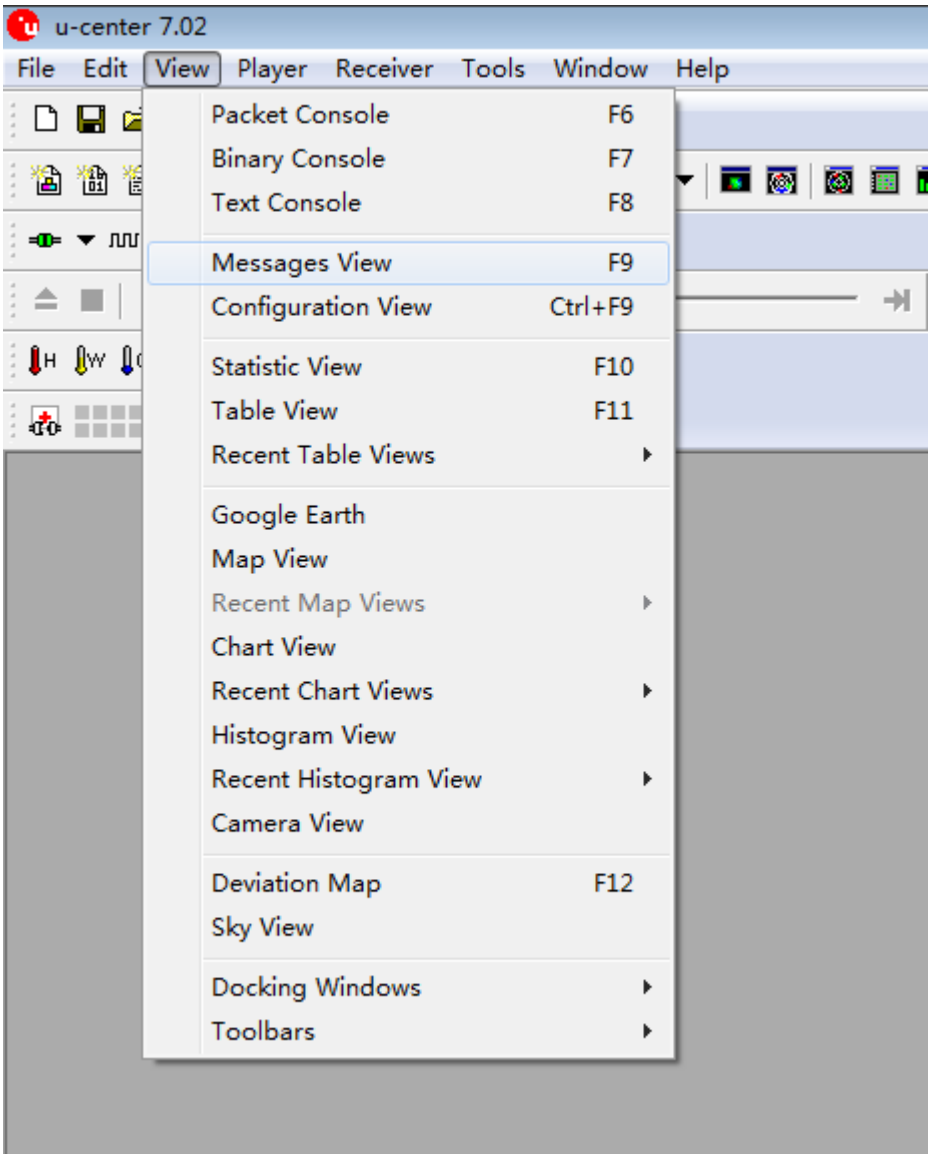
在 Receiver 菜单下，选择 Port，选择实际的串口端口号。Baudrate:38400(默认)。
点击连接按钮，即可连接到 NEO-6M GPS 模块。u-center 将显示各种信息，如下图：



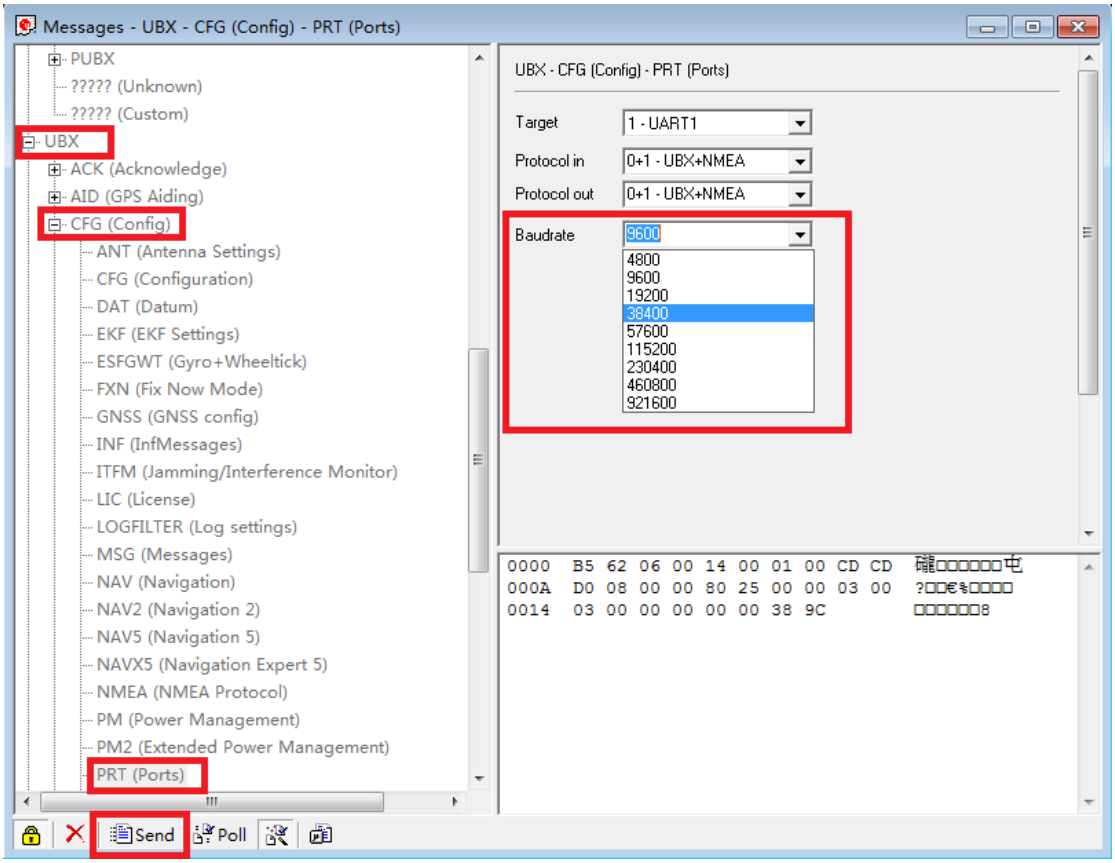
可以安装 GoogleEarthPluginSetup.exe 插件。
安装完成后，在 View 菜单下选择 Google Earth。可以更好的查看所在区域。
注：由于 GPS 存在静态漂移，在 Google Earth 中看到的结果与实际会存在误差。

通过 u-center 修改波特率：

选择 View->Messages View

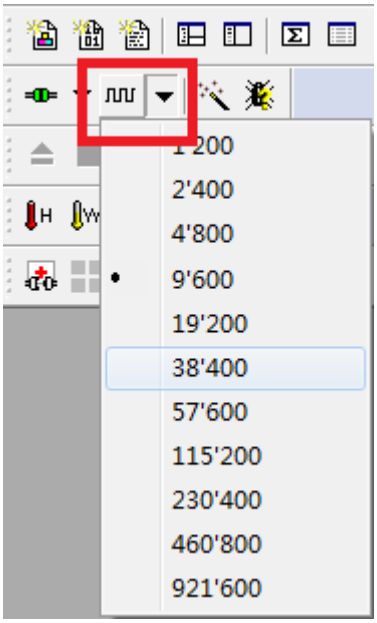


选择 UBX->CFG(Config)->PRT(Ports)，在 Baudrate 选择相应的波特率。点击 Send，既修改好了波特率。此时如果将模块断电，则下一次启动将恢复原来的波特率。如需将配置信息保存到存储器中，则需要进行保存配置参数的相关操作。

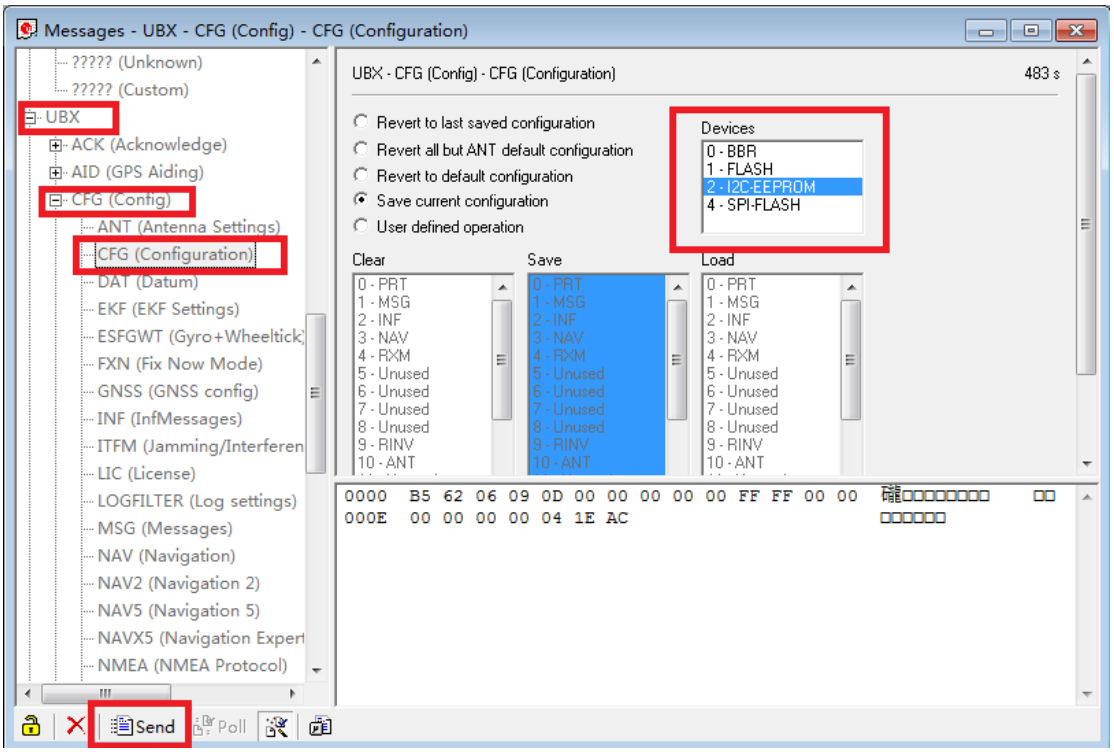


保存配置参数：

在配置好波特率后，需修改相应波特率后方可进行下一步操作。



选择 UBX->CFG(Config)->CFG(Configuration)，在 Devices 中选择 2-I2CEEPROM，点击 Send，既完成保存配置参数。



关于 u-center，更多的使用说明，请查看 software\u-centersetup-7.0.2.1\User_Guide.pdf

在定位后，将收到如下信息：

```
03:07:40  $GPGSV,3,1,10,05,17,097,26,12,08,153,17,14,13,249,26,15,43,026,31*70
03:07:40  $GPGSV,3,2,10,18,39,327,44,21,62,293,42,22,10,305,29,24,71,110,*79
03:07:40  $GPGSV,3,3,10,26,10,045,22,29,16,207,38*7E
03:07:40  $GPGLL,2232.73842,N,11404.58561,E,030740.00,A,A*6D
03:07:41  $GPRMC,030741.00,A,2232.73836,N,11404.58542,E,0.153,,070314,,,A*71
03:07:41  $GPVTG,,T,,M,0.153,N,0.283,K,A*2D
03:07:41  $GPGGA,030741.00,2232.73836,N,11404.58542,E,1,08,1.07,90.9,M,-2.3,M,,*79
03:07:41  $GPGSA,A,3,29,21,18,15,05,14,22,26,,,,,2.02,1.07,1.71*02
03:07:41  $GPGSV,3,1,10,05,17,097,24,12,08,153,15,14,13,249,26,15,43,026,31*70
03:07:41  $GPGSV,3,2,10,18,39,327,44,21,62,293,42,22,10,305,29,24,71,110,*79
03:07:41  $GPGSV,3,3,10,26,10,045,19,29,16,207,39*77
03:07:41  $GPGLL,2232.73836,N,11404.58542,E,030741.00,A,A*6E
03:07:42  $GPRMC,030742.00,A,2232.73830,N,11404.58520,E,0.356,,070314,,,A*77
03:07:42  $GPVTG,,T,,M,0.356,N,0.659,K,A*29
03:07:42  $GPGGA,030742.00,2232.73830,N,11404.58520,E,1,08,1.07,91.0,M,-2.3,M,,*70
03:07:42  $GPGSA,A,3,29,21,18,15,05,14,22,26,,,,,2.02,1.07,1.71*02
03:07:42  $GPGSV,3,1,10,05,17,097,21,12,08,153,13,14,13,249,25,15,43,026,30*71
03:07:42  $GPGSV,3,2,10,18,39,327,44,21,62,293,42,22,10,305,29,24,71,109,*71
03:07:42  $GPGSV,3,3,10,26,10,045,16,29,16,207,39*78
03:07:42  $GPGLL,2232.73830,N,11404.58520,E,030742.00,A,A*6F
03:07:43  $GPRMC,030743.00,A,2232.73824,N,11404.58528,E,0.062,,070314,,,A*7F
03:07:43  $GPVTG,,T,,M,0.062,N,0.114,K,A*23
03:07:43  $GPGGA,030743.00,2232.73824,N,11404.58528,E,1,08,1.01,91.5,M,-2.3,M,,*7F
03:07:43  $GPGSA,A,3,29,21,18,15,05,14,22,26,,,,,1.96,1.01,1.68*02
03:07:43  $GPGSV,3,1,10,05,17,097,18,12,08,153,13,14,13,249,23,15,43,026,29*75
03:07:43  $GPGSV,3,2,10,18,39,327,44,21,62,293,42,22,10,305,28,24,71,109,*70
03:07:43  $GPGSV,3,3,10,26,10,045,13,29,16,207,39*7D
03:07:43  $GPGLL,2232.73824,N,11404.58528,E,030743.00,A,A*63
```

现在以 $GPRMC,030742.00,A,2232.73830,N,11404.58520,E,0.356,,070314,,,A*77 为例进行说明（其它类似），来说明这些信息代表什么。

首先打开 chip PDF\NMEA0183.pdf

找到 GPRMC 的相关说明，如下图：

RMC Recommended Minimum Navigation Information

```
                                                           12
       1          2 3       4 5         6 7   8   9    10  11|
       |          | |       | |         | |   |   |    |   | |
$--RMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,xxxx,x.x,a*hh

 1) Time (UTC)
 2) Status, V = Navigation receiver warning
 3) Latitude
 4) N or S
 5) Longitude
 6) E or W
 7) Speed over ground, knots
 8) Track made good, degrees true
 9) Date, ddmmyy
10) Magnetic Variation, degrees
11) E or W
12) Checksum
```

将 $GPRMC,030742.00,A,2232.73830,N,11404.58520,E,0.356,,070314,,,A*77 与标准协议进行对比，可以知道。

030742.00 为<1>UTC 时间，由于测试地点为东八区，所以实际时间为 03+8=11 时，07 分，42 秒。
A 为<2>定位状态，表示有效定位。
2232.73830 为<3>纬度值。
N 为<4>纬度半球，北半球。
11404.58520 为<5>经度值。
E 为<6>经度半球，东半球。
0.356 为<7>地面速率，0.356 节。（1 节=1 海里/小时，相当于每秒半米。）
<8> 为空。
070314 为<9>UTC 日期，表示 14 年 3 月 7 日。
<10><11> 为空。
A 为<12>模式指示，自主定位。
77 为<13>校验和。通过$与*之间所有字符 ASCII 码的异或运算得到。

在 GPRMC 这条信息中，我们可能并不关心所有数据。
例如：我们只想知道 UTC 时间，则可以用以下方法得到：

```
#include <string.h>

char *p, *q;
char buf[100];      //用于存放结果
unsigned int n;     //用于存放需复制的长度
p = strchr(str, ',');   //找到 str 中第一个出现','的位置
++p;
q = strchr(p, ',');
```

```
n = q - p;
strncpy(buf, p, n);   //将结果复制到目标 buf
buf[n] = '\0';
```

要得到其它信息也可以用类似操作字符串的方法。