

QuecPython GPS 用户指导

LTE Standard 模块系列

版本：1.0.0

日期：2020-11-09

状态：临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：support@quectel.com。

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。因未能遵守有关操作或设计规范而造成的损害，上海移远通信技术股份有限公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

免责声明

上海移远通信技术股份有限公司尽力确保开发中功能的完整性、准确性、及时性或效用，但不排除上述功能错误或遗漏的可能。除非其他有效协议另有规定，否则上海移远通信技术股份有限公司对开发中功能的使用不做任何暗示或明示的保证。在适用法律允许的最大范围内，上海移远通信技术股份有限公司不对任何因使用开发中功能而遭受的损失或损害承担责任，无论此类损失或损害是否可以预见。

保密义务

除非上海移远通信技术股份有限公司特别授权，否则我司所提供文档和信息的接收方须对接收的文档和信息保密，不得将其用于除本项目的实施与开展以外的任何其他目的。未经上海移远通信技术股份有限公司书面同意，不得获取、使用或向第三方泄露我司所提供的文档和信息。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，上海移远通信技术股份有限公司有权追究法律责任。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2020，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2020.

文档历史

修订记录

版本	日期	作者	变更表述
-	2020-11-09	Kenney/Rivern	文档创建
1.0.0	2020-11-09	Kenney/Rivern	临时版本

目录

文档历史	2
目录	3
图片索引	4
1 引言	5
2 GPS 概述.....	6
3 GPS 模块使用流程.....	7
3.1. 准备工作	7
3.2. 数据处理	9
4 附录	12

图片索引

图 1：连接 GPS 模块与开发板	7
图 2：L80-R GPS 模块功能	8

1 引言

本文档以 EC100Y-CN 模块为例介绍如何使用 GPS 功能，从而帮助用户使用 GPS 模块和天线。

本文档适用以下移远通信模块：

- EC100Y-CN
- EC600S-CN

2 GPS 概述

利用 GPS 定位卫星，在全球范围内实时进行定位、导航的系统，称为全球卫星定位系统，简称 GPS。NEMA-0183，是 GPS 接收机应当遵守的标准协议，也是目前 GPS 接收机上使用最广泛的协议，大多数常见的 GPS 接收机、GPS 数据处理软件、导航软件都遵守或者至少兼容这个协议。

GPS 现已被广泛应用于交通、测绘等许多行业。GPS 的所有应用领域，都是基于定位、或从定位延伸出去的，主要包括：运动导航、轨迹记录、大地测量、周边信息查询等。

3 GPS 模块使用流程

3.1. 准备工作

步骤1: 首先准备 EC100Y-CN 开发板和 L80-R GPS 模块，将 L80-R GPS 模块与开发板相连。注意使用时应处于室外环境以便于接收 GPS 信号。如下图所示：

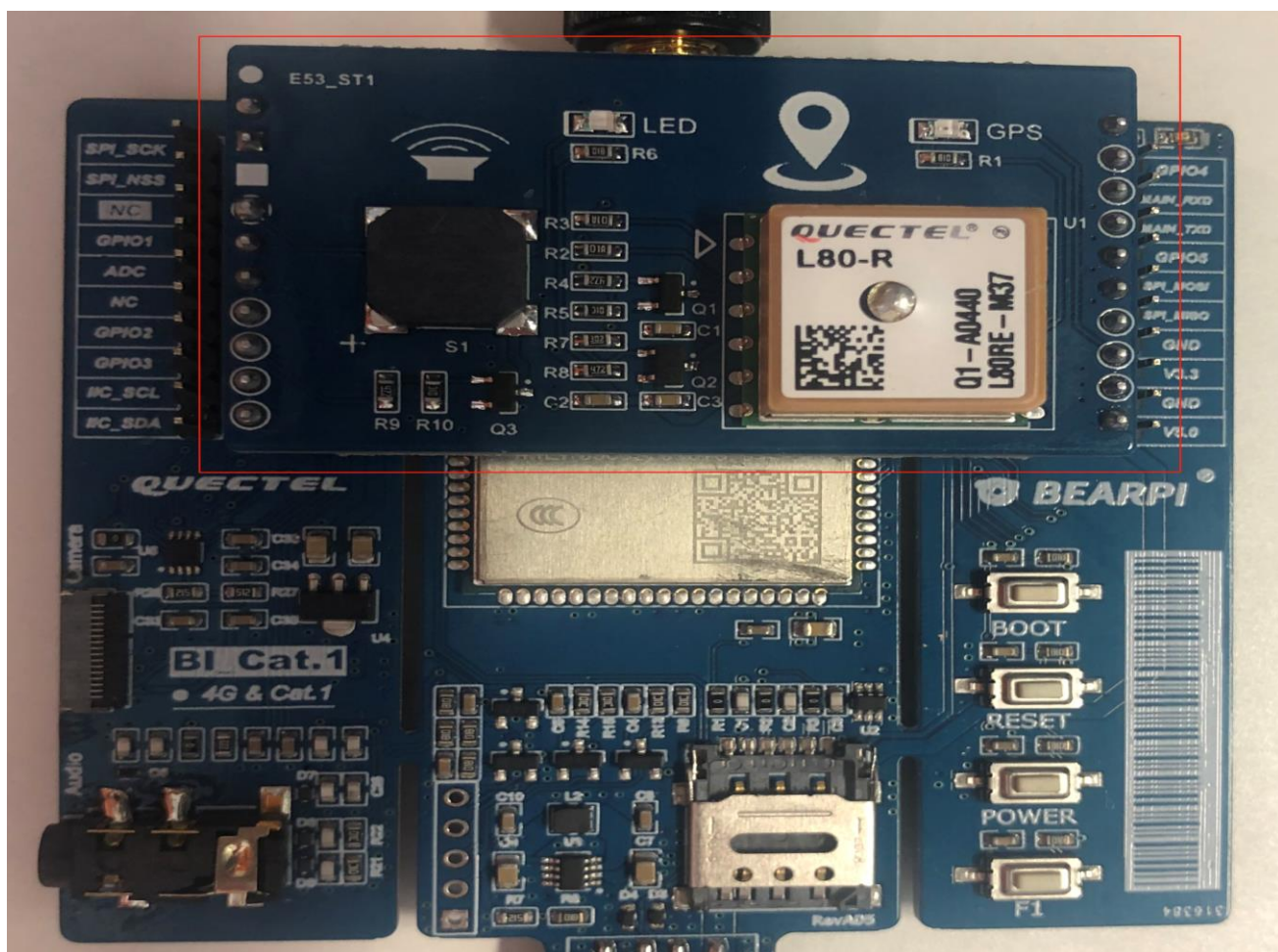


图 1：连接 GPS 模块与开发板

L80-R GPS 模块功能如下图所示：

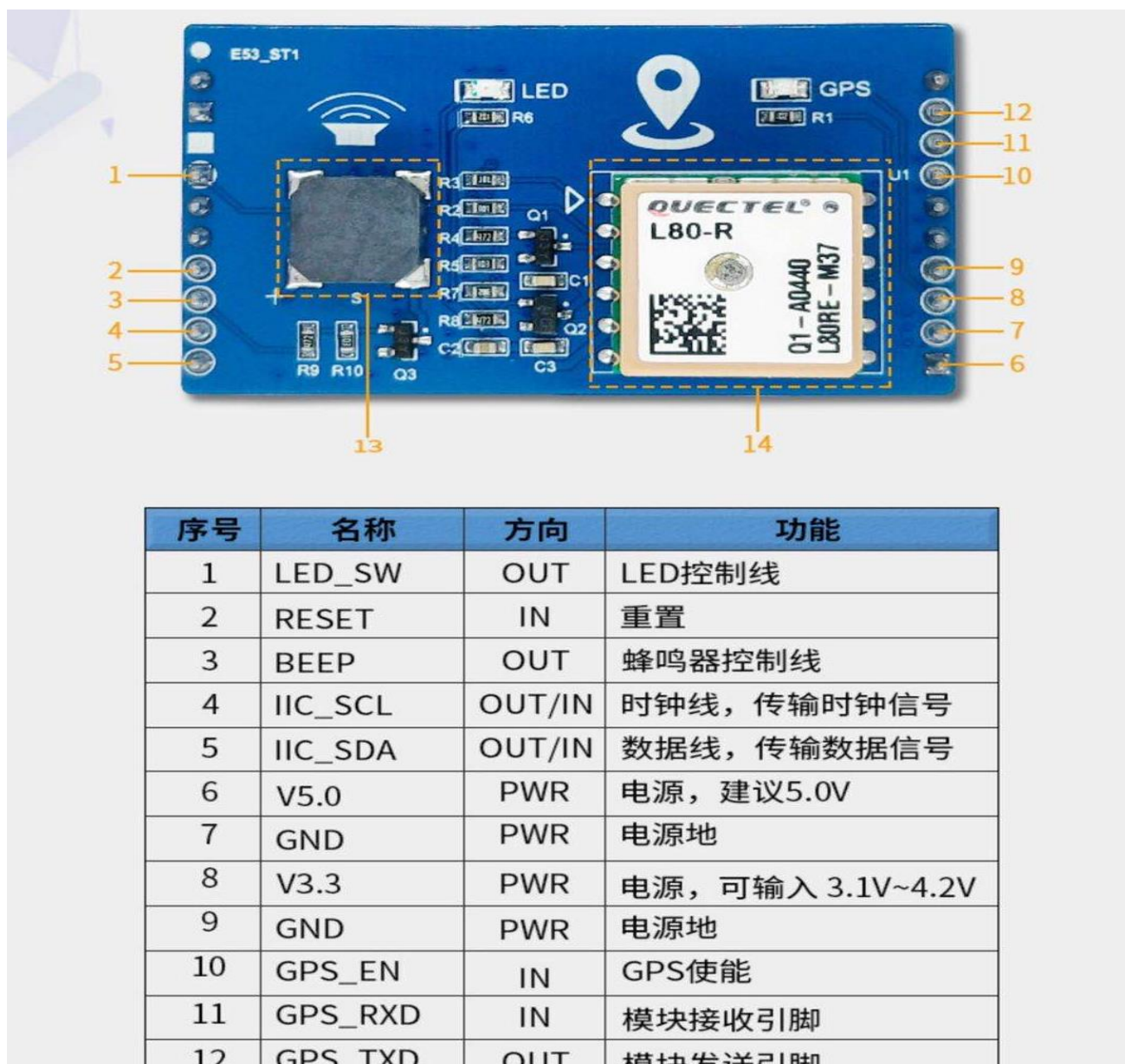


图 2: L80-R GPS 模块功能

步骤2: 通过 xshell 工具连接开发板，进入交互页面，由上文模块功能图可知 `GPS_EN` 引脚对应的开发板上的串口为 `GPIO5`，故通过 `GPIO5` 启动 GPS 功能，执行代码如下：

```
>>> from machine import *
>>>
>>>
>>> gpio5 = Pin(Pin.GPIO5,Pin.OUT,Pin.PULL_DISABLE,1)
>>>
>>> gpio5.write(0)
0
>>>
```

步骤3: 执行完以上代码后，即可看到模块的 GPS 指示灯常亮，同理也可以通过设置 GPIO1 的电压来点亮 LED 灯。

3.2. 数据处理

步骤1: 启用模块的 GPS 功能后，GPS 数据会通过模块的 GPS_TXD 串口发送到开发板中，接下来通过 EC100Y-CN QuecPython 的 machine 模块 UART 串口的数据传输功能将数据输出，代码如下所示：

```
>>>
>>> from machine import UART
>>> uart1 = UART(UART.UART2,9600,8,0,1,0)
>>> uart1.any()
8192
>>> uart1.read(1024)
b'$GPGGA,000102.262,,,,,0,0,,,M,,,*4D\r\n$GPGLL,,,,,000102.262,V,N*7F\r\n$GPGSA,A,1,,,,,,,,,,,,,*1E\r\n$G
00103.262,V,N*7E\r\n$GPGSA,A,1,,,,,,,,,,,,,*1E\r\n$GPGSV,1,1,00*79\r\n$GPRMC,000103.262,V,,,,,0.00,0.00,060
GSV,1,1,00*79\r\n$GPRMC,000104.262,V,,,,,0.00,0.00,060180,,N*41\r\n$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32\r\n$
80,,N*40\r\n$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32\r\n$GPGGA,000106.262,,,,,0,0,,,M,,,*49\r\n$GPGLL,,,,,00010
GGA,00'
```

machine 模块的详细 API 接口说明请参考文档《Quectel QuecPython 类库 API 说明》。

步骤2: 然后，对接收的 NMEA-0183 协议定义的数据格式进行处理，这里需要处理的语句是 \$GNGGA。例如：

```
$GNGGA,092204.999,4250.5589,S,14718.5084,E,1,04,24.4,19.7,M,,,0000*1F
```

字段 0: \$GPGGA, 语句 ID, 表明该语句为 Global Positioning System Fix Data (GPS 定位信息)

字段 1: UTC 时间, hhmmss.sss, 时分秒格式

字段 2: 纬度, ddmm.mmmm, 度分格式 (前导位数不足则补 0)

字段 3: 纬度 N (北纬) 或 S (南纬)

字段 4: 经度, dddmm.mmmm, 度分格式 (前导位数不足则补 0)

字段 5: 经度 E (东经) 或 W (西经)

字段 6: GPS 状态, 0=未定位, 1=非差分定位, 2=差分定位, 3=无效 PPS, 6=正在估算

字段 7: 正在使用的卫星数量 (00~12) (前导位数不足则补 0)

字段 8: HDOP 水平精度因子 (0.5~99.9)

字段 9: 海拔高度 (-9999.9~99999.9)

字段 10: 地球椭球面相对大地水准面的高度

字段 11: 差分时间 (从最近一次接收到差分信号开始的秒数, 如果不是差分定位将为空)

字段 12: 差分站 ID 号 (0000~1023) (前导位数不足则补 0, 如果不是差分定位将为空)

字段 13: 校验值

步骤3: 最后，运行以下代码获取 GPS 信息中的时间和经纬度，通过不断获取时间和经纬度可以实时追踪位置。详细运行方法请参考文档《Quectel QuecPython 基础操作说明》。

```

from machine import UART
import utime
import modem
import _thread
uart = UART(2,115200,8,0,1,0)
def gngga():
    while True:
        #获取当前 RTC 时间
        time = utime.localtime( )
        #获取设备 IMET
        imei = modem.getDevImei( )
        if uart.any() > 0:
            buf = uart.read(uart.any())
            buf = str(buf,"utf8" )
            try :
                gngga1 = buf.split("$GNGGA,")[1].split("\r\n" )[0].split(",")
                # UTC 时间， hhmmss.sss,时分秒格式
                time_gps = gngga1[0]
                #纬度 ddm. mmm,度分格式前导位数不足则补 0
                _latitude = float(gngga1[1])
                #经度 dddmm. mmm,度分格式前导位数不足则补 0
                _longitude = float(gngga1[3])
                # UTC 时间转化
                _Clock = int(time[0:2])
                _Minute = time[2:4]
                _Second = time[4:6]
                _Clock = _Clock + 8
                #防止超过 24 小时
                if ( _Clock >= 24):
                    _Clock = _Clock % 24
                #最终获得时间
                Effect_time = str(_clock) + ':' + _Minute + ':' + _Second
                #最终获得纬度
                Effect_latitude = int(_latitude / 100)+ ((_latitude % 100) / 60)
                #最终获得经度
                Effect_longitude = int(_longitude / 100) + ((_longitude % 100)/ 60)
                print( '当前时间:',time)
                print( 'GPS 时间:',Effect_time)
                print('设备 IMET',imei)
                print(gngga1[2],"",str(Effect_latitude ))
                print(gngga1[4],"",str(Effect_longitude ))

            utime.sleep(2)

```

```
        except:
            print('数据格式有误或数据受损')
            continue
def run():
    _thread.start_new_thread(gngga, ())
run()
```

4 附录

表 1：术语缩写

缩写	英文全称	中文全称
API	Application Programming Interface	应用程序编程接口
GPS	Global Positioning System	全球定位系统
GPIO	General-Purpose Input/Output	通用型输入/输出
GGA	Global Positioning System Fix Data	全球定位系统定位数据
HDOP	Horizontal Dilution of Precision	水平精度因子
LED	Light Emitting Diode	发光二极管