



GSM/GPRS/GNSS HAT

User Manual

OVERVIEW

This is a handy, low power Raspberry Pi HAT which features multi communication functionalities: GSM, GPRS, GNSS and Bluetooth.

It allows your Pi to easily make a telephone call, send messages, connect to wireless Internet, global position, transfer data via Bluetooth, and so on.

FEATURES

GENERAL

- Raspberry Pi connectivity, compatible with Raspberry Pi 2B/3B/3B+/Zero/Zero W
- Supports SMS, phone call, GPRS, DTMF, HTTP, FTP, MMS, email, etc.
- Support GPS, COMPASS, Glonass, LBS base station positioning, omni-positioning
- Bluetooth 3.0, supports data transferring through Bluetooth
- Onboard USB TO UART converter CP2102 for UART debugging
- 6x LEDs for indicating the module working status
- SIM card slot for 1.8V/3V SIM card
- RTC with backup battery holder
- Baudrate auto detection (1200bps ~115200bps)

- Control via AT commands (3GPP TS 27.007, 27.005, and SIMCOM enhanced AT Commands)
- Supports SIM application toolkit: GSM 11.14 Release 99
- Comes with development resources and manual (examples for Raspberry Pi/Arduino/STM32)

GSM/GPRS

- Band
 - GSM 850/EGSM 900/DCS 1800/PCS 1900 MHz
 - Quad-band auto search
 - Compliant to GSM phase 2/2+
- Emitting power
 - Class 4 (2W @ GSM 850/EGSM 900 MHz)
 - Class 1 (1W @ DCS 1800/PCS 1900 MHz)
- GPRS connectivity
 - GPRS multi-slot class 12 (default)
 - GPRS multi-slot class 1~12 (configurable)
- GPRS data feature
 - Downlink speed: max 85.6kbps
 - Uplink speed: max 85.6kbps
 - Coding schemes: CS-1\CS-2\CS-3\CS-4

- Supports PAP (Password Authentication Protocol) for PPP connection
- Supports PBCCH
- Supports USSD
- SMS
 - Supports: MT/MO/CB/Text/PDU mode
 - SMS storage: SIM card
- Audio
 - Voice encode/decode mode: Half Rate\Full Rate\Enhanced Full Rate\Adaptive muti rate
 - Supports echo cancellation
 - Supports noise reduction

GNSS

- Receiver type
 - 33 tracking channels
 - 99 acquisition channels
 - GPS L1 C/A code
- Sensitivity
 - Tracking: -165 dBm
 - Cold starts : -148 dBm

- Time-To-First-Fix
 - Cold starts : 28s (typ.)
 - Hot starts : < 1s
 - Warm starts: 26s
- Accuracy
 - Horizontal position : <2.5m CEP

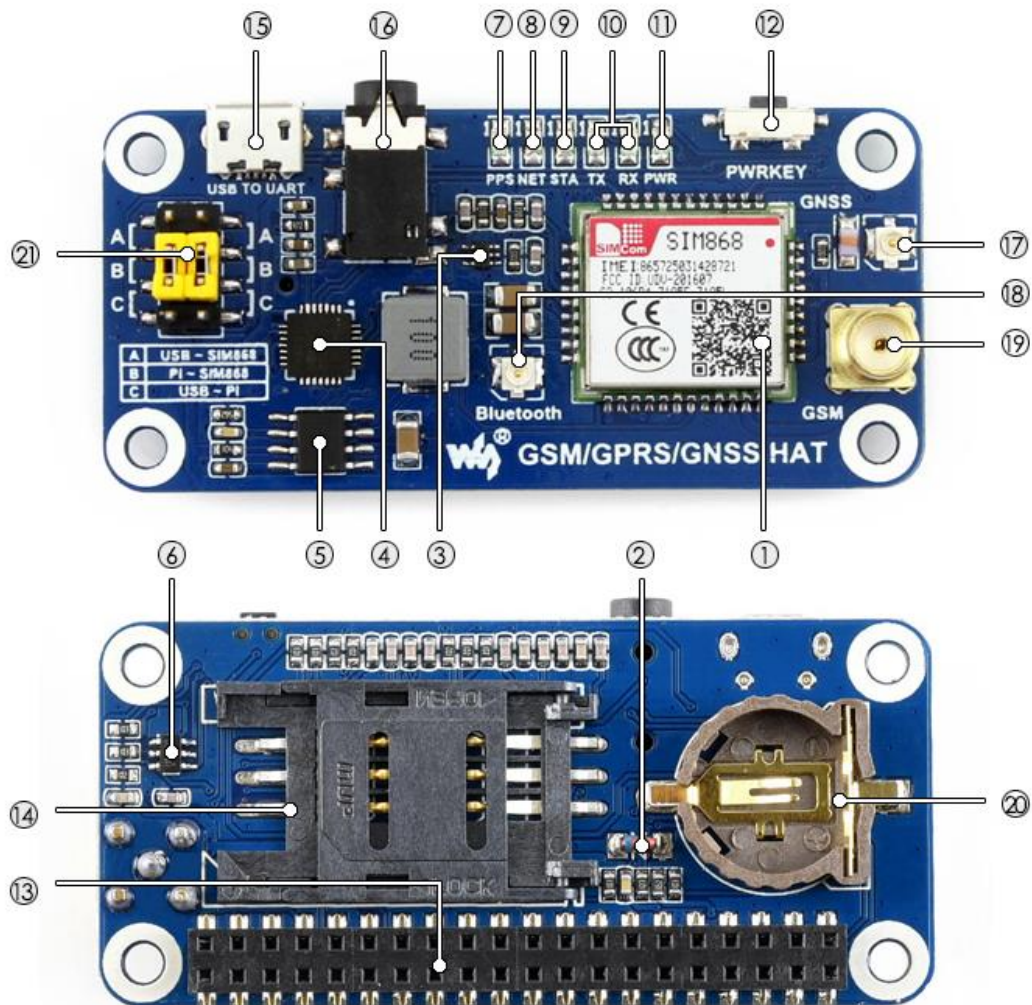
BLUETOOTH

1. Integrates AT commands
2. Compliant to Bluetooth specification 3.0 + EDR
3. Supports SPP, OPP, HFP/HSP, etc.

MISC

- Operating voltage: 5V
- Operating temperature: -40°C ~ 85°C
- Storage temperature: -45°C ~ 90°C
- Dimensions: 30.2mm x 65mm

WHAT' S ON BOARD



1. **SIM868 module**
2. **ZMM5V1:** regulator diode
3. **SMF05C:** TVS diode
4. **CP2102:** USB TO UART converter
5. **MP1482:** power chip
6. **NDC7002N:** voltage level translator
7. **GPS status indicator**

8. NET indicator:

- flashes fast when the module starts up
- flashes slowly after GSM register succeed

9. STA module working status indicator**10. SIM868 UART Tx/Rx indicator****11. Power indicator**

12. SIM868 control button: press the button and hold for 1s, to startup/shutdown the SIM868

13. Raspberry Pi GPIO connector**14. SIM card slot****15. USB TO UART interface****16. 3.5mm earphone/mic jack****17. GNSS antenna connector****18. Bluetooth antenna connector****19. GSM antenna connector**

20. CR1220 battery holder: for RTC backup battery

21. UART selection switch

- A: control the SIM868 through USB TO UART
- B: control the SIM868 through Raspberry Pi
- C: access Raspberry Pi through USB TO UART

CONTENT

Overview.....	1
Features.....	1
General.....	1
GSM/GPRS.....	2
GNSS.....	3
Bluetooth.....	4
Misc.....	4
What' s on Board.....	5
Test on PC.....	9
Hardware configuration.....	9
GSM Testing.....	11
Common AT Commands.....	11
Take calls.....	13
Answer calls.....	14
SMS.....	14
Receive SMS.....	15
GNSS Testing.....	16
Common AT commands.....	16
Configuration.....	17

GPRS Testing	19
Common AT Commands	19
Setting local virtual server.....	20
Searching WAN IP	20
Testing.....	21
Sending data.....	22
Receiving Data.....	22
Deactivate Connection	23
Bluetooth Testing	24
Common AT Commands	24
Setting	25
Connecting.....	26
Demo code (Raspberry Pi)	29
Hardware Configuration.....	29
UART Setting (Raspberry Pi)	30
Testing with minicom.....	31
Demo codes.....	31

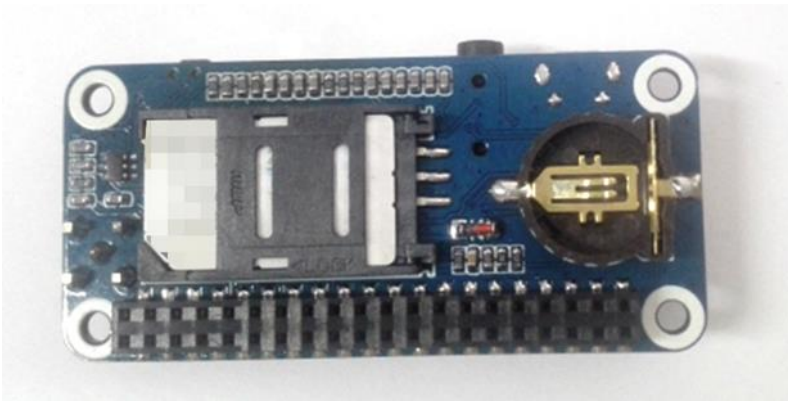
TEST ON PC

HARDWARE CONFIGURATION

This module comes with a power adapter, micro USB cable, GSM antenna, GPS antenna and Bluetooth antenna. Besides these you should prepare two more things:

- A SIM card, the card should be usable and GPRS access (for testing the GPRS)
- An earphone with a microphone on it (For testing Call function)

1. Insert the SIM card to the card slot, plug the earphone and connect the GSM antenna



2. Install CP2102 driver, plug the jumper B, and connect the USB to UART interface of GSM/GPRS/GNSS HAT to PC with a micro USB cable. Then the PWR indicator will keep bright.
3. Open Device Manager to get the corresponding COM port number of CP2102. For example, it is COM7 as below. Users need to choose the correct port according to the Manager.
4. Press the PWRKEY button and hold for 1s, the NET indicator will blink as below. 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 inserted correctly.

GSM TESTING

COMMON AT COMMANDS

Command	Description	Return
AT	Inquiry states of SIM module	OK
0x1A	End mark. Check the option " Send As Hex" then send it	
Taking call		
AT+CPIN?	Inquiry states of SIM card	+CPIN: READY
AT+CLIP	AT+CLIP=1 Enable +CLIP notification	OK
ATD<phone_number>;	Call a phone number, for example: ATD10086; Must finished with Halfwidth semicolon	OK
ATA	Answer the phone	OK
ATH	Hang up the phone	OK
SMS		
AT+CSCA=" xxxx"	Set local SMS service center	OK
AT+CMGF	AT+CMGF=1 Set the format of messages to Text mode AT+CMGF=0 Set the format of messages to PDU mode	OK

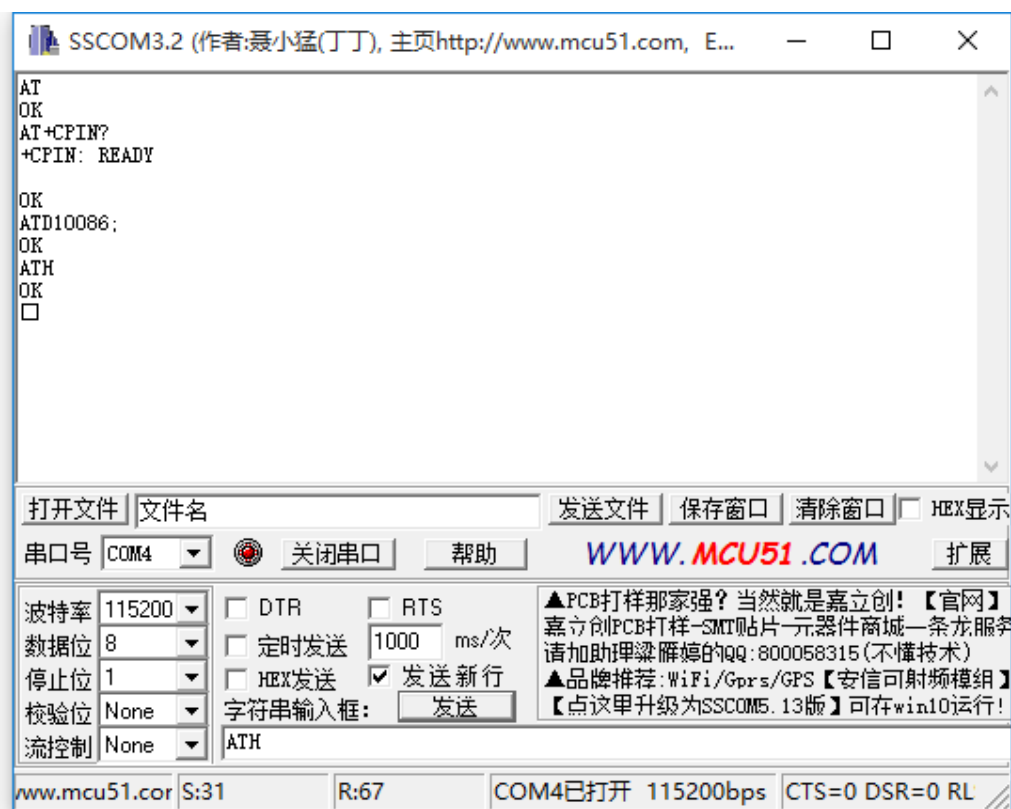
AT+CSCS	Select TE character set AT+CSCS="GSM" GSM 7 bit default alphabet AT+CSCS="UCS2" 16-bit universal multiple-octet coded character set	OK
AT+CSMP	AT+CSMP=17,168,2,25 Set SMS text mode parameters	OK
AT+CMGR	AT+CMGF=1 Read the message which is saved at place 1	
AT+CMGS	AT+CMGS="phone_number" Send SMS to the corresponding phone	>
其他指令		
ATE	ATE1 Echo mode on ATE1 Echo mode off	OK
AT+COLP	AT+COLP=1 Enable +COLP notification	AT+COLP
AT+CNMI	AT+CNMI=2,1 Enable new SMS message indicator	OK

For more details of AT commands, please refer to: [SIM800 Series AT Command](#)

[Manual V1.09](#)

TAKE CALLS

1. Insert the SIM card, connect the GSM antenna and TTL serial wires correctly. Then power on the board;
2. Check whether the indicators blink correctly (PWR and STA keep bright, NET flashes every three seconds).
3. Send “AT+CPIN?” and Enter to query the status of SIM card. Getting “+CPIN: READY” if the SIM card is ready
4. Call number: for example, “ATD10086;” (10086 is the number of China Mobile Communications Corporation)
5. Send “ATH” and Enter to hang up the call, as below:



ANSWER CALLS

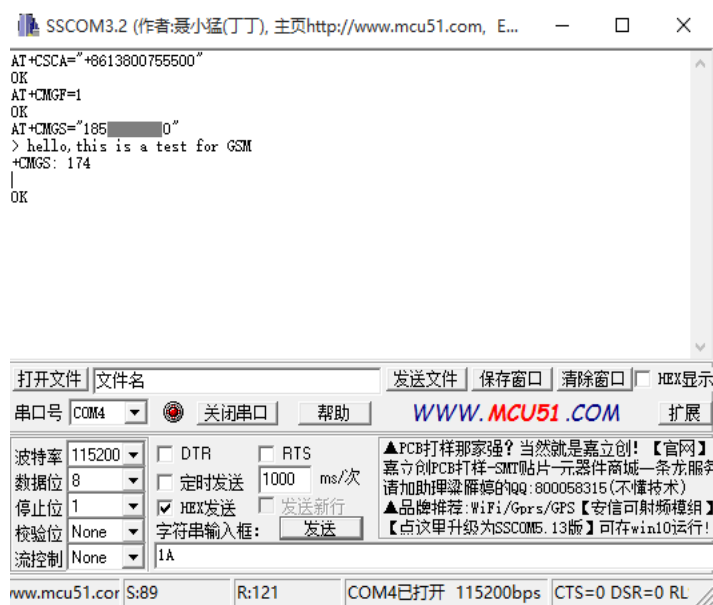
1. Enable CLIP notification: AT+CLIP=1 then Enter
2. Answering: ATA and Enter;
3. Hang up: ATH then Enter



SMS

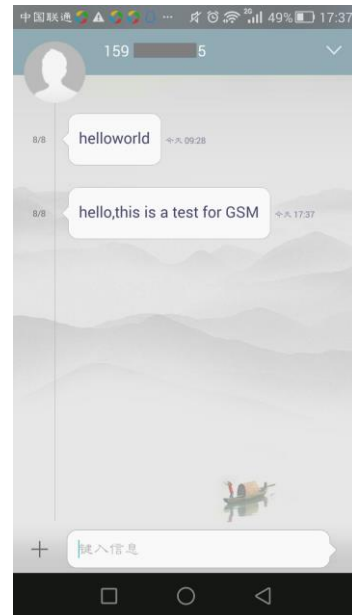
1. Set the local SMSC: AT+CSCA="+8613800755500" then Enter, get response "OK". Note: The SMSC will be different on different area. Here, it is Shenzhen China as examples.
2. AT+CMGF=1: Set SMS to TEXT mode;
3. AT+CMGS="xxxxxxxxxx" then Enter, set the number of receiver, then you will get response: >, edit the content of message (needn't Enter at the end). After editing,

send 1A in HEX format to send the message (0x1A is key value of "CTRL+Z" , it will tell the module to send). If the message is send successfully, module will get the reply +CNGS: 174 as below figures.



```

SSCOM3.2 (作者:景小猛(丁丁), 主页http://www.mcu51.com, E...
AT+CSCA="+8613800755500"
OK
AT+CMGF=1
OK
AT+CMGS="185[redacted]0"
> hello, this is a test for GSM
+CNGS: 174
OK
  
```



RECEIVE SMS

1. Send a message "this is a receive test" to the GSM/GPRS/GSNN HAT from your phone
2. While receiving messages, module will report it to UART: "SM" , 3, it means that there are 3 messages in SM, and the message we receive just now is the third message.
3. Reading messages: AT+CMGR=3 to read the third message (AT+CMGL="ALL" to read all messages)
4. Deleting message: AT+CMGD=3, to delete the message as below



GNSS TESTING

COMMON AT COMMANDS

Command	Description	Return
AT+CGNSPWR	AT+CGNSPWR=1 Turn on the power of GPS AT+CGNSPWR=0 Turn off the power of GPS	OK
AT+CGNSIPR	AT+CGNSIPR? Enquiry the baud rate of GPS AT+CGNSIPR=<Baudrate> Set the baud rate of GPS	The current baud rate OK
AT+CGNSTST	AT+CGNSTST=1 Send data received to UART AT+CGNSTST=0 Stop sending data received to UART	OK


AT+CGNSINF	GNSS navigation information parsed from NMEA sentences	Print GPS information
AT+CGPSSTATUS	Check GPS status	

For more details of AT commands, please refer to: [SIM868 Series GNSS Application](#)

[Note V1.00](#)

CONFIGURATION

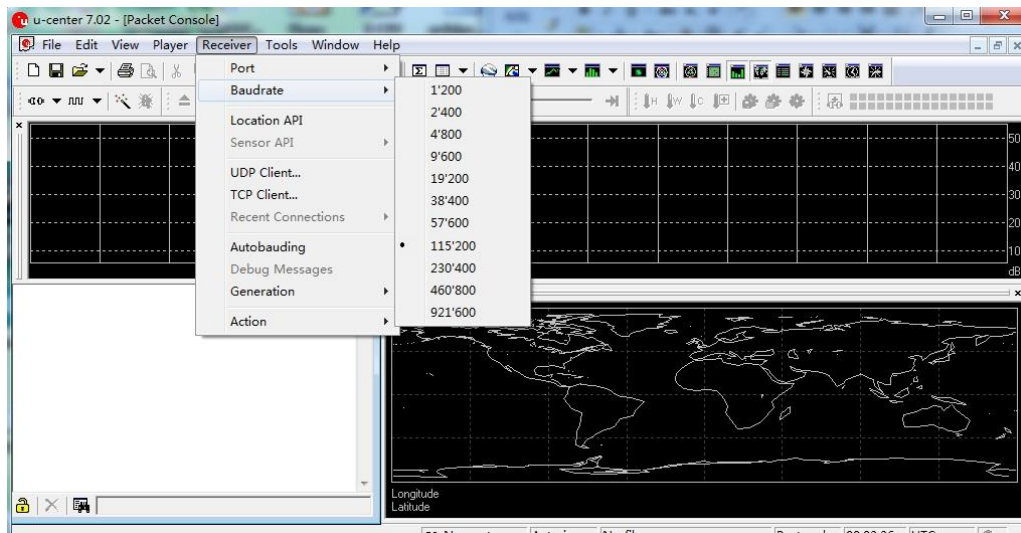
1. Connecting the GPS antenna, and place the receiver on open area to receive GPS signal
2. Turn on power of GNSS: AT+CGNSPWR=1
3. Check the baud rate: AT+CGNSIPR?
4. Start to sending data received to UART: AT+CGNSTST=1:



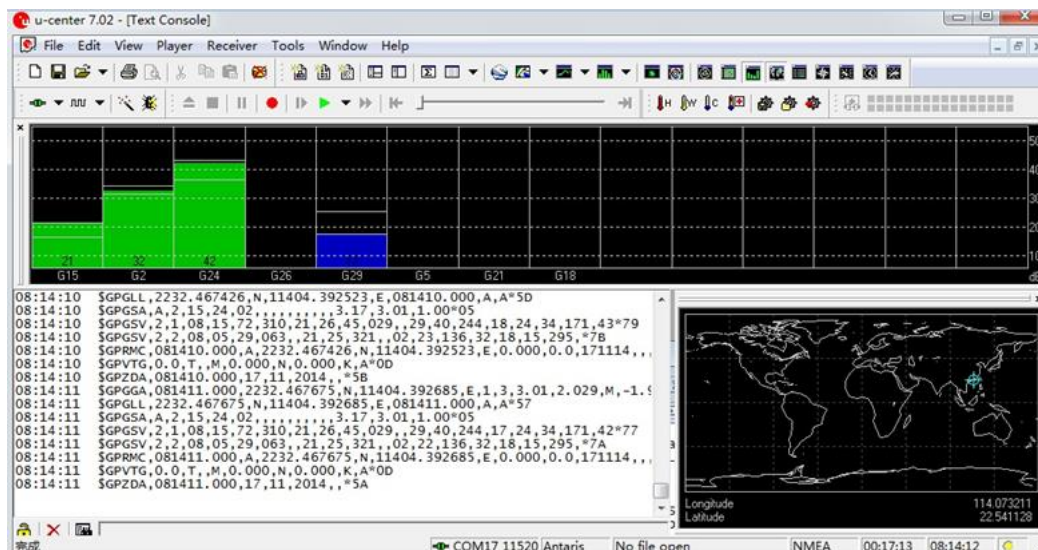
```

SSCOM3.2 (作者:聂小猛(丁丁), 主页http://www.mcu51.com, Email: mc...
AT+CGNSPWR=1
OK
AT+CGNSIPR?
ERROR
AT+CGNSIPR?
+CGNSIPR: 9600
OK
AT+CGNSTST=1
OK
$GPGGA,000013.021,,,,,0,0,,M,,M,,*49
$GPGLL,,,,,000013.021,V,N*7B
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,1,1,00*79
$GPRMC,000013.021,V,,,,,0.00,0.00,060180,,,N*43
$GPVTG,0.00,T,,M,0.00,N,0.00,K,N*32
$GPGGA,000014.021,,,,,0,0,,M,,M,,*4E
$GPGLL,,,,,000014.021,V,N*7C
  
```

- Close the Com Assistant software SSCOM. Open u-center and set the Port and Baudrate. The Port is the port number recognized by PC. Set Baudrate as the value we got before (You can configure in Receiver option or click icons on tool bar directly):



- Choose Player>, then the GNSS information will be printed as below:



GPRS TESTING

COMMON AT COMMANDS

Command	Description	Return
AT+CGATT	AT+CGATT? Check the state of GPRS attachment	+CGATT:1 1 Attached
AT+CSTT	AT+CSTT="CMNET" Set APN to CMNET	OK
AT+CIICR	Bring up wireless connection with GPRS	OK
AT+CIFSR	Get local IP address	OK
AT+CIPSTART	AT+CIPSTART="Mode", "IP_Addr", "Port" Mode: connection type; IP_Add: Remote server IP address; Port: Remote server port	CONNECT OK
AT+CIPSEND	Send data	OK
AT+CIPCLOSE	Close TCP or UDP connection	CLOSE OK
AT+CIPSHUT	Deactivate GPRS PDP Context	SHUT OK
1A	(HEX format) Tell module to send data	SEND OK

For more details of AT commands, please refer to: [SIM800 Series AT Command](#)

[Manual V1.09](#)

SETTING LOCAL VIRTUAL SERVER

Virtual server defines 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. (Please refer to guide manual of your router)

1. Log in Management Console of your router with browser (read your router's guide manual for specific address)
2. Set Port: 5000 (The Port can't be conflict to others. Here we set it to 5000)
3. Set LAN IP address for your computer (you can run CMD on your computer, and execute command ipconfig to enquiry the address of IPv4), 192.168.1.14 as examples

wan1_pp poel	TCP/UDP	5000-5000	5000-5000	192.168.1.14
-----------------	---------	-----------	-----------	--------------

SEARCHING WAN IP

You can search "IP" on browser to get WAN IP address of your PC as below: (This method is only workable in China)

[IP地址查询](#)



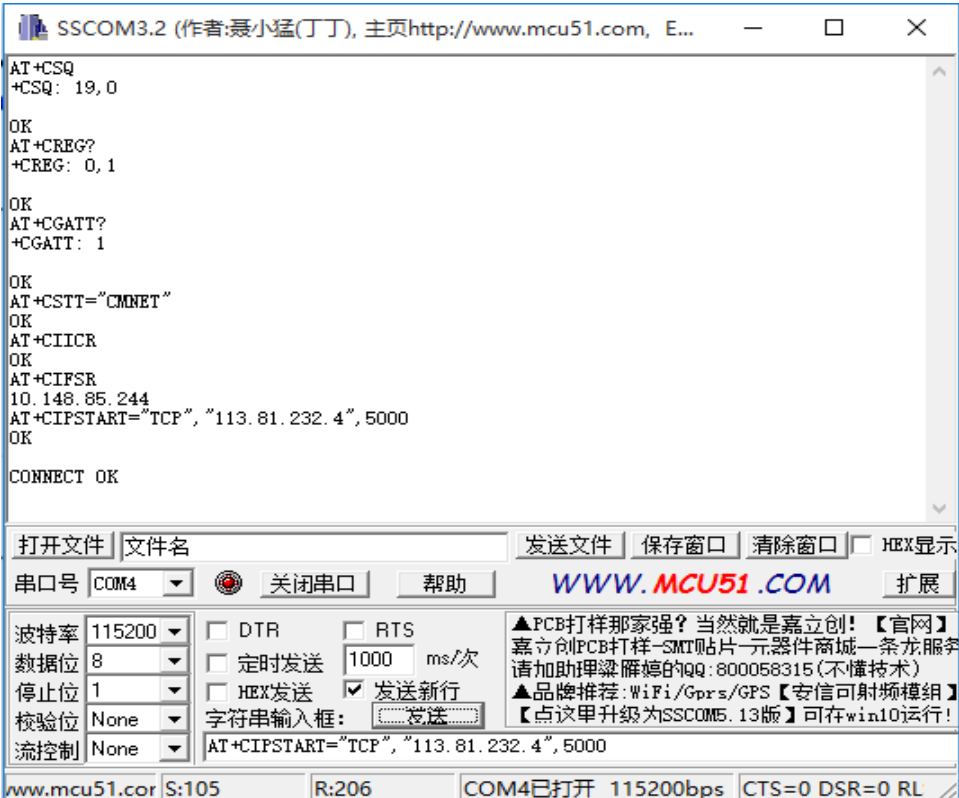
iP 本机IP: 113.81.232.4 广东省深圳市 电信

请输入ip地址

[本机IP查看方法](#) [IP地址设置方法](#)

TESTING

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+CIICR Bring up wireless connection with GPRS
6. AT+CIFSR Get the local IP address
7. AT+CIPSTART="TCP", "113.81.232.4", 5000 Establish TCP/IP connection



```
AT+CSQ
+CSQ: 19,0

OK
AT+CREG?
+CREG: 0,1

OK
AT+CGATT?
+CGATT: 1

OK
AT+CSTT="CMNET"
OK
AT+CIICR
OK
AT+CIFSR
10.148.85.244
AT+CIPSTART="TCP", "113.81.232.4", 5000
OK

CONNECT OK
```

打开文件 | 文件名 | 发送文件 | 保存窗口 | 清除窗口 | HEX显示

串口号 COM4 | 关闭串口 | 帮助 | WWW.MCU51.COM | 扩展

波特率 115200 | DTR | RTS | 定时发送 1000 ms/次 | HEX发送 | 发送新行

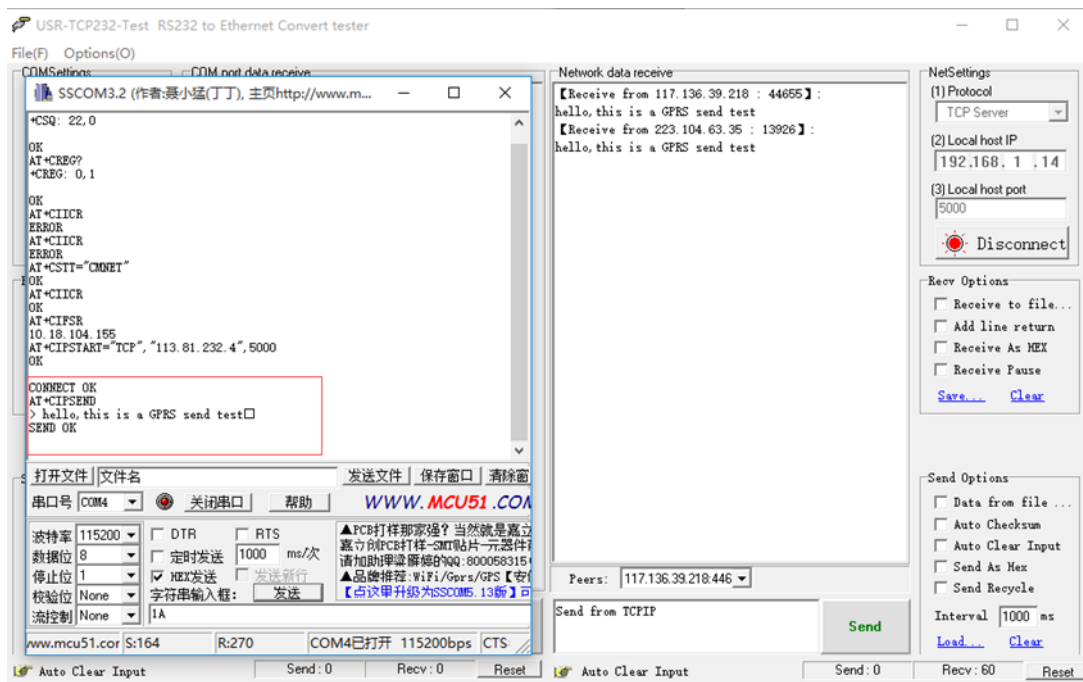
数据位 8 | 停止位 1 | 校验位 None | 流控制 None | 字符串输入框: [] 发送

▲PCB打样那家强? 当然就是嘉立创! 【官网】
嘉立创PCB打样-SMT贴片-元器件商城一条龙服务
请加助理梁婷婷的QQ:800058315(不懂技术)
▲品牌推荐:WiFi/Gprs/GPS【安信可射频模组】
【点这里升级为SSCOM5.13版】可在win10运行!

www.mcu51.cor S:105 R:206 COM4已打开 115200bps CTS=0 DSR=0 RL

SENDING DATA

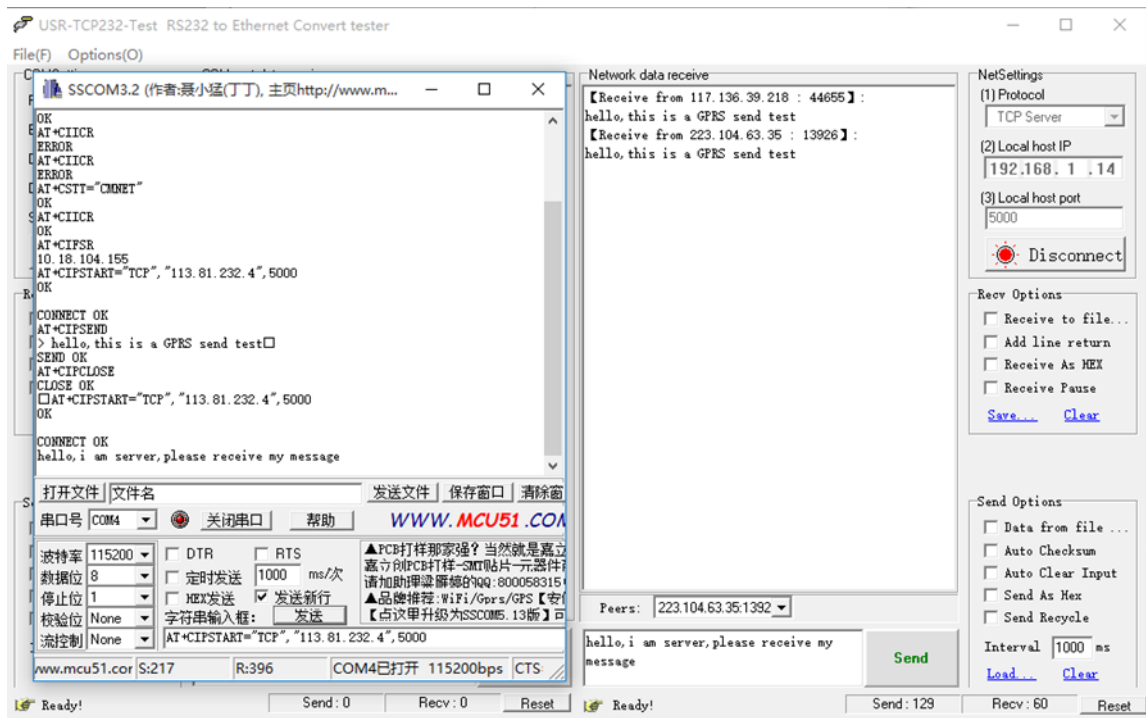
1. AT+CIPSEND, module is going to send data to server
2. After getting the response >, edit the contents of message (has been converted) without Enter at the end. Then send 1A in HEX format as below
3. If the data sent successfully, the server will receive the data.



RECEIVING DATA

1. Choose the IP address of module on Peers input box
2. Input the data which you want to send: "hello, i am server, please receive my message"

3. Click Send button, you can see that module receive the data



DEACTIVATE CONNECTION

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



BLUETOOTH TESTING

COMMON AT COMMANDS

Command	Description	Return
AT+BTPOWER	AT+BTPOWER=1 Turn on Bluetooth AT+BTPOWER=0 Turn off Bluetooth	OK
AT+BTHOST	AT+BTHOST? Enquiry the name and address of Bluetooth AT+BTHOS= <Name> Change the name of Bluetooth	Device name and MAC address of Bluetooth
AT+BTSCAN	AT+BTSCAN=1,10 Search nearby Bluetooth devices	Information of Bluetooth device searched +BTSCAN:<Num>
AT+BTPAIR	AT+BTPAIR=0,1 Pair with device 1 AT+BTPAIR=1,1 Confirm pairing	OK
AT+BTUNPAIR	AT+BTUNPAIR=0 Delete the pairing data	OK
AT+BTACPT	AT+BTACPT=1 Confirm SPP connection	OK
AT+BTSPSEND	AT+SPPSEND Send data	>

SETTING

1. Connecting the Bluetooth antenna to the GSM/GPRS/GNSS HAT. Open the Bluetooth service of your phone and set it visible to all nearby Bluetooth.
2. AT+BTPOWER=1, Turn on Bluetooth
3. AT+BTHOST? Enquiry the name and MAC address of module, you can also use AT+BTHOST=<Name> Command to change the Bluetooth name of GSM/GPRS/GNSS HAT
4. AT+BTSCAN=1,10 To search available nearby devices. Search time is 10s. During searching, module will print the information of available device to UART.

For example, +BTSCAN: 0,2,"H60-L01",50:a7:2b:bb:a4:50,-47

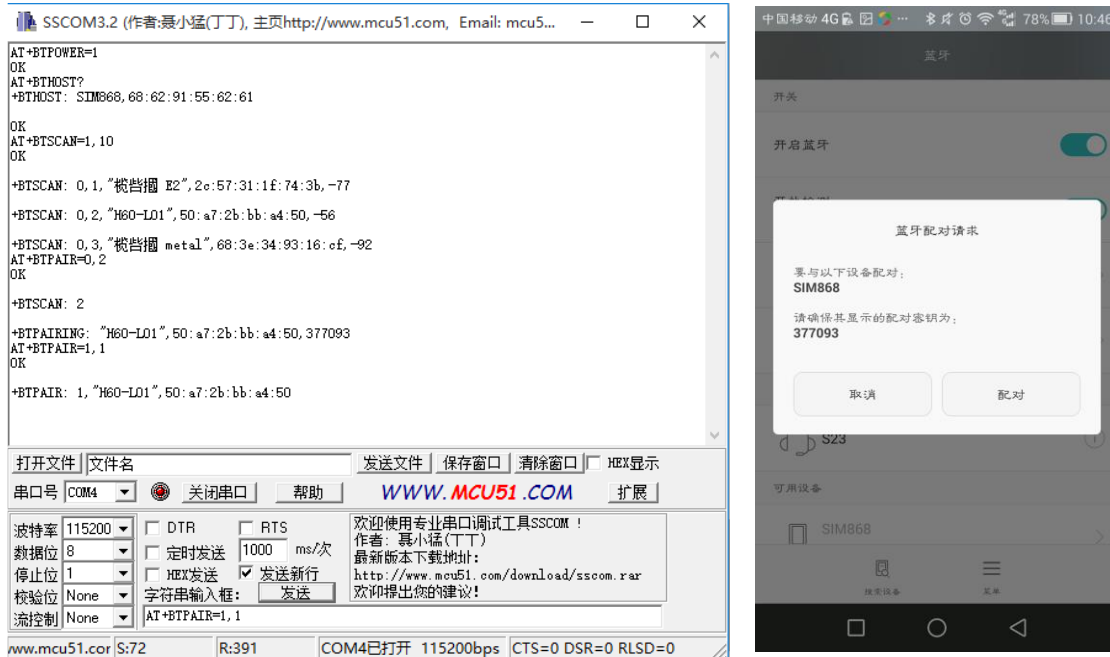
"0,2" is the ID of this device;

"H60-L01" is name of device;

"50:a7:2b:bb:a4:50" is MAC address of device;

"-47" is the RSSI value.
5. AT+BTPAIR=0,2 , Pair with available device, the second parameter here is the ID we got at step 4 . If get the response Error, just use AT+BTUNPAIR to delete the pairing data and pair again.
6. After sending the pair command successfully, you can see that your telephone prompt the pair request. Then you need to click Pair to accept the request and send AT+BTPAIR on PC to confirm the pair behavior. With these, module pair with the telephone successfully. If you don't send the AT command to confirm pairing,

the telephone will prompt information that fail to pair after a while. In this case, you need to send the pair command again. You can also use the telephone to send the pairing request, then input AT+BTPAIR=1,1 to confirm it.



CONNECTING

To test Bluetooth of GSM/GPRS/GNSS, you should use the Bluetooth APP.

Scanning the QR code below to download the APP, then you can use its Serial function to test. (The APP is just used for simply test)

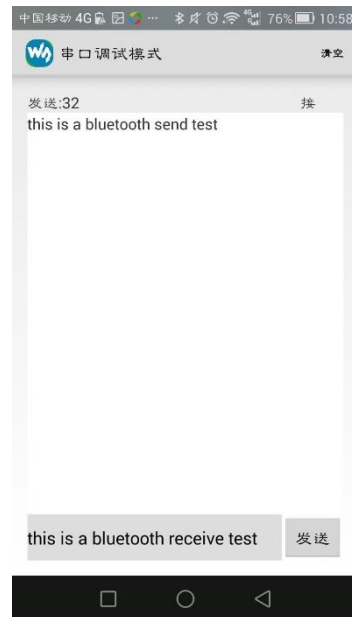


1. Open the APP, click SCAN button on the upper right to scan the available devices.

Then choose "SIM868" ("SIM868" is default Bluetooth name of GSM/GPRS/GNSS HAT) and click UART. Then you will get the information that +BTCONNECTING on the COM assistant, which means there are connecting request. If you don't reply it for a while, the APP will prompt that connecting failed. And failed information "+BTDISCONN" will be printed on PC

2. Connecting again, send command AT+BTACPT=1 to confirm and accept Bluetooth connection. You can see that +BRCONNECT is printed on PC shows that succeed in SPP connecting. Then you can send data to the module with the APP.
3. While receiving the data send from the telephone, the data will be printed with ID of device, the length of string and the content on PC.
4. Send AT+BTSPSEND command to transmit data. After getting the response >, input the content without Enter at the end. Send 1A in HEX begin to transmit the data. Then you can see that the data are received by your phone.





DEMO CODE (RASPBERRY PI)

HARDWARE CONFIGURATION

To communicate with Raspberry Pi, you can use USB interface or UART interface.

1. Inserting GSM/GPRS/GNSS HAT to Raspberry Pi

2. Jumpers Setting:

- If you want to use USB interface, connect to USB port of Pi with micro USB cable and set the jumpers to A



- If you want to use UART interface, just need to set the jumper to B as below



3. Power on your Raspberry Pi and the SIM module:

-Manually power on SIM module: Press the PWRKEY button for about 3s

-Software power on SIM module: Writing script to pull-down P4 (BCM2835 number) for about 3s. (You can use [this code](#) as reference)

UART SETTING (RASPBERRY PI)

To use UART interface of Raspberry Pi, you should first enable the hardware serial of Raspberry Pi:

```
sudo raspi-config
```

Choose Interfaces Options ->Serial ->no->yes. To close serial debug and enable the hardware serial.

Restart your Raspberry Pi

TESTING WITH MINICOM

1. Install minicom to your Raspberry Pi

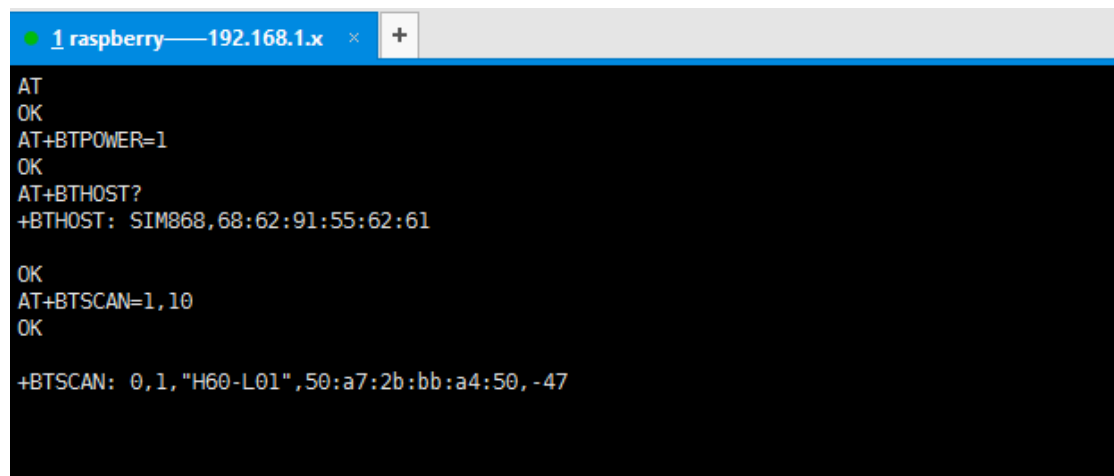
```
sudo apt-get install minicom
```

2. Open minicom for testing

```
minicom -D /dev/ttyS0
```

ttyS0 is the serial port of Pi 3B/3B+, ttyAMA0 of Pi 2B/Zero. If you use USB interface, the port maybe ttyUSB*

3. Here we test Bluetooth as example



```
1 raspberry — 192.168.1.x × +
AT
OK
AT+BTPOWER=1
OK
AT+BTHOST?
+BTHOST: SIM868,68:62:91:55:62:61

OK
AT+BTSCAN=1,10
OK

+BTSCAN: 0,1,"H60-L01",50:a7:2b:bb:a4:50,-47
```

DEMO CODES

We provide some python code for testing

1. Call_phone.py: edit this code, change the W_buf_phone to the one you want to call

and save. Then run the code with command: `sudo python call_phone.py`

```
W_buf_login = "AT+CREG?\r\n"
W_buf_phone = "ATD10086;\r\n"
ser.write(W_buf_login)

print W_buf_login
```

2. Send_message.py: change the phone number and the message as well. Then run

the code with the command: `sudo python send_message.py`

```
import serial
import time
ser = serial.Serial("/dev/ttyS0",115200)
W_buff = ["AT\r\n", "AT+CMGF=1\r\n", "AT+CSCA="+8613800755500+"\r\n", "AT+CMGS="+18000000000+"\r\n", "helloworld"]
ser.write(W_buff[0])
ser.flushInput()
```

3. gps.py: Run this code to get gps information output with command: `sudo python`

gps.py

```
#!/usr/bin/python
# Filename: text.py
import serial
import time
ser = serial.Serial("/dev/ttyS0",115200)

W_buff = ["AT+CGNSPWR=1\r\n", "AT+CGNSSEQ="RMC"\r\n", "AT+CGNSINF\r\n", "AT+CGNSURC=2\r\n","AT+CGNSTST=1\r\n"]
ser.write(W_buff[0])
ser.flushInput()
data = ""
num = 0

try:
    while True:
        while ser.inWaiting() > 0:
            data += ser.read(ser.inWaiting())
            if data != "":
                print data
                time.sleep(0.5)
                ser.write(W_buff[num+1])
                num = num + 1
                if num == 4:
                    time.sleep(0.5)
                    ser.write(W_buff[4])
                    data = ""
except KeyboardInterrupt:
    if ser != None:
        ser.close()
```