

OpenJumper

高性能 GPS 模块

OpenJumper GPS 模块用户手册

目录

1. 特性参数	2
2. 使用说明	3
2.1 模块引脚说明.....	3
2.2 模块使用说明.....	6
2.2.1 NMEA-0183 协议简介	6
2.2.2 模块与单片机连接.....	11
2.2.3 u-center 软件使用简介	11

1. 特性参数

OpenJumper GPS 是一款高性能GPS定位模块。该模块特点包括：

- (1) 模块采用 U-BLOX NEO-6M 模组，体积小，性能优异。
- (2) 模块自带陶瓷天线及 MAXIM 公司 20.5dB 高增益 LNA 芯片，搜星能力强。
- (3) 模块可通过串口进行各种参数设置，并可保存在 EEPROM，使用方便。
- (4) 模块自带 IPX 接口，可以连接各种天线，适应能力强。
- (5) 模块兼容3.3V/5V电平，方便连接各种单片机系统。
- (6) 模块自带可充电后备电池，可以掉电保持星历数据1。

注1：在主电源断开后，后备电池可以维持半小时左右的GPS星历数据的保存，以支持温启动或热启动，实现快速定位。

模块通过串口与外部系统连接，串口波特率支持4800、9600、19200、38400（默认）、57600、115200、230400等不同速率，兼容5V/3.3V单片机系统，可以非常方便的与您的产品进行连接。

该模块各参数如表1.1和表1.2所示：

项目	说明
接口特性	TTL，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	2.5 mCEP (SBAS: 2.0mCEP)
更新速率	最大5Hz
捕获时间	冷启动1: 27S (最快) 温启动: 27S 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、19200、38400 (默认)、57600、 115200、230400
工作温度	-40°C~85°C
模块尺寸	25.5mm*31mm

表1.1 OpenJumper GPS 基本特性

注1：冷启动是指模块所有保存的GPS接收历史信息都丢失了（相当于主电源和后备电池都没电了），这种情况下重启，称之为冷启动。温启动是指模块保存了GPS接收历史信息，但是当前可视卫星的信息和保

存的信息不一致了，这样的条件下重启，称之为温启动。热启动则是指在模块保存了GPS接收历史信息且与当前可视卫星信息一致，这样的条件下重启，称之为热启动。

项目	说明
工作电压 (VCC)	DC2.7V~5.0V
工作电流	45mA1
Voh	VCCX2-0.4V (Min)
Vol	0.4V (Max)
Vih	0.7*VCC (Min)
Vil	0.2*VCC (Max)
TXD/RXD阻抗 ³	510欧

表1.2 OpenJumper GPS电气特性

注1：此电流为连续工作模式下的电流，可以选择省电模式 (Power Save Mode) 以节省用电。

注2：当VCC大于3.3V时VCCX=3.3V，否则VCCX=3.3V。

注3：模块的TXD和RXD脚内部接了510欧电阻，做输出电平兼容处理，所以在使用的时候要注意，导线电阻不可过大(尤其是接USB转TTL串口模块的时候，如果模块的TXD、RXD上带了LED，那就会有问題)，否则可能导致通信不正常。OPENJUMPER GPS GPS模块支持多种通信波特率，通过串口进行设置，并可以保存在模块自带的EEPROM里面，模块默认波特率为：38400 (8位数据位，1位停止位，无奇偶校验)，详细的设置方法，我们会在后面的2.2.3.1节介绍。

2. 使用说明

2.1 模块引脚说明

OpenJumper GPS模块非常小巧 (35.5mm x 25mm)，模块通过5个2.54mm间距的排针与外部连接，在模块的下方有4个安装孔，方便大家安装到自己的设备里面，模块外观如图2.1.1所示：

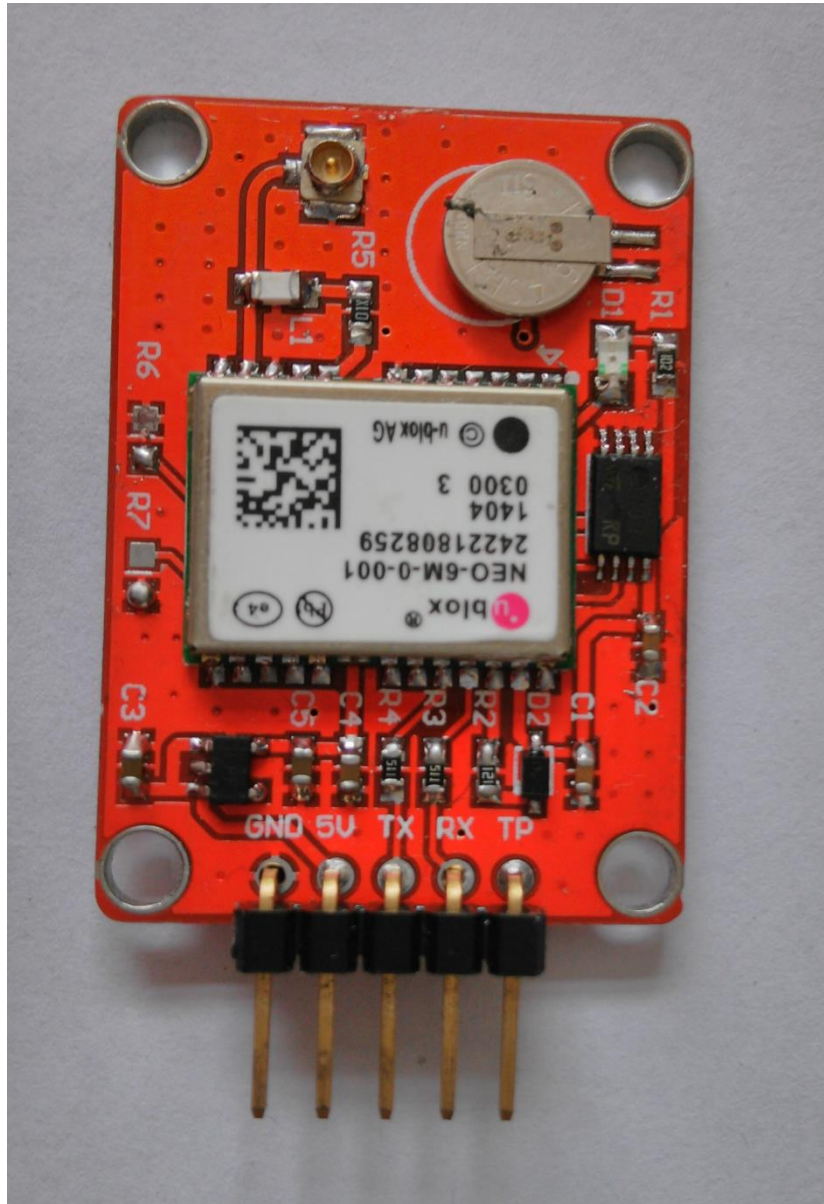


图2. 1. 1 OpenJumper GPS模块外观图

图2. 1. 1中，从右到左，依次为模块引出的PIN1~PIN5脚，各引脚的详细描述如表

2. 1. 1

序号	名称	说明
1	PPS (TP)	时钟脉冲输出脚
2	RXD	模块串口接收脚(TTL电平, 不能直接接RS232电平!), 可接单片机的TXD
3	TXD	模块串口发送脚(TTL电平, 不能直接接RS232电平!), 可接单片机的RXD
4	VCC	电源 (3. 3V~5. 0V)

5	GND	地
---	-----	---

表2.1.1 OpenJumper GPS模块各引脚功能描述

其中，PPS/TP引脚同时连接到了模块自带的状态指示灯：PPS，该引脚连接在UBLOX NEO-6M模组的TIMEPULSE端口，该端口的输出特性可以通过程序设置。PPS指示灯（即PPS/TP引脚），在默认条件下（没经过程序设置），有2个状态：

- 1 常亮，表示模块已开始工作，但还未实现定位。
- 2 闪烁（100ms灭，900ms亮），表示模块已经定位成功。

这样，通过PPS指示灯，我们就可以很方便的判断模块的当前状态，方便大家使用。另外，图2.1.1中，左上角的IPX接口，可以用来外接一个有源天线，从而进一步提高模块的接收能力，通过外接有源天线，我们就可以把模块放到室内，天线放到室外，实现室内定位。一般GPS有源天线都是采用SMA接口，我们需要准备一根IPX(IPEX)转SMA的连接线，从而连接OpenJumper GPS模块与天线

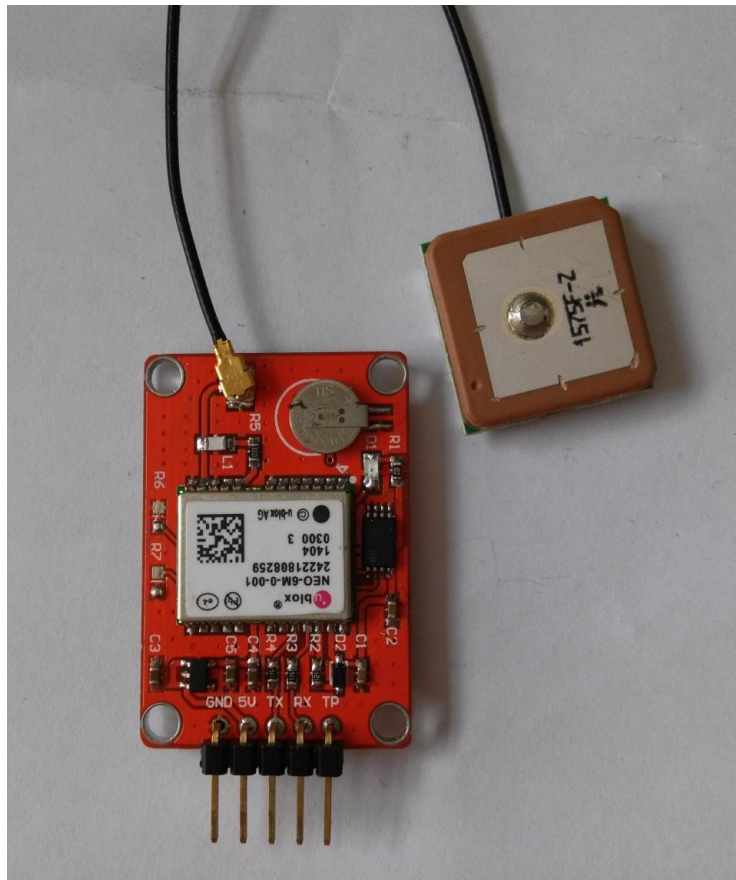


图2.1.2 OpenJumper GPS外接天线

注：图2.1.2中是使用的的IPX的接口天线

2.2 模块使用说明

OpenJumper GPS模块同外部设备的通信接口采用UART（串口）方式，输出的GPS定位数据采用NMEA-0183协议（默认），控制协议为UBX协议（该协议的详细介绍请看u-blox6_ReceiverDescriptionProtocolSpec_GPS.G6-SW-10018-C.pdf这个文档）。

这里，我们将向大家介绍NMEA-0183协议、模块与单片机的连接方法、并结合ublox提供的u-center软件介绍OpenJumper GPS模块的使用。

2.2.1 NMEA-0183协议简介

NMEA 0183是美国国家海洋电子协会（National Marine Electronics Association）为海用电子设备制定的标准格式。目前业已成了GPS导航设备统一的RTCM（Radio Technical Commission for Maritime services）标准协议。

NMEA-0183协议采用ASCII码来传递GPS定位信息，我们称之为帧。

帧格式形如：**\$aacc,ddd,ddd,···,ddd*hh(CR)(LF)**

- (1) “\$”：帧命令起始位
- (2) aacc：地址域，前两位为识别符（aa），后三位为语句名（ccc）
- (3) ddd···ddd：数据
- (4) “*”：校验和前缀（也可以作为语句数据结束的标志）
- (5) hh：校验和（check sum），\$与*之间所有字符ASCII码的校验和（各字节做异或运算，得到校验和后，再转换16进制格式的ASCII字符）
- (6) (CR)(LF)：帧结束，回车和换行符

NMEA-0183常用命令如表2.2.1.1所示：

序号	命令	说明	最大帧长
1	\$GPGGA	GPS定位信息	72
2	\$GPGSA	当前卫星信息	65
3	\$GPGSV	可见卫星信息	210
4	\$GPRMC	推荐定位信息	70
5	\$GPVTG	地面速度信息	34
6	\$GPGLL	大地坐标信息	
7	\$GPZDA	当前时间(UTC1)信息	

表2.2.1.1 NMEA-0183常用命令表

- (3) 正在用于定位的卫星号 (01~32)
- (4) PDOP综合位置精度因子 (0.5-99.9)
- (5) HDOP水平精度因子1 (0.5-99.9)
- (6) VDOP垂直精度因子 (0.5-99.9)

举例如下：

```
$GPGSA, A, 3, 26, 02, 05, 29, 15, 21, , , , , , 2.45, 1.49, 1.94*0E
```

注1：精度因子值越小，则准确度越高。

3, \$GPGSV (可见卫星数, GPS Satellites in View)

\$GPGSV语句的基本格式如下：

```
$GPGSV, (1), (2), (3), (4), (5), (6), (7), ..., (4), (5), (6), (7)*hh (CR) (LF)
```

- 1 GSV语句总数。
- 2 本句GSV的编号。
- 3 可见卫星的总数 (00~12, 前面的0也将被传输)。
- 4 卫星编号 (01~32, 前面的0也将被传输)。
- 5 卫星仰角 (00~90度, 前面的0也将被传输)。
- 6 卫星方位角 (000~359度, 前面的0也将被传输)
- 7 信噪比 (00~99dB, 没有跟踪到卫星时为空)。

注：每条GSV语句最多包括四颗卫星的信息，其他卫星的信息将在下一条\$GPGSV语句中输出。

举例如下：

```
$GPGSV, 3, 1, 12, 02, 39, 117, 25, 04, 02, 127, , 05, 40, 036, 24, 08, 10, 052, *7E
```

```
$GPGSV, 3, 2, 12, 09, 35, 133, , 10, 01, 073, , 15, 72, 240, 22, 18, 05, 274, *7B
```

4, \$GPRMC (推荐定位信息, Recommended Minimum Specific GPS/Transit Data)

\$GPRMC语句的基本格式如下：

```
$GPRMC, (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12)*hh (CR) (LF)
```

- (1) UTC时间, hhmmss (时分秒)

- (2) 定位状态, A=有效定位, V=无效定位
- (3) 纬度ddmm. mmmmm (度分)
- (4) 纬度半球N (北半球) 或S (南半球)
- (5) 经度dddmm. mmmmm (度分)
- (6) 经度半球E (东经) 或W (西经)
- (7) 地面速率 (000.0~999.9节)
- (8) 地面航向 (000.0~359.9度, 以真北方为参考基准)
- (9) UTC日期, ddmmyy (日月年)
- (10) 磁偏角 (000.0~180.0度, 前导位数不足则补0)
- (11) 磁偏角方向, E (东) 或W (西)
- (12) 模式指示 (A=自主定位, D=差分, E=估算, N=数据无效)

举例如下:

```
$GPRMC, 023543.00, A, 2308.28715, N, 11322.09875, E, 0.195, , 240213, ,
, A*78
```

5, \$GPVTG (地面速度信息, Track Made Good and Ground Speed)

\$GPVTG语句的基本格式如下:

```
$GPVTG, (1), T, (2), M, (3), N, (4), K, (5)*hh (CR) (LF)
```

- (1) 以真北为参考基准的地面航向 (000~359度, 前面的0也将被传输)
- (2) 以磁北为参考基准的地面航向 (000~359度, 前面的0也将被传输)
- (3) 地面速率 (000.0~999.9节, 前面的0也将被传输)
- (4) 地面速率 (0000.0~1851.8公里/小时, 前面的0也将被传输)
- (5) 模式指示 (A=自主定位, D=差分, E=估算, N=数据无效)

举例如下:

```
$GPVTG, , T, , M, 0.195, N, 0.361, K, A*2A
```

6, \$GPGLL (定位地理信息, Geographic Position)

\$GPGLL语句的基本格式如下

```
$GPGLL, (1), (2), (3), (4), (5), (6), (7)*hh (CR) (LF)
```

- (1) 纬度ddmm. mmmmm (度分)
- (2) 纬度半球N (北半球) 或S (南半球)

- (3) 经度dddmm. mmmmm (度分)
- (4) 经度半球E (东经) 或W (西经)
- (5) UTC时间: hhmmss (时分秒)
- (6) 定位状态, A=有效定位, V=无效定位
- (7) 模式指示 (A=自主定位, D=差分, E=估算, N=数据无效)

举例如下:

\$GPGLL, 2308. 28715, N, 11322. 09875, E, 023543. 00, A, A*6A

7, \$GPZDA (当前时间信息)

\$GPZDA语句的基本格式如下:

\$GPZDA, (1), (2), (3), (4), (5), (6)*hh (CR) (LF)

- (1) UTC时间: hhmmss (时分秒)
- (2) 日
- (3) 月
- (4) 年
- (5) 本地区域小时 (NEO-6M未用到, 为00)
- (6) 本地区域分钟 (NEO-6M未用到, 为00)

举例如下:

\$GPZDA, 082710. 00, 16, 09, 2002, 00, 00*64

NMEA-0183协议命令帧部分就介绍到这里, 接下来我们看看NMEA-0183协议的校验, 通过前面的介绍, 我们知道每一帧最后都有一个hh的校验和, 该校验和是通过计算\$与*之间所有字符ASCII码的异或运算得到, 将得到的结果以ASCII字符表示就是该校验 (hh)。

例如语句: \$GPZDA, 082710. 00, 16, 09, 2002, 00, 00*64, 校验和 (红色部分参与计算) 计算方法为:

0X47 xor 0X50 xor 0X5A xor 0X44 xor 0X41 xor 0X2C xor 0X30 xor 0X38
xor 0X32 xor 0X37 xor 0X31 xor 0X30 xor 0X2E xor 0X30 xor 0X30 xor 0X2C
xor 0X31 xor 0X36 xor 0X2C xor 0X30 xor 0X39 xor 0X2C xor 0X32 xor 0X30
xor 0X30 xor 0X32 xor 0X2C xor 0X30 xor 0X30 xor 0X2C xor 0X30 xor 0X30

得到的结果就是0X64, 用表示就是64。

NMEA-0183协议我们就介绍到这里，了解了该协议，我们就可以编写单片机代码，解析NMEA-0183数据，从而得到GPS定位的各种信息了。

2.2.2 模块与单片机连接

模块与单片机连接最少只需要4根线即可：VCC、GND、TXD、RXD。其中VCC和GND用于给模块供电，模块TXD和RXD则连接单片机的RXD和TXD即可。本模块兼容5V和3.3V单片机系统，所以可以很方便的连接到你的系统里面去。

OpenJumper GPS模块与单片机系统的典型连接方式如图2.2.2.1所示：

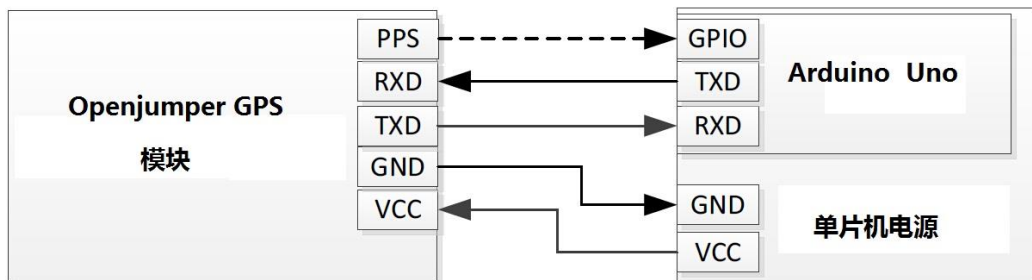


图2.2.2.1 OpenJumper GPS模块与单片机系统连接示意图

图中，PPS/TP与单片机GPIO（通用IO口）的连接不是必须的，大家可以根据自己的需要选择连接还是不连接，这个引脚不影响模块的正常使用。

这里特别注意，模块的TXD和RXD脚是TTL电平，不能直接连接到电脑的RS232串口上，必须经过电平转换芯片(MAX232之类的)，做电平转换后，才能与之连接。

2.2.3 u-center软件使用简介

u-center是由ublox公司提供的GPS评估软件，功能十分强大，可以对我们的OpenJumper GPS模块进行全面的测试，该软件（u-center-7.0.2.1）在我们附赠的资料里面有，大家可以直接安装（注意要联网）。

打开u-center软件，界面如图2.2.3.1所示：

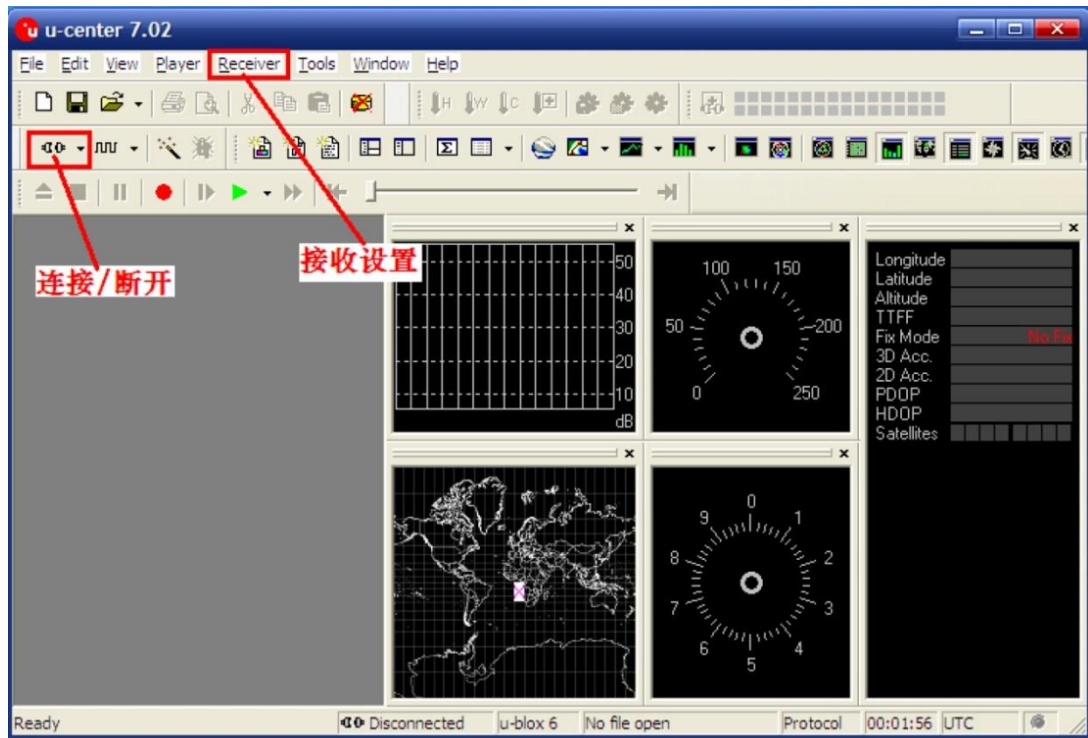


图2. 2. 3. 1 u-center启动界面

然后我们在图中的Receiver菜单里面，选择Port为COM2（**连接GPS模块的串口端口号，请根据您的实际情况选择！**），Baudrate为38400（模块出厂时的波特率是38400，如果您自己改动了波特率，请根据您的改动进行设置！！）。

如果您忘记了自己设置为多少了，那么先给模块断电，然后用一个镊子，将模块上面的EEPROM芯片的SDA和SCL引脚短路（24C32的5脚和6脚），然后重新上电（上电的时候，必须保持SDA和SCL短路2秒左右），上电完成后，模块的波特率就是：9600，然后，设置u-center的Baudrate为9600，此时便可以通过u-center来访问模块了。

再点击图中的连接/断开按钮，即可连接上OpenJumper GPS模块，同时软件开始显示各种信息，如图2. 2. 3. 2所示：

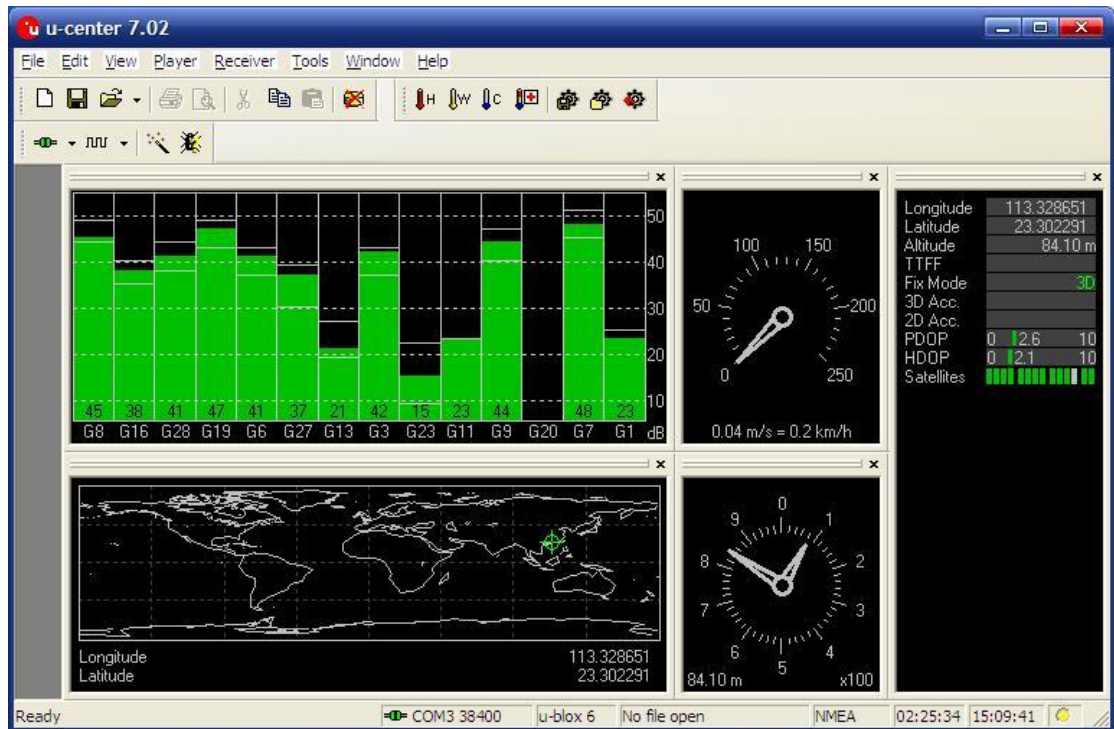


图2. 2. 3. 2 显示各种信息

图中只是显示了5个默认的消息窗口，u-center还提供其他很多窗口视图，比如按下F8，就可以调出Text Console窗口，观看来自GPS模块的原始数据，如图2. 2. 3. 3所示：

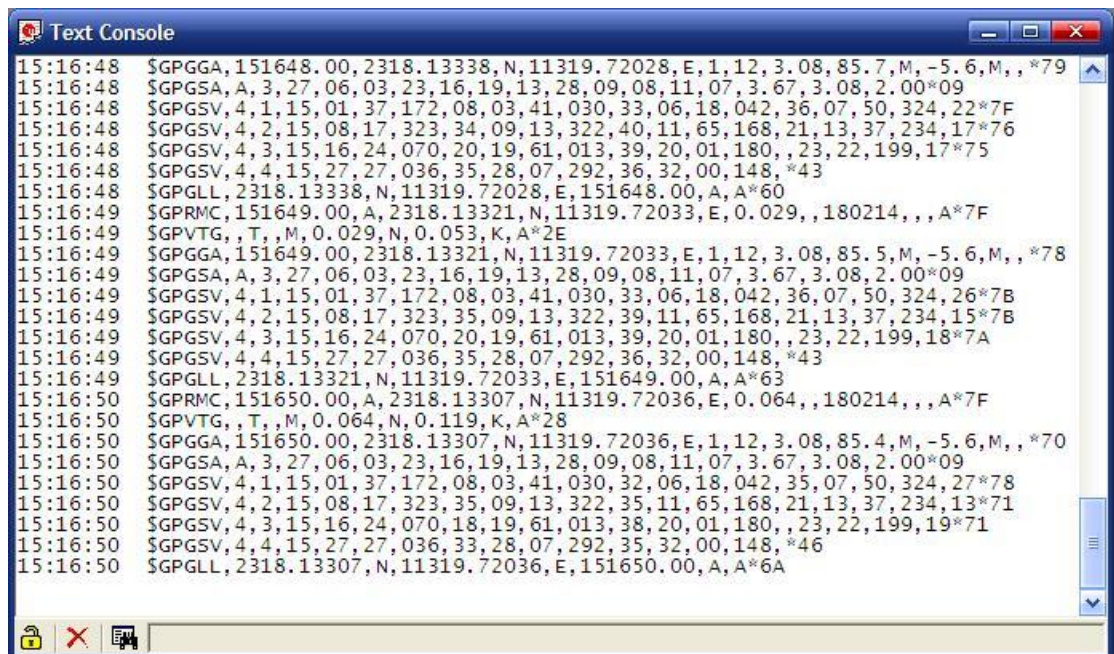


图2. 2. 3. 3 GPS模块返回的NMEA-0183协议数据

其他窗口，在此，我们就不一一介绍了，接下来我们介绍下如何利用u-center设置GPS模块。

2.2.3.1 如何设置波特率

模块默认的波特率是38400，这里，我们通过u-center设置模块波特率为9600，并进行保存。

点击View Messages View，调出该窗口，然后点击UBX CFG(Config) PRT(Ports)，设置Baudrate为9600，如图2.2.3.1.1所示：

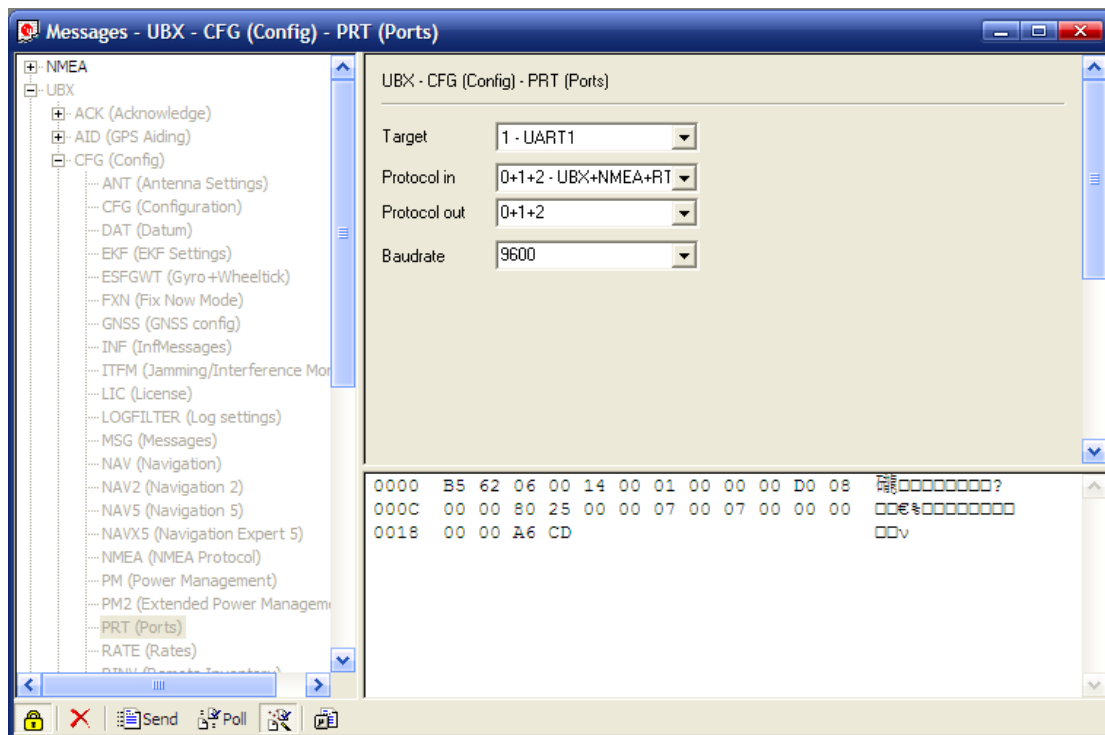


图2.2.3.1.1 设置模块波特率为9600

在配置好了之后，点击窗口左下角的Send按钮，就可以将配置发往OpenJumper GPS模块。这样，我们就设置模块波特率为9600了，此时u-center必须重新设置串口波特率为9600，才可以和模块进行通信，如图2.2.3.1.2所示：

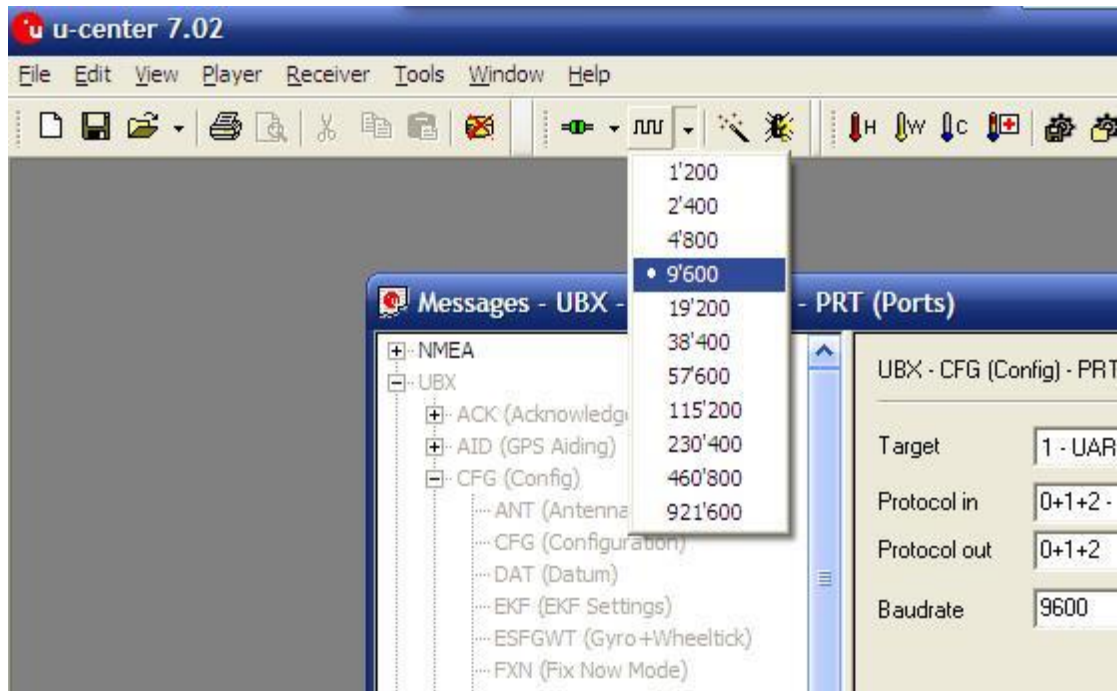


图2. 2. 3. 1. 2 设置u-center软件波特率为9600

设置好之后，u-center即可与模块重新通信，但是模块的波特率数据，并没有保存在EEPROM里面，只要模块一断电，下次上电，波特率就还是38400。

因此我们还需要对刚刚的配置进行一个保存操作，在Messages View窗口，选择UBX CFG(Config) CFG(Configuration)，选择Save current configuration，Device选择：2-I2C-EEPROM，如图2. 2. 3. 1. 3所示：

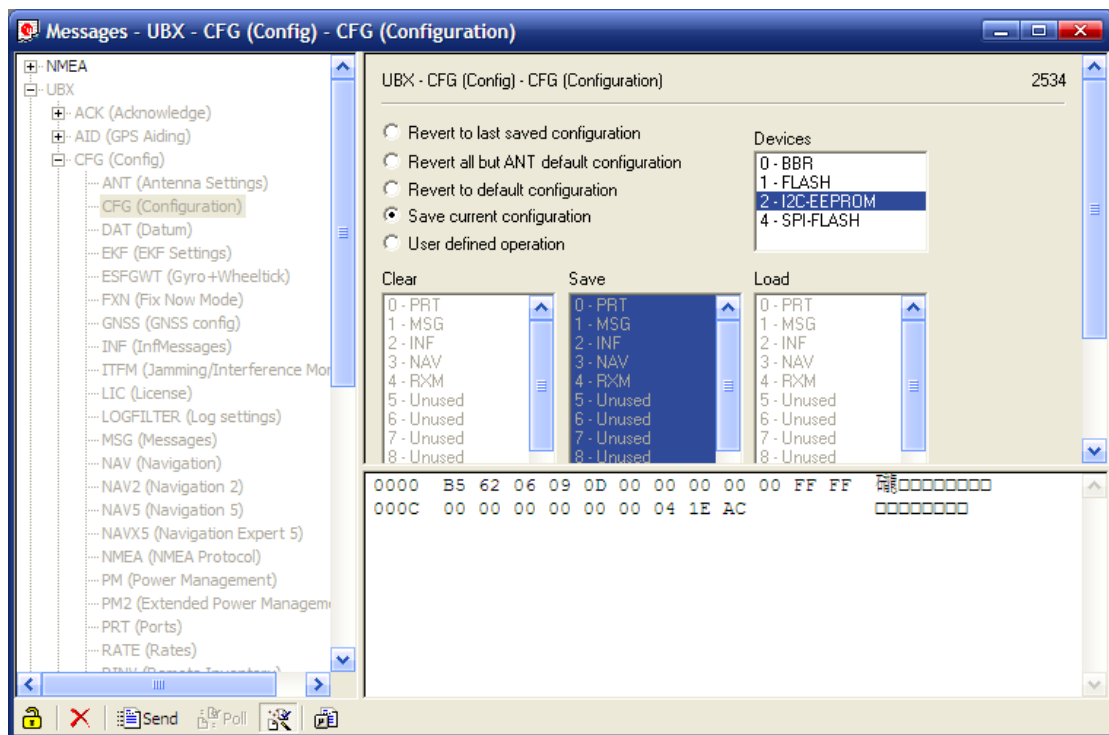


图2. 2. 3. 1. 3 将当前配置保存在EEPROM

在设置好了之后，点击窗口左下角的Send按钮，就可以将指令发往OpenJumper GPS模块，从而将当前的模块配置信息，保存在EEPROM里面。这样，模块的波特率就被固定设置为9600了，断电也可以保存，下次上电就是9600的波特率了。

2. 2. 3. 2 如何设置输出信息

模块默认会输出6种帧数据：GPRMC、GPVTG、GPGGA、GPGSA、GPGSV和GPGGL。如图2. 2. 3. 2. 1所示：

```

08:06:54 $GPGGA,080654.00,2318.21572,N,11319.75058,E,1,06,1.38,188.4,M,-5.6,M,,*4B
08:06:54 $GPGSA,A,3,18,31,25,22,14,12,,,,,3.34,1.38,3.04*01
08:06:54 $GPGSV,3,1,10,12,32,054,46,14,47,348,41,18,50,148,27,21,05,175,*76
08:06:54 $GPGSV,3,2,10,22,84,226,23,24,03,048,31,25,59,099,10,29,03,134,*7D
08:06:54 $GPGSV,3,3,10,31,41,257,19,32,01,320,37*72
08:06:54 $GPGGL,2318.21572,N,11319.75058,E,080654.00,A,A*69
08:06:55 $GPRMC,080655.00,A,2318.21645,N,11319.75062,E,0.241,,110214,,A*7F
08:06:55 $GPVTG,,T,,M,0.241,N,0.446,K,A*22
08:06:55 $GPGGA,080655.00,2318.21645,N,11319.75062,E,1,06,1.38,188.3,M,-5.6,M,,*43
08:06:55 $GPGSA,A,3,18,31,25,22,14,12,,,,,3.34,1.38,3.04*01
08:06:55 $GPGSV,3,1,10,12,32,054,48,14,47,348,42,18,50,148,27,21,05,175,*7B
08:06:55 $GPGSV,3,2,10,22,84,226,24,24,03,048,35,25,59,099,10,29,03,134,*7E
08:06:55 $GPGSV,3,3,10,31,41,257,20,32,01,320,39*76
08:06:55 $GPGGL,2318.21645,N,11319.75062,E,080655.00,A,A*66
08:06:56 $GPRMC,080656.00,A,2318.21706,N,11319.75064,E,0.109,,110214,,A*73
08:06:56 $GPVTG,,T,,M,0.109,N,0.202,K,A*2B
08:06:56 $GPGGA,080656.00,2318.21706,N,11319.75064,E,1,06,1.38,188.2,M,-5.6,M,,*41
08:06:56 $GPGSA,A,3,18,31,25,22,14,12,,,,,3.34,1.38,3.04*01
08:06:56 $GPGSV,3,1,10,12,32,054,48,14,47,348,42,18,50,148,27,21,05,175,*7B
08:06:56 $GPGSV,3,2,10,22,84,226,23,24,03,048,34,25,59,099,09,29,03,134,*70
08:06:56 $GPGSV,3,3,10,31,41,257,18,32,01,320,38*7C
08:06:56 $GPGGL,2318.21706,N,11319.75064,E,080656.00,A,A*65

```

图2. 2. 3. 2. 1 模块默认输出的帧数据

有时候，我们并不需要这么多信息，比如我们只需要GPRMC的信息就够了。这里，我们将教您如何设置模块，使得模块只输出GPRMC定位信息。

在Messages View窗口，选择UBX CFG(Config) MSG(Messages)，在Message的下拉菜单里面，选择：F0-00 NMEA GxGGA，然后UART1的ON勾选框去掉勾选，如图2. 2. 3. 2. 2所示：

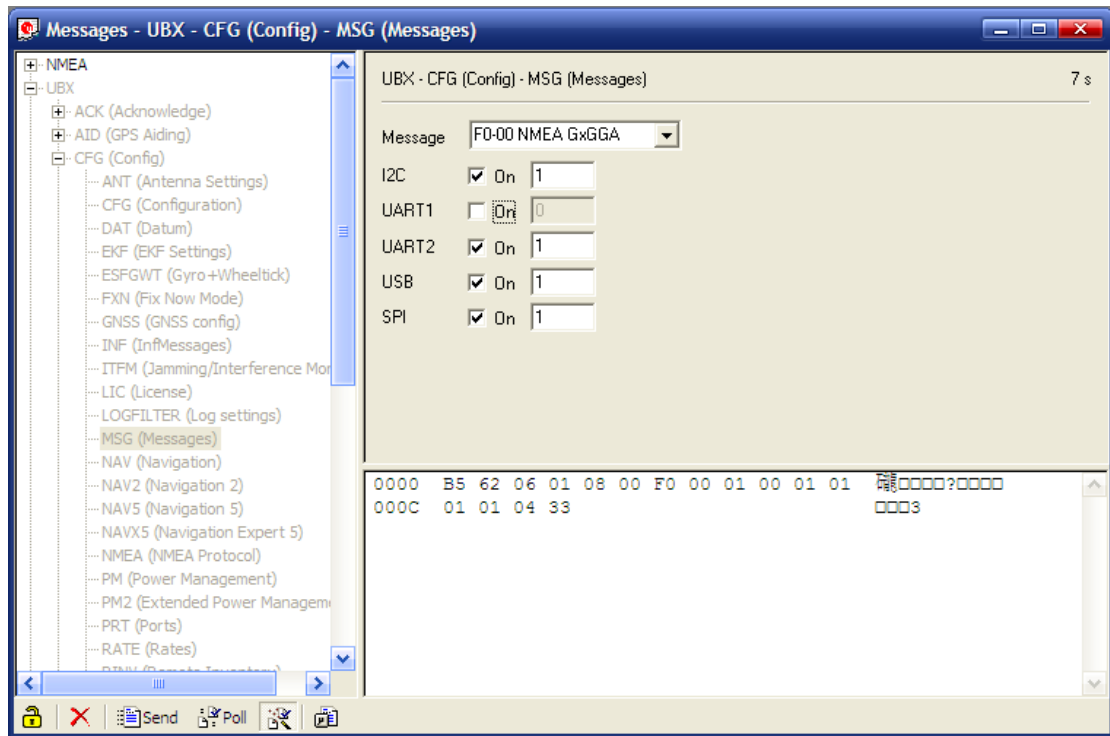


图2. 2. 3. 2. 2 关闭GPGGA输出

如上图的设置，我们将NMEA协议的GPGGA输出（UART1的输出）关闭了，设置好之后，点击右下角的Send按钮，这样模块就不会再输出GPGGA信息了。同样的方法，我们将其他的GPVTG、GPGSA、GPGSV和GPGGL等依次关闭，最后，模块就只会输出GPRMC帧了，如图2. 2. 3. 2. 3所示：

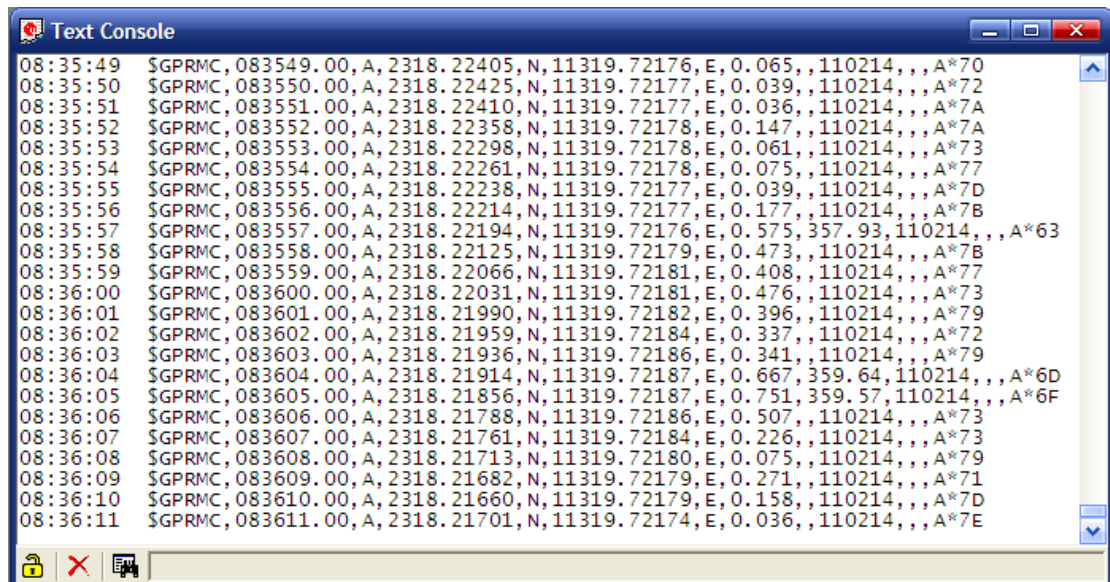


图2. 2. 3. 2. 3 仅输出GPRMC信息

同样，这样的配置，我们如果不保存，下次上电，模块就还是会输出其他帧信息，如果我们要保存，就按照2. 2. 3. 1节所介绍的保存操作方法，保

存一下即可。

2.2.3.3 如何设置测量频率

模块支持最快5Hz的测量频率，也就是1秒钟最快可以输出5次定位信息。这里我们将实现：设置OpenJumper GPS模块的测量频率为5Hz（默认为1Hz），还是在Messages View窗口，点击UBX CFG(Config)RATE(Rates)，设置Measurement Period为200ms，如图2.2.3.3.1所示：

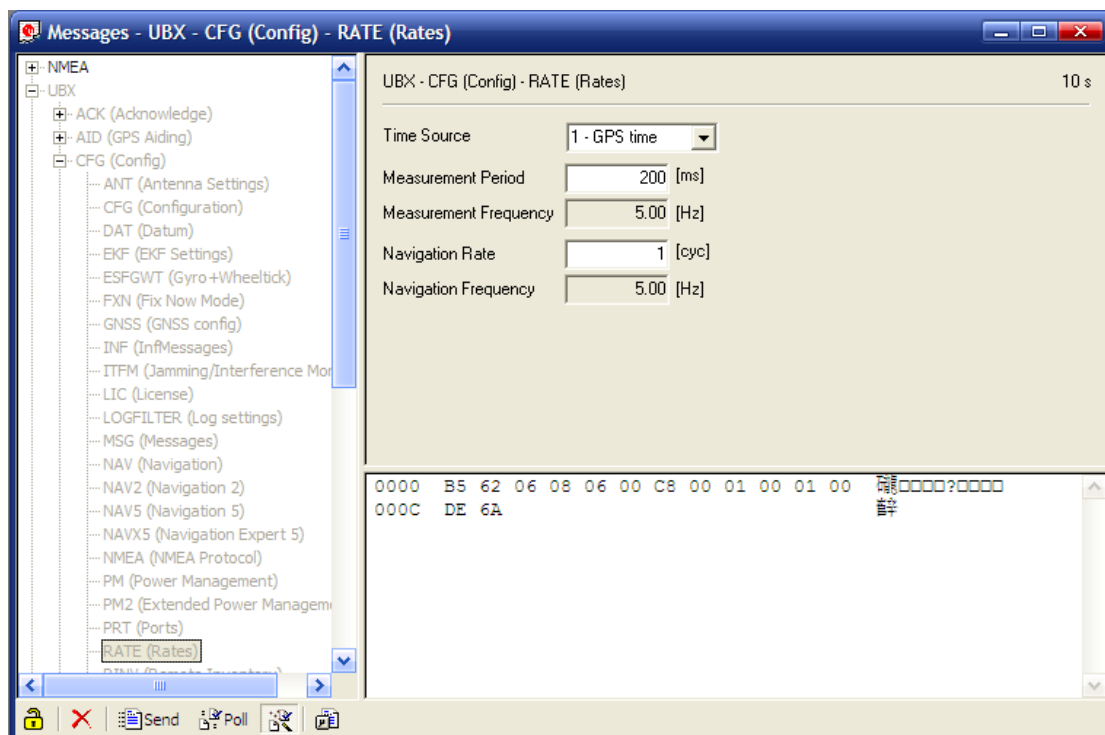


图2.2.3.3.1 设置测量频率为5Hz

在配置好了之后，点击窗口左下角的Send按钮，就可以将配置发往OpenJumper GPS模块。然后，可以看到其他信息窗口的数据更新速度明显变快了，说明我们的设置成功了。

同样，如果要保存该设置，请按照2.2.3.1节所介绍的保存操作方法，进行保存。

2.2.3.4 如何设置时钟脉冲（PPS）

模块的PPS/TP引脚，可以输出时钟脉冲，默认是1Hz，最快可以到1KHz，时钟脉冲的输出频率和占空比都是可以设置的。这里我们将设置模块的PPS/TP引脚输出1Hz的脉冲，且占空比为50%。

在Messages View窗口，点击UBX CFG(Config) TP(Timepulse)，设置

Pulse Lenth为500ms，如图2. 2. 3. 4. 1所示：

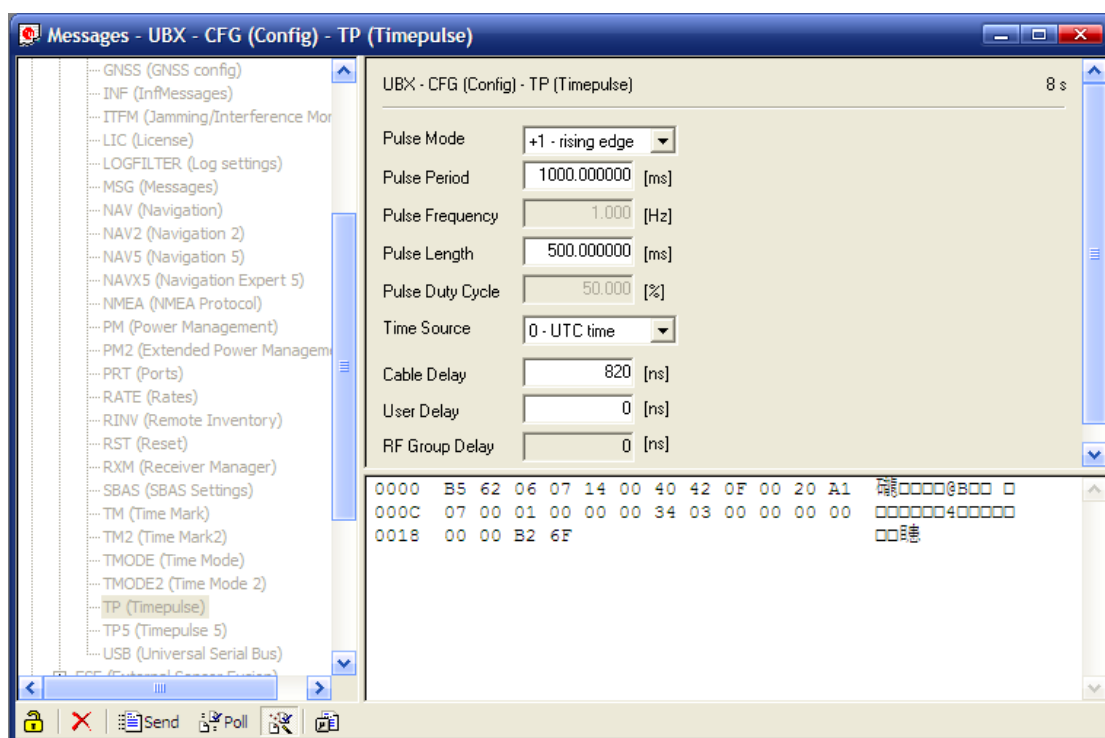


图2. 2. 3. 4. 1 设置时钟脉冲为50%占空比的方波

在配置好之后，点击Send按钮，发送配置到OpenJumper GPS模块。然后，可以看模块PPS信号灯（定位成功后）变成500ms亮，500ms灭的闪烁了。说明我们的设置也成功了。

同样，如果要保存该设置，请按照2. 2. 3. 1节所介绍的保存操作方法，进行保存。