**SIM7080G Cat-M/NB-IoT HAT**

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **SIM7080G Cat-M/NB-IoT HAT** **Compatible with Raspberry Pi 2B/3B/Zero/Zero W** | |  | |
|  |
|  |
|  |
|  |

Introduction

NB-IoT/Cat-M(eMTC)/GNSS HAT for Raspberry Pi, based on SIM7080G, globally applicable.

|  |
| --- |
| [More](http://www.waveshare.com/SIM7080G-Cat-M-NB-IoT-HAT.htm) |

Features

* Standard Raspberry Pi 40PIN GPIO extension header, supports Raspberry Pi series boards.
* Supports communication protocols such as TCP/UDP/HTTP/HTTPS/TLS/DTLS/PING/LWM2M/COAP/MQTT.
* Supports GNSS positioning (GPS, GLONASS, BeiDou, and Galileo).
* Onboard USB interface, to test AT Commands, get GPS positioning data, and so on.
* Breakout UART control pins, to connect with host boards like Arduino/STM32.
* Onboard voltage translator, 3.3V by default, allows being switched to 5V via onboard jumper.
* SIM card slot, supports ONLY 1.8V SIM card (3V SIM card is not available).
* 3 x LED indicators, easy to monitor the working status.
* Baudrate: 300~3686400 bps.
* Common baudrate auto-negotiation: 9600/19200/38400/57600/115200 bps.

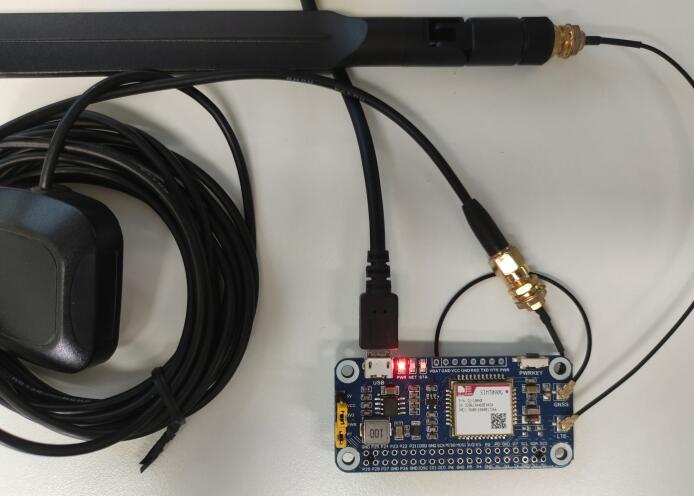
User Guide

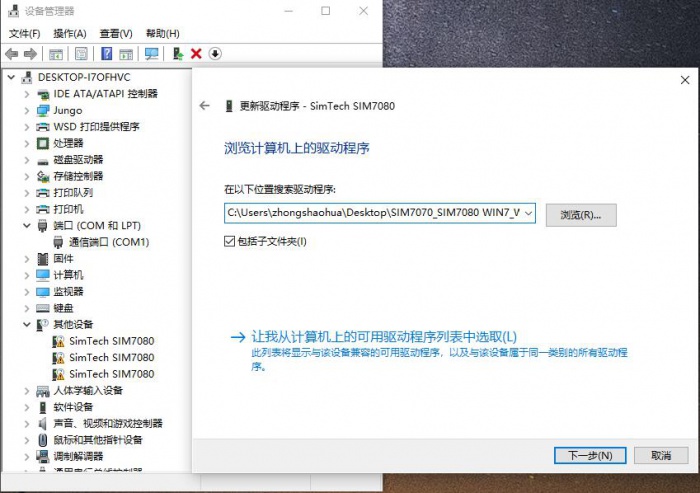
Testing it in Windows PC

**1. Hardware connection and driver installation**

* Before testing, you should prepare
  + A Micro USB cable
  + LTE and GPS Antennas
  + The SIM7080G Cat-M/NB-IoT HAT
  + An NB-IoT SIM card.
* Insert the SIM card into the SIM7080 board, connect the LTE antenna and the GPS antenna. Connect the micro USB of SIM7080G board to PC by USB cable.
* Hold the PWRKEY button for about 1s to power on the module. After powering on, the STA indicator lights on, NET indicator quick-blinks firstly and then slow-blinks.
* Download the [driver file](https://www.waveshare.com/w/upload/2/2f/SIM7070_SIM7080_WIN7_WIN10_USB_Driver_V1.00.rar), unzip it. Then update the driver by Devices Manager
* Generally, the SIM7080G will be recognized by PC with three USB ports. To update the driver, you can right-click the USB port and update it with the driver downloaded
* AT Port COM31 is the one used as the communication port. We will use it to test the SIM7080G board.

Note: The SIM7080G modules support global networks. The time to log in the network is different among different areas. And to use the SIM7080G, you should first confirm that LTE-M or LTE-NB network are available in the local area.

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_006.jpg)

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_002.jpg)

**2. Test network**

For more information about the AT commands, you can refer to [SIM7080 Series\_AT Command Manual\_V1.01](http://cn.simcom.com/upload/file/1574403025.pdf)

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_004.jpg)

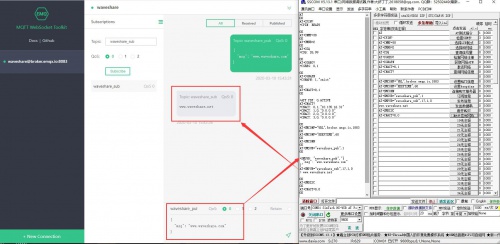
Test network

|  |  |  |
| --- | --- | --- |
| **Command** | **Descriotion** | **Return** |
| AT | Test command | OK |
| AT+CPIN? | Check SIM card | OK |
| AT+CNMP=38 | Choose LTE mode | OK |
| AT+CMNB=2 | Choose NB network | OK |
| AT+CSQ | Check signal intensity | OK |
| AT+CGREG? | Check netwok status | OK |
| AT+CGNAPN | Query APN | OK |
| AT+CPSI? | Request UE system information | OK |
| AT+CNACT=0,1 | activate application network | OK |
| AT+CNACT? | Check network IP | OK |
| AT+CNCAT=0,0 | Close application network | OK |

**3. Test MQTT**

For more information about the MQTT, please refer to [SIM7080 Series\_MQTT(S)\_Application Note\_V1.00](http://cn.simcom.com/upload/file/1568960374.pdf)

Please test Web by [clicking here](https://www.emqx.io/mqtt/mqtt-websocket-toolkit). Please check the TCP port and domain name when testing, for details, please[refer to here](https://www.emqx.io/mqtt/public-mqtt5-broker)

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_005.jpg)

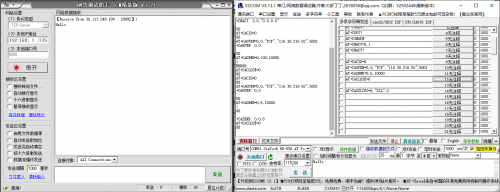
Test MQTT

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Return** |
| AT+SHCONF | Setting HTTPS(S) | OK |
| AT+SHSSL | Select SSL setting | OK |
| AT+SHBOD | Set Body | OK |
| AT+SHAHEAD | Add Head | OK |
| AT+SHPARA | Set HTTP(S) Para | OK |
| AT+SHCPARA | Clear HTTP(S) Para | OK |
| AT+SHCHEAD | Clear Head | OK |
| AT+SHSTATE | Query HTTP(S) connecting status | OK |
| AT+SHREQ | Set respond type | OK |
| AT+SHREAD | Queru respond data | OK |
| AT+SHDISC | Disconnect HTTP(S) | OK |

**3. Test TCP**

For more information abou tthe TCP, you can refer to[Series\_TCPUDP(S)\_Application Note\_V1.00](http://cn.simcom.com/upload/file/1572402939.pdf)

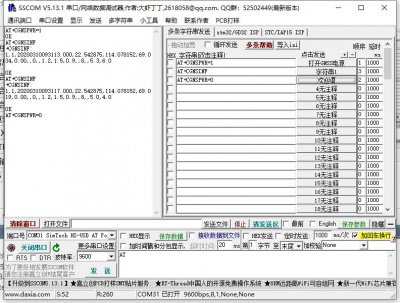
Note that to test TCP you need to map LAN IP and router firstly.

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_008.png)

Test TCP

|  |  |  |
| --- | --- | --- |
| **Command** | **Description** | **Return** |
| AT+CACID | Set TCP/UDP index | OK |
| AT+CASSLCFG | Set SSL parameters | OK |
| AT+CAOPEN | Open a TCP/UDP connection | OK |
| AT+CASEND | Send data | OK |
| AT+CARECV | Receive data | OK |
| AT+CACLOSE | Cose TCP/UDP connection | OK |
| AT+CACFG | Set transparent parameters | OK |
| AT+CASWITCH | Enable transparent mode | Ok |

**4. Test GNSS**

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_009.jpg)

Test GNSS

For more information, please refer to [SIM7080 Series\_GNSS\_Application Note\_V1.00](http://cn.simcom.com/upload/file/1573629551.pdf)

When testing GNSS, you should set the GPS antenna in the open area. You can also use u-center software for locating.

|  |  |  |
| --- | --- | --- |
| **Commands** | **Description** | **Return** |
| AT+CGNSPWR | Open/close GNSS power | OK |
| AT+CGNSINF | GNSS information | OK |
| AT+CGNSCOLD | GNSS cold starting | OK |
| AT+CGNSWRAM | GNSS warm starting | OK |
| AT+CGNSHOT | GNSS hot starting | OK |
| AT+CGNSMOD | GNSS workign mode setting | OK |
| AT+CGNSXTRA | Open GNSS XTRA | OK |
| AT+CGNSCPY | Copyt GNSS XTRA file | OK |
| AT+SGNSCFG | GNSS NMEA output setting | OK |
| AT+SGNSCMD | AT port output GNSS NMEA data | OK |

Use it with Raspberry Pi

**Initialize IO and install software**

The PWRKEY pin is connected to P7 (wiringPi number) of Raspberry Pi by jumper. To let the SIM7080G board work normally, you should firstly initialize P7 pin. Download codes and run by the following commands.

sudo apt-get update

sudo pip install RPi.GPIO

sudo apt-get install python-serial

wget -P ~/Documents/ https://www.waveshare.com/w/upload/e/e6/SIM7080G\_Cat\_M\_NB\_IoT\_HAT\_Code.tar.gz

cd ~/Documents

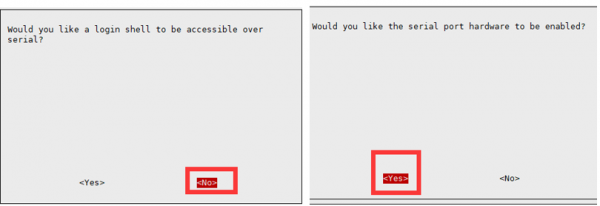
tar -xzf SIM7080G\_Cat\_M\_NB\_IoT\_HAT\_Code.tar.gz

sh ~/Documents/SIM7080G\_Cat\_M\_NB\_IoT\_HAT\_Code/RaspberryPi/pi\_gpio\_init.sh

**Enbale UART and hardware connection**

Open raspi-config by following commands, and choose Interfacing Options -> Serial, close login sheel and open the hardware serial port.

sudo rapsi-config

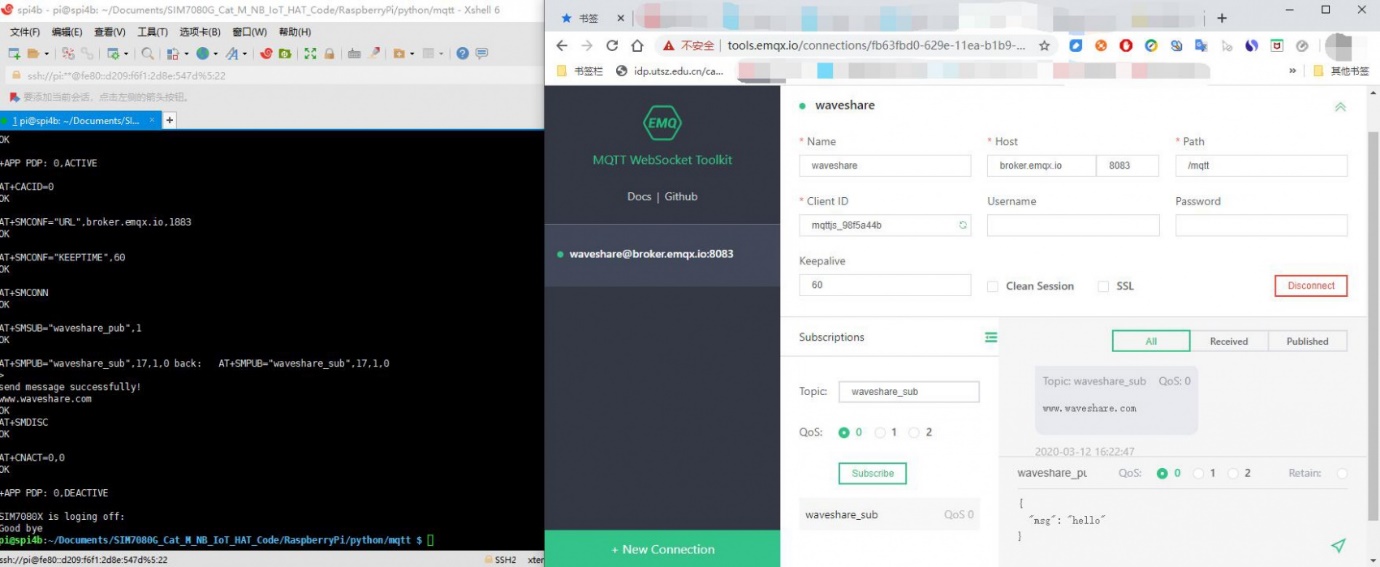
[](https://www.waveshare.com/wiki/File:L76X_GPS_Module_rpi_serial.png)[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_011.jpg)

**Python codes**

The python codes is provided for MQTT function

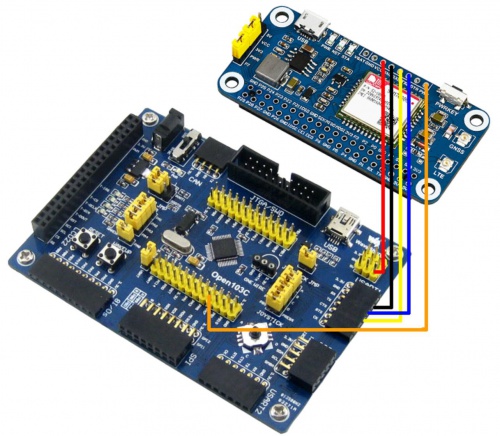
cd ~/Documents/SIM7080G\_Cat\_M\_NB\_IoT\_HAT\_Demo\_Code/python/mqtt

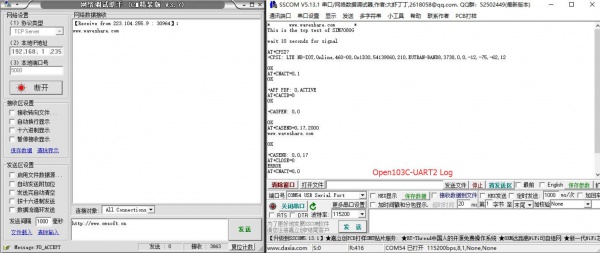
sudo python mqtt.py

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_012.jpg)

Use it with STM32

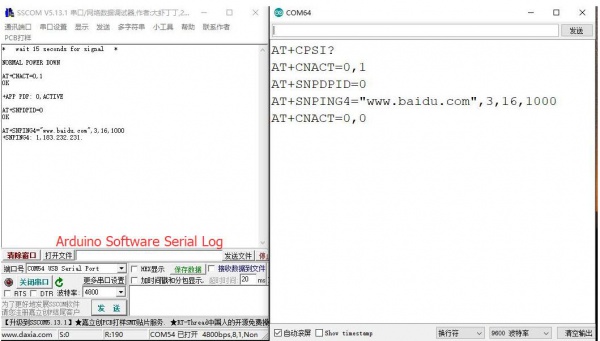
The demo codes provided is based on Waveshare Open103C, it is use for TCP connecting.

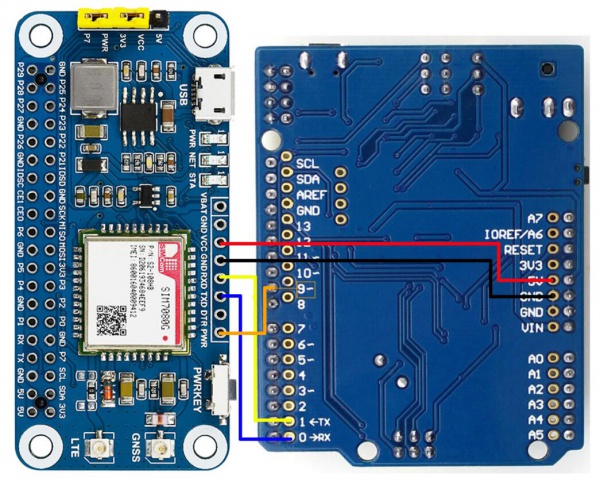
[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_010.jpg)

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_013.jpg)

Use it with Arduino

The demo codes provide is based on Arduino UNO. The D10 (RX) and D11 (TX) is used for software serial.

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_015.jpg)

[](https://www.waveshare.com/wiki/File:SIM7080G_Cat-M_NB-IoT_HAT_016.jpg)