QuecPython GPS User Guide

LTE Standard Module Series

Version: 1.0.0

Date: 2020-11-09

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to support@quectel.com.

General Notes

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

Disclaimer

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

Duty of Confidentiality

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

Copyright

About the Document

Revision History

Version	Date	Author	Description
-	2020-11-09	Kenney/Rivern	Creation of the document
1.0.0	2020-11-09	Kenney/Rivern	Preliminary

Contents

Figure Index

1 Introduction

This document takes EC100Y-CN as an example to introduce how to use the GPS module and antenna with GPS function.

This document is applicable to the following Quectel modules:

- EC100Y-CN
- EC600S-CN

2 GPS Overview

The system that uses GPS positioning satellites to perform positioning and navigation on a global scale in real time is called Global Satellite Positioning System, or GPS. NEMA-0183 is a standard protocol that GPS receivers should comply with, and it is also the most widely used protocol on GPS receivers. Most common GPS receivers, GPS data processing software, and navigation software all comply with or are at least compatible with this protocol.

GPS has been widely used in many industries such as transportation, surveying and mapping. All the application fields of GPS are based on positioning, mainly including: motion navigation, track recording, geodetic survey, surrounding information query, etc.

3 GPS Module Usage Process

3.1. Preparation

Step1: First, prepare the EC100Y-CN EVB and the L80-R GPS module, and connect the L80-R GPS module to the EVB. It should be in an outdoor environment to facilitate receiving GPS signals. As shown below:

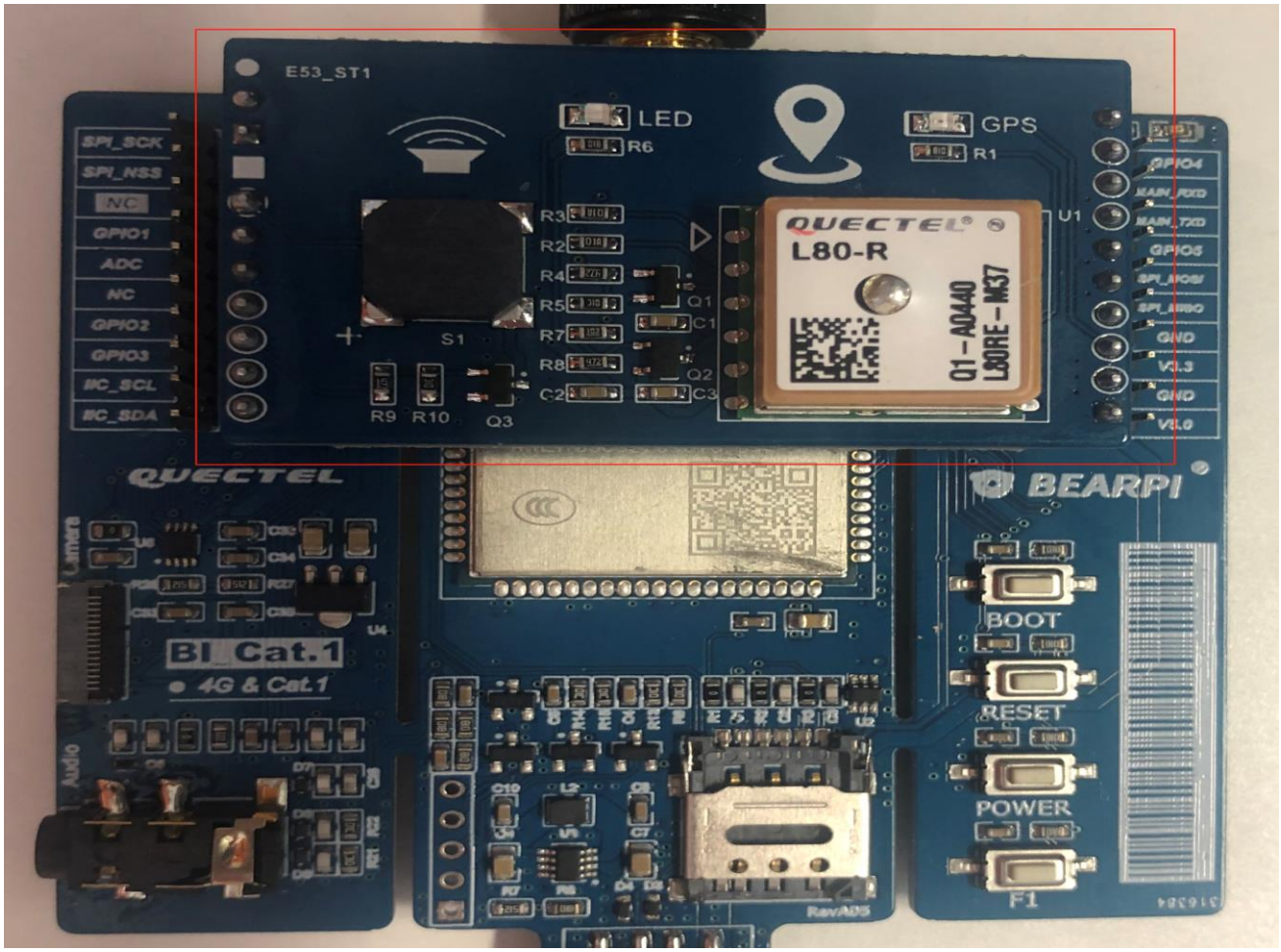
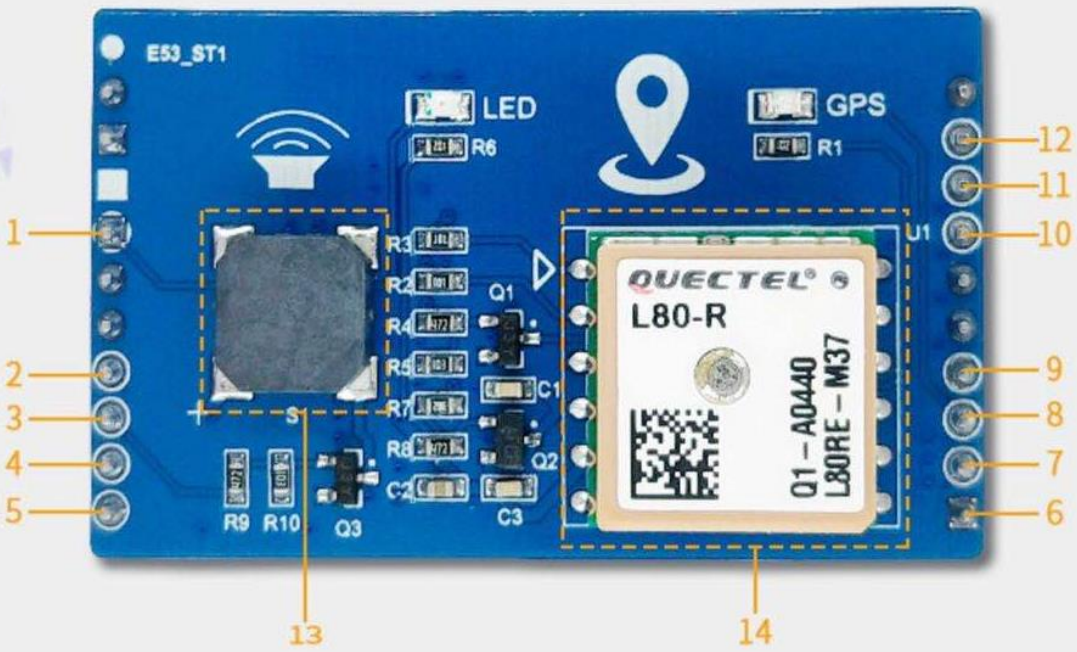


Figure 1: Connect GPS Module to EVB

The functions of the L80-R GPS module are shown below:



Number	Name	Direction	Function
1	LED_SW	OUT	LED control line
2	RESET	IN	Reset
3	BEEP	OUT	Buzzer control line
4	IIC_SCL	OUT/IN	Clock line, transfer clock signal
5	IIC_SDA	OUT/IN	Data line, transfer data signal
6	V5.0	PWR	Power supply,recommended 5.0 V
7	GND	PWR	Power ground
8	V3.3	PWR	Power supply, can input 3.1 V-4.2 V
9	GND	PWR	Power ground
10	GPS_EN	IN	GPS enable
11	GPS_RXD	IN	Module receiving pin
12	GPS_TXD	OUT	Module sending pin

Figure 2: L80-R GPS Module Function

Step2： Connect the EVB through xshell tool and enter the interactive page. From the module function above, it can be seen that the serial port on the EVB corresponding to the *GPS_EN* pin is GPIO5, so the GPS function is enabled via GPIO5. The execution code is as follows:

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

Step3： After executing the above code, you can see that the GPS indicator light of the module is always on. Similarly, you can also set the voltage of GPIO1 to light up the LED.

3.2. Data Processing

Step1： After the GPS function of the module is enabled, the GPS data is sent to the EVB via the GPS_TXD serial port of the module, and the data is output via the data transmission function of the UART serial port of EC100Y-CN QuecPython machine module. The code is shown in the figure below:

```
>>>
>>> from machine import UART
>>> uart1 = UART(UART.UART2,9600,8,0,1,0)
>>> uart1.any()
8192
>>> uart1.read(1024)
b'$GPGGA,000102.262,,,,,0,0,,,M,,M,,*4D\r\n$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$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32\r\n$G
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,,M,,*49\r\n$GPGLL,,,,,0001
GGA,00'
>>>
```

For detailed API interface description of machine module, please refer to *Quectel QuecPython Class Library API Description*.

Step2： Then, process the data format defined by the received NMEA-0183 protocol. The sentence to be processed is $GNGGA. For example:

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

Field 0: $GPGGA, sentence ID, indicating that the sentence is Global Positioning System Fix Data
Field 1: UTC time, hhmmss.sss, in hour, minute and second format
Field 2: Latitude, ddmm.mmmm, in degrees and minutes format (if the leading digit is insufficient, add 0)

Field 3: Latitude N (north latitude) or S (south latitude)
Field 4: Longitude, dddmm.mmmm, in degrees and minutes format (if the leading digit is insufficient, add 0)
Field 5: Longitude E (East longitude) or W (West longitude)
Field 6: GPS status, 0 indicates not positioned, 1 indicates non-differential positioning, 2 indicates differential positioning, 3 indicates invalid PPS, 6 indicates under estimation
Field 7: The number of satellites in use (00-12) (If the leading digit is insufficient, add 0)
Field 8: HDOP level precision factor (0.5-99.9)
Field 9: Altitude (-9999.9-99999.9)
Field 10: The height of the earth ellipsoid relative to the geoid
Field 11: Differential time (the seconds number since the last received differential signal, if it is not differential positioning, it is blank)
Field 12: Differential station ID number (0000-1023) (If the leading digit is insufficient, add 0, if it is not differential positioning, it is blank)
Field 13: Check value

Step3： Finally, run the following code to obtain the time, latitude and longitude in the GPS information. By continuously obtaining the time, latitude and longitude, the location can be tracked in real time. For detailed operation methods, please refer to *Quectel_QuecPython_Basic Operation Instructions*.

```
from machine import UART
import utime
import modem
import _thread
uart = UART(2,115200,8,0,1,0)
def gngga():
    while True:
        #Get current RTC time
      time = utime.localtime( )
      #Get device 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 time, hhmmss.sss, in hour, minute and second format
          time_gps = gngga1[0]
          #Latitude, ddmm.mmmm, in degrees and minutes format (if the leading digit is insufficient, add 0)
          _latitude = float(gngga1[1])
          #Longitude, dddmm.mmmm, in degrees and minutes format (if the leading digit is insufficient, add 0)
```

```
                _longitude = float(gngga1[3])
                #UTC time conversion
                _Clock = int(time[0:2])
                _Minute = time[2:4]
                _Second = time[4:6]
                _Clock =_Clock + 8
                #Prevent more than 24 hours
                    if (_Clock >= 24):
                        _Clock = _Clock % 24
                    #Finally get the time
                    Effect_time = str(_clock) + ':' +_Minute + ':' +_Second
                    #Finally get the latitude
                    Effect_latitude = int(_latitude / 100)+ ((_latitude % 100) / 60)
                    #Finally get the longitude
                    Effect_longitude = int(_longitude / 100) +((_longitude % 100)/ 60)
                    print( 'current time:',time)
                    print( 'GPS time:',Effect_time)
                    print('Equipment IMET ',imei)
                    print(gngga1[2],'',str(Effect_latitude ))
                    print(gngga1[4],'',str(Effect_longitude ))

                    utime.sleep(2)
                except:
                    print('Incorrect data format or data corruption')
                    continue
def run():
    _thread.start_new_thread(gngga, ())

run()
```

4 Terms and Abbreviations

Table 1: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
GPS	Global Positioning System
GPIO	General-Purpose Input/Output
GGA	Global Positioning System Fix Data
HDOP	Horizontal Dilution of Precision
LED	Light Emitting Diode