

# SIM800C shield-V1.2 for Raspberry Pi 2 user manual V1.0

---Designed by UNV in CHINA

## Revision History

REV.	Description	Date
V1.0	Initial version	2016-04-10

## 一、SIM800C module Features

- Quad-band 850/900/1800/1900MHz
- GPRS multi-slot class 12/10
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Class 4 (2 W @ 850/900 MHz)
- Class 1 (1 W @ 1800/1900 MHz)
- Control via AT commands• SIM application toolkit
- Supply voltage range: GPRS: 3.4 ~ 4.4 V
- Operation temperature : -40 °C to +85 °C

### SMS

- Point-to-point MO and MT
- SMS cell broadcast
- Text and PDU mode

### Audio

- Triodes

Half rate (HR)

Full rate (FR)

Enhanced Full rate (EFR)

- Hands-free operation
- Echo cancellation

### GPRS

- GPRS class 8/10: max. 85.6 kbps (downlink)
- PBCCH support
- Coding schemes CS 1, 2, 3, 4
- PPP-stack

### CSD

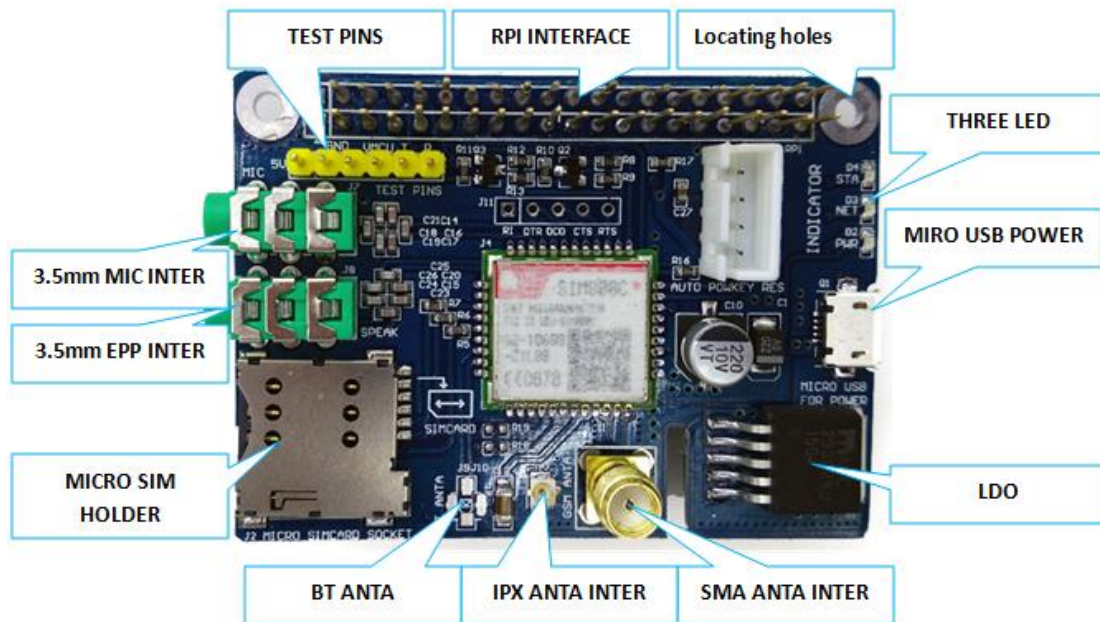
- CSD up to 14.4 kbps
- USSD
- Non transparent mode

## 二、Getting Started

A little preparation goes a long way, so make sure you've covered the following points:

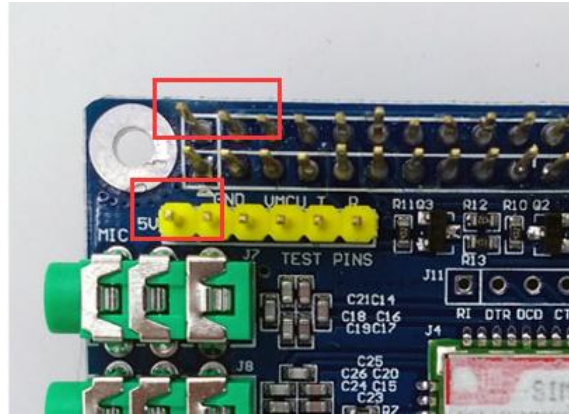
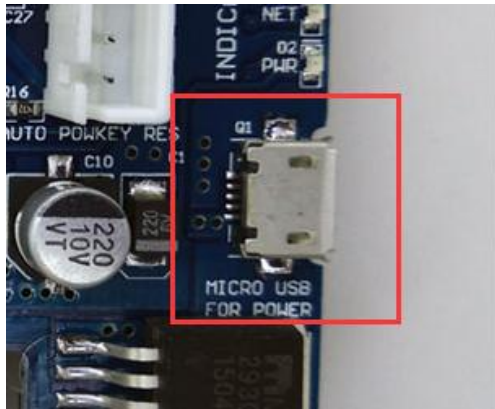
- Regarding your cellular provider. Do you have coverage on a GSM 850 MHz, GSM 900 MHz, DCS 1800 MHz or PCS 1900 MHz network? When we say GSM that means 2G – not 3G, 4G or LTE. Will they allow the use of non-supported devices on the network? Some carriers will block IMEI numbers that were not provided by their sales channel. Or you may have to call the provider and supply the IMEI of your GSM module to allow it on the network. Finally, it would be wise to use either a prepaid or an account that offers unlimited SMS text messaging – you don't want any large bills if things go wrong.
- Power. Do you have adequate power for your SIM800C module? Some shields will use more current than the RPI can supply (up to 2A), so you may need an external high-current supply. We needs 5V up to 2A. Otherwise, check with your supplier
- Antenna. If your module/shield etc. doesn't have an antenna – get one. You do need it.
- Insert the Micro SIMCARD correctly following the notes near by the SIMCARD holder.
- And as always, please don't make an auto-dialer

Furthermore, download the **SIM800C hardware manual** (.pdf) and the **AT command manual** (.pdf), as we'll refer to those throughout the tutorial.



### Power

There is a MICRO USB socket on the shield, which is for a 5V-20V power supply. And you can power the shield by other interfaces to the board.

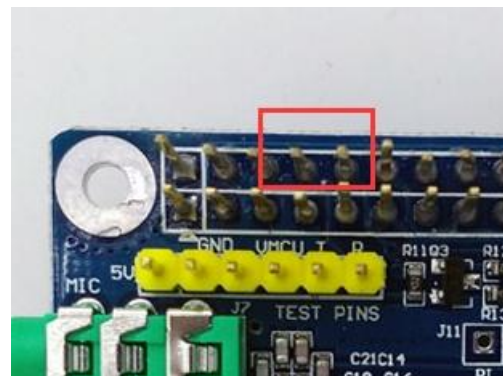
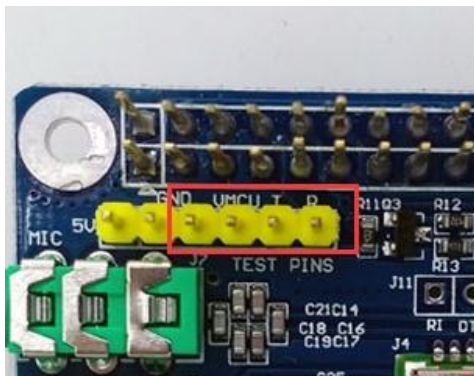


SIMCOM hardware manual for the module notes that it can draw up to 2A for short bursts. So get yourself a 5V 2A power supply and connect it via the DC socket.

If SIM800C will drop off while working, please connect a external at least 5V/2A power supply to this MICRO USB.

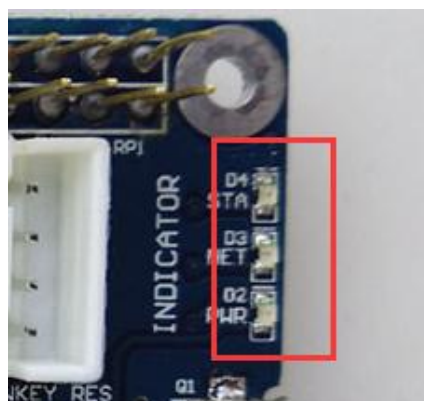
### ***UART TTL Serial***

This shield has a set a TTL UART serial, which is capable with 5v system by default. This UART serial can commutate with RPI. We can find two groups of such serial ports on the board. And there is no difference between them. The pin of VMCU is used to match the TTL level. It is connected with the 5V by default.



### ***LED indicator***

There are three LED on board, one is the POWER LED, and the other two led are the NET LED and STATUS LED. When the board is power, The bright “PWR” LED will come on, and afer five seconds later, the bright “STA” LED will come on , and then the “NET” LED will blink once every800 milliseconds- until the GSM module has found the network, at which point it will blink once every three seconds.



Nothing can happen until that magic three-second blink – so if that doesn't appear after a minute, something is wrong. Check your shield has the appropriate power supply, the antenna is connected correctly, and the SIM card is seated properly and locked in- and that your cellular account is in order. Finally, you may not have reception in that particular area, so check using a phone on the same network or move to a different location.

SIM800C NET LED is used to indicate network status; the working status of the indicator is as below:

<b>NET LED Status</b>	<b>SIM800C behavior</b>
OFF	SIM800 does not work
64ms on/800ms off	SIM800 does not find network
64ms on/3000ms off	SIM800 registers the network
64ms on/300ms off	GPRS communicates

***COMMUNICATE WITH PC THROUGH USB-TTL MODULE***

You can connect the test pins to the USB-TTL module, then inserted the MICRO SIMCARD correctly and power this board from MICRO USB finally. You can send the AT command to test SIM800C board by serial software.

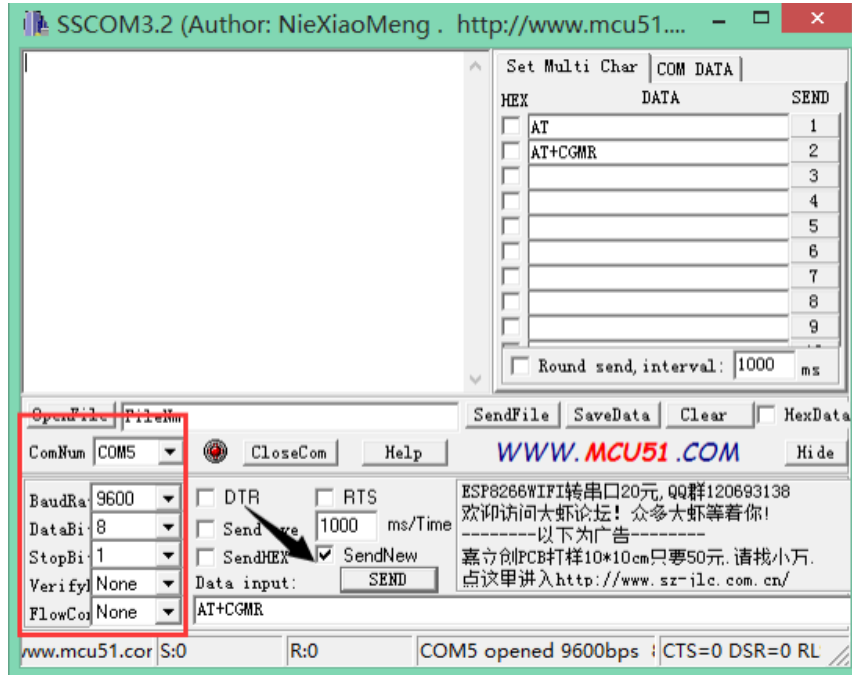
The hardware connection between SIM800C shield and USB-TTL board is:

<b>SIM800C</b>	<b>USB-TTL</b>
T	RXD
R	TXD
GND	GND

And when the NET LED is blinking, you can send AT command through the serial software.

The serial of SIM800C automatically detect the baud rate of the host device. Autobauding supports the following baud rates: 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200bps.

Here we use the SSCOM.32 serial software to test the board; the configuration of the sscom.exe is like this:



In this example, the Com Num is set to COM5. In practical application, please right click the icon My Computer->Property->Device Manager, in order to check corresponding port number.

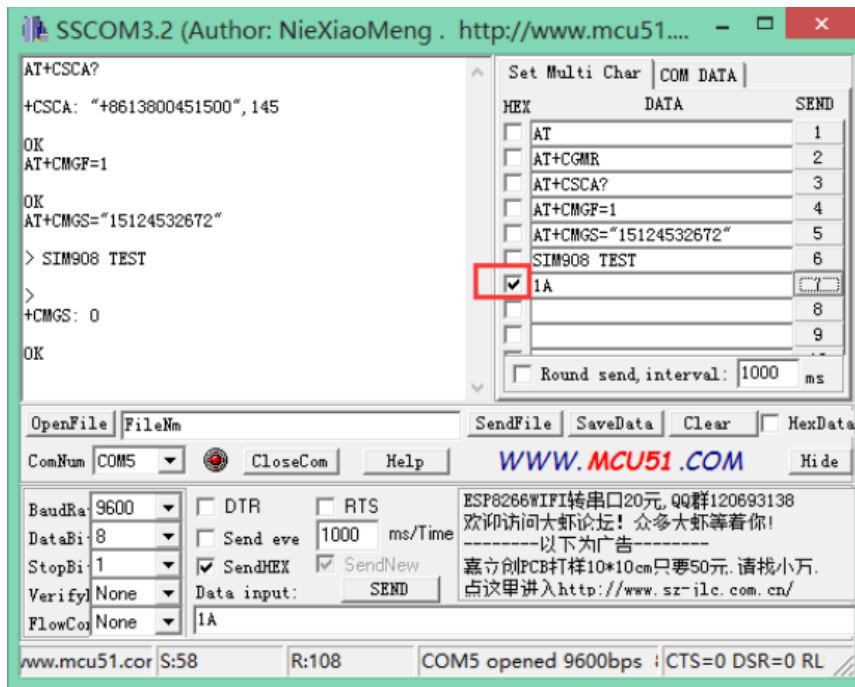
When control this module with a MCU, the software is like this, please be attention to the `\r\n`:  
**Uart1Sends("AT+CGMR\r\n");**

### SEND a message by PC

Connect the USB-TTL to the module, insert the SIMCARD, and power the shield.

Common command descriptions for message transmission

Commands	Descriptions Return	values
AT Make sure the module is working	properly	AT OK
AT+CSCA?	Get the core number of message	+cscA:"*****"
AT+CMGF=1	Select SMS message format	AT+CMGF=1 OK
AT+CMGS="15124532672"	Set the message transmission number and send SMS message. After receiving the symbol >, the message Hello World!(*) can be sent out	>
SIM908 test	The context of message	SIM908 test
1A	This is a terminator. Before sending it out, you should check the option Send As Hex	
	Send successfully	+CMGS:* OK



Other more function information please refers to  
SIM800\_Series\_AT\_Command\_Manual\_V1.07 or other application document.