Overview

This Raspberry Pi HAT features multi communication functionalities: NB-IoT, eMTC, EDGE, GPRS, and GNSS.

The NB-IoT (NarrowBand-Internet of Things) and eMTC (enhanced Machine Type Communication) are rising IoT communication technologies evolved from LTE (4G), with advantages include low power, low cost, wide coverage, etc. They are suited for applications such as intelligent instruments, remote controlling, asset tracking, remote monitoring, E-health, mobile POS terminals, sharing bikes, and so on. While the GSM/GPRS, and EDGE are traditional 2G/2.5G technologies capable of sending SMS or making other wireless communications.

Therefore, the SIM7000E NB-IoT HAT would be an ideal choice for either evaluating new rising technologies, or simply communicating/positioning via multiple ways.

Features

- Raspberry Pi connectivity, compatible with Raspberry Pi Zero/Zero W/Zero WH/2B/3B/3B+
- Supports TCP, UDP, PPP, HTTP, FTP, MQTT, SMS, Mail, etc.
- 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 to be switched to 5V via 0Ω resistor
- SIM card slot, compatible with both normal SIM card and NB-IoT specific card
- 2x LED indicators, easy to monitor the working status
- Baudrate: 300bps~3686400bps
- Control via AT commands (3GPP TS 27.007, 27.005, and SIMCOM enhanced AT Commands)
- Supports SIM application toolkit: SAT Class 3, GSM 11.14 Release 98, USAT
- Comes with development resources and manual (examples for Raspberry /Arduino/STM32)
## Contents

Overview...........................................................................................................................................1

Features...................................................................................................................................................1

1. Hardware configuration .....................................................................................................................4
   1.1. Hardware configuration .............................................................................................................4

2. GPRS Debugging ..................................................................................................................................6
   2.1. General AT commands .............................................................................................................6
   2.2. Local virtual severs settings ....................................................................................................7
   2.3. Searching WAN IP ....................................................................................................................7
   2.4. GPRS Setting ............................................................................................................................8
   2.5. Sending data ...............................................................................................................................8
   2.6. Receiving data ...........................................................................................................................9
   2.7. Deactivating Connection ...........................................................................................................9

3. NB-IoT Debugging .............................................................................................................................9
   3.1. General AT commands .............................................................................................................9
   3.2. Local virtual severs settings ....................................................................................................错误！未定义书签。
   3.3. Searching WAN IP ....................................................................................................................错误！未定义书签。
   3.4. NB-IoT Setting ..........................................................................................................................10
   3.5. Sending data ...............................................................................................................................11
   3.6. Receiving data ...........................................................................................................................11
   3.7. Deactivating Connection ...........................................................................................................11

4. GNSS Debugging ...............................................................................................................................12
1. **Hardware configuration**

1.1. **Hardware configuration**

This module comes with GSM antenna, GPS antenna and micro USB cable. Besides these you should prepare a sim card or a nb-iot card:

1) Insert the SIM card to the card slot and connect the GSM antenna.
2) Connect the USB interface of SIM7000E NB-IoT HAT to PC with a micro USB cable. Then the PWR indicator will keep bright.

*Figure : Hardware connection*
3) Generally, the NET indicator will fast flash firstly (1 time per second), which means that the module has not logged in the Network. After logging in, the indicator become to flash slowly (1 time every three seconds). Up to the local GSM network, this process that logging in will last several seconds to dozens of seconds.

If you take too much time to log in and failed, please check that whether the GSM antenna is connected correctly, and whether the SIM card is usable and insert correctly.

4) Install SIM7000 driver (windows driver: www.waveshare.com/wiki/File:SIM7X00-Driver.7z)
Open Device Manager to get the corresponding COM port number of SIM7000. For example, the AT Port is COM19 as below. Users need to choose the correct port according to the Manager.

Figure: Devices Manager

1.2. Solder joints
2. GPRS Debugging

2.1. General AT commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CGATT?</td>
<td>Check the state of GPRS attachment</td>
<td>+CGATT:1</td>
</tr>
</tbody>
</table>
### 2.2. Local virtual servers settings

Virtual servers define the mapping between service ports of WAN and web servers of LAN. All requests from Internet to service ports of WAN will be redirected to the computer (web servers of LAN) specified by the server IP. (see your router’s guide manual)

1) Log in Management Console of your router with browser (read your router’s guide manual for specific address)
2) Set Port: 1822 (The Port can’t be conflict to other’s. Here we set 1822)
3) Set LAN IP address of your computer (you can run CMD on your computer, and execute command `ipconfig` to enquiry the address of IPv4), 192.168.6.168 as examples

<table>
<thead>
<tr>
<th>SIM7000E Test</th>
<th><code>wan1_pp</code></th>
<th>TCP/UDP</th>
<th>1822-1822</th>
<th>1822-1822</th>
<th>192.168.6.168</th>
</tr>
</thead>
</table>

### 2.3. Searching WAN IP

You can search “IP” on browser to get your WAN IP address.
2.4. GPRS Setting

1) AT+CSQ //To enquiry the quality of signal. The first parameter of response is signal quality (Max is 31). The signal stronger, the value bigger.
2) AT+CREG? //Check Network registration. If the second parameter of response is 1 or 5, it means that Network has been registered successfully
3) AT+CGATT? //Check the state of GPRS attachment
4) AT+CSTT="CMNET" //Set the Network according to actual situation. Here we use CMNET
5) AT+CICR //Bring up wireless connection with GPRS
6) AT+CIFSR //Get the local IP address
7) AT+CIPSTART="TCP","113.81.232.178",1822 //Established TCP/IP connection

2.5. Sending data

1) AT+CIPSEND=33 //Send fixed length data
2) AT+CIPSEND // Send changeable length data
3) After getting the response >, edit the contents of message (has been converted) without Enter at the end. Then send 1A in HEX format as below

4) If the data sent successfully, the server will receive the data.

### 2.6. Receiving data

1) Choose the IP address of module on peers input box

2) Input the data which you want to send: SIM7000 Test Successfully

3) Click Send button, you can see that module receive the data with COM assistant software

### 2.7. Deactivating Connection

1) Send AT+CIPCLOSE or AT+CIPSHUT to deactivate connection.

### 3. NB-IoT Debugging

#### 3.1. General AT commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CGATT?</td>
<td>Check the state of GPRS attachment</td>
<td>+CGATT:1\n1 Attached</td>
</tr>
<tr>
<td>AT+CPSI?</td>
<td>AT+CPSI?: Inquiring UE information</td>
<td>+CPSI: OK</td>
</tr>
<tr>
<td>AT+CGNAPN</td>
<td>Check available APN</td>
<td>+CGNAPN: OK</td>
</tr>
<tr>
<td>AT+CSTT</td>
<td>AT+CSTT =”cmnet”: Set APN to CMNET</td>
<td>OK</td>
</tr>
<tr>
<td>AT+CIICR</td>
<td>Bring up wireless connection with GPRS</td>
<td>OK</td>
</tr>
<tr>
<td>AT+CIFSR</td>
<td>Get local IP address</td>
<td>OK</td>
</tr>
<tr>
<td>AT+CIPSTART</td>
<td>AT+CIPSTART=&quot;Mode&quot;, &quot;IP_Addr&quot;, &quot;Port&quot;</td>
<td>CONNECT OK</td>
</tr>
<tr>
<td></td>
<td>Mode: connection type;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IP_Add: Remote server IP address;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Port: Remote server port</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Send data</td>
<td>OK</td>
</tr>
</tbody>
</table>
### 3.2. NB-IoT Setting

1) **AT+CSQ**  //To enquiry the quality of signal. The first parameter of response is signal quality (Max is 31). The signal stronger, the value bigger.

2) **AT+CREG?**  //Check Network registration. If the second parameter of response is 1 or 5, it means that Network has been registered successfully

3) **AT+CGATT?**  //Check the state of GPRS attachment

4) **AT+CPSI?**  //Inquiring UE system information

5) **AT+CGNAPN**  //Get network APN in CAT-M or NB-IOT

6) **AT+CSTT="T"**  //Set the Network according to actual situation. Here we use T

7) **AT+CICR**  //Bring up wireless connection with GPRS

8) **AT+CIFSR**  //Get the local IP address

9) **AT+CIPSTART="TCP","118.190.93.84",2317**  //Established TCP/IP connection
3.3. Sending data

5) AT+CIPSEND=33  //Send fixed length data
6) AT+CIPSEND    // Send changeable length data
7) After getting the response >, edit the contents of message (has been converted) without Enter at the end. Then send 1A in HEX format as below
8) If the data sent successfully, the server will receive the data.

3.4. Receiving data

4) Choose the IP address of module on peers input box
5) Input the data which you want to send: SIM7000 Test Successfully
6) Click Send button, you can see that module receive the data with COM assistant software

3.5. Deactivating Connection

Send AT+CIPCLOSE or AT+CIPSHUT to deactivate connection.
4. GNSS Debugging

4.1. General AT commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+CGNSPWR</td>
<td>GNSS Power Control:</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>AT+CGNSPWR =1: Turn on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AT+CGNSPWR =0: Turn off</td>
<td></td>
</tr>
<tr>
<td>AT+CGNSTST</td>
<td>AT+CGNSTST=1: Send data received to AT Port</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>AT+CGNSTST=0: Stop sending data received to AT Port</td>
<td></td>
</tr>
<tr>
<td>AT+CGNSINF</td>
<td>GNSS navigation information parsed from NMEA sentences</td>
<td>+CGNSINF: OK</td>
</tr>
</tbody>
</table>

4.2. GPS Debugging

1) Connecting the GPS antenna, and place the receiver on open area outdoor
2) AT+CGNSPWR=1  // Turn on power of GPS
3) AT+CGNSTST=1  // Start to sending data received to AT Port
   Open u-center and set the Port and Baudrate(AT Port, COM25).
   Of course, you can use another port (NMEA Port, COM27)
   - SimTech HS-USB AT Port 9001 (COM25)
   - SimTech HS-USB Audio 9001 (COM24)
   - SimTech HS-USB Diagnostics 9001 (COM28)
   - SimTech HS-USB NMEA 9001 (COM27)
4) AT+CGNSTST=0  // Stop sending data received to UART
5) AT+CGNSINF   // Print the GPS information
6) AT+CGNSPWR=0  // Turn off power of GPS
5. **Using with Raspberry Pi**

5.1. **Interface overview**

The default relationship between SIM7000 control pins and Raspberry Pi IOs is shown in Table 1.

**Table 1: The relationship between SIM7000 control pins and Raspberry Pi IOs**

<table>
<thead>
<tr>
<th>SIM7000</th>
<th>IO of Raspberry Pi B+</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>5V</td>
<td>Power supply (5V)</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>TXD</td>
<td>P15 / RXD</td>
<td>UART pin</td>
</tr>
<tr>
<td>RXD</td>
<td>P14 / TXD</td>
<td>UART pin</td>
</tr>
<tr>
<td>PWR</td>
<td>P4 (Need to modify the back solder joint)</td>
<td>Power up the module</td>
</tr>
</tbody>
</table>

5.2. **UART configuration of Raspberry Pi**

Because UART of Raspberry Pi is used for Linux console output by default, if we want to use the UART, we need to change the settings. Executing this command to enter the configuration page:

```
sudo raspi-config
```
Choose Interfacing Options -> Serial -> no->yes, to disable Linux’s use of console UART.

Open /boot/config.txt file, find the below statement and uncomment it to enable the UART. You can directly append it at the end of file as well.

```
enable_uart=1
```

Then reboot.

### 5.3. Minicom for UART debugging on Raspberry Pi

Inserting the module to Raspberry Pi and plug the jumper B,

Install minicom. minicom is a text-based modem control and terminal emulation program for Linux:

```
sudo apt-get install minicom
```

Execute command: `minicom -D /dev/ttyS0` (ttyS0 is the UART of Raspberry Pi 3B)

Baud rate is 115200 by default. If you need to change the baud rate, for example 9600, you can add the parameter `-b 9600`.

The user UART device of Raspberry Pi 2B/Zero is ttyAMA0, and ttyS0 of Raspberry Pi 3B

Testing Bluetooth function as examples:
5.4. Examples

Download the demo code from wiki and copy to the Raspberry Pi (for example, `/home/pi/SIM7000X`)

Enter the bcm2835 directory, compile and install the BCM2835 library:

```
chmod +x configure && ./configure && sudo make && sudo make install
```

5.4.1. AT_Test

```
 cd example/AT_Test && sudo ./AT_Test
```
5.4.2. GPS_Positioning

```bash
cd example/GPS_Positioning && sudo ./GPS_Positioning
```
5.4.3. NB-IoT (TCP_IP)

```
cd example/NB-IoT && sudo ./TCP_IP
```

5.4.4. For more demo code, please visit the website wiki.