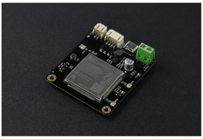
1. Introduction
2. Features
3. Application
4. Technical Specifications:
5. Functional Diagram
6. Dimensional Diagram
7. Common AT Commands
8. Sample Applications
9. Arduino Example Code

## SKU:TEL0161

## 1. Introduction



The SIM7600G is an integrated wireless communication module featuring 4G CAT1 (with an uplink speed of 5Mbps and a downlink speed of 10Mbps) and GNSS positioning (supporting GPS, BD, GLONASS). It extensively supports multiple bands, including LTE-FDD, LTE-TDD, WCDMA, and GSM, making it suitable for global regions and most operators' networks. The specific bands are as follows:

- LTE-TDD: B34/B38/B39/B40/B41
- LTE-FDD: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66
- UMTS/HSDPA/HSPA+: B1/B2/B4/B5/B6/B8/B19
- GSM/GPRS/EDGE: 850/900/1800/1900MHz

Additionally, the SIM7600G module integrates support for various network protocols including TCP/IP, IPV4, Multi-PDP, FTP, FTPS, HTTP, HTTPS, DNS, and more. This allows for complex network communications using AT commands without the need to learn or understand the underlying network protocols, offering strong compatibility and ease of use.

The SIM7600G communication module comes equipped with both UART and USB interfaces. It can connect to other MCUs or main controllers via UART or to computers, Raspberry Pi, etc., as a USB network card through the USB interface. It supports mainstream operating system drivers (Windows, Linux, Android, and other USB drivers).

Furthermore, the SIM7600G module includes a microphone and audio power amplifier circuit, enabling it not only to function for data transmission but also for making calls and sending text messages. When combined with GNSS positioning, it can be conveniently used in scenarios like anti-theft, logistics, and transportation.

The SIM series wireless communication modules offer multiple products to choose from. This series of 4G products is differentiated by communication rates and satellite positioning capabilities. The table below lists the functional parameters of each 4G product for reference in selection.

| Product name | CAT1: SIM7600G 4G communication module | CAT4: SIM7600G-H 4G communication module | CAT1: A7670G 4G communication module |
|---|---|---|---|
| Product SKU | TEL0161 | TEL0162 | TEL0163 |
| Network classification | CAT1 | CAT4 | CAT1 |
| Communication rate | 10Mbps/5Mbps | 150Mbps/50Mbps | 10Mbps/5Mbps |
| Frequency band | LTE-FDD: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66 LTE-TDD: B34/B38/B39/B40/B41 WCDMA: B1/B2/B4/B5/B6/B8/B19 GSM: 850/900/1800/1900 | LTE-FDD: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66 LTE-TDD: B34/B38/B39/B40/B41 WCDMA: B1/B2/B4/B5/B6/B8/B19 GSM: 850/900/1800/1900 | LTE-FDD: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66 LTE-TDD: B38/B39/B40/B41 GSM: 850/900/1800/1900 |
| Frequency band applicable region | global | global | global |
| GNSS satellite positioning | GNSS engine (GPS,GLONASS and BD) Protocol: NMEA 0183 | GNSS engine (GPS,GLONASS and BD) Protocol: NMEA 0183 | × |
| USB interface | √ | √ | √ |
| Onboard microphone | √ | √ | √ |
| External power supply voltage | 5V~12V | 5V~12V | 5V~12V |
| Network protocol | TCP/IP/IPV4/Multi-PDP/FTP/FTPS/HTTP/HTTPS/DNS | TCP/IP/IPV4/Multi-PDP/FTP/FTPS/HTTP/HTTPS/DNS | TCP/IP/IPV4/Multi-PDP/FTP/FTPS/HTTP/HTTPS/DNS |
| Short message | MT/MO/CB/TEXT /PDU | MT/MO/CB/TEXT /PDU | MT/MO/CB/TEXT /PDU |

## 2. Features

- Controlled via AT commands, eliminating the need to learn complex underlying network protocols.
- Supports GNSS satellite positioning (GPS, GLONASS, BD).
- Supports multiple frequency bands including LTE-FDD, LTE-TDD, WCDMA, and GSM.
- SMS functionality supports MT, MO, CB, Text, and PDU modes.

- Supports various network protocols such as TCP/IP, IPV4, Multi-PDP, FTP, FTPS, HTTP, HTTPS,and DNS.
- Capable of data communication, voice calls, and SMS functionalities.
- Provides USB network card functionality.

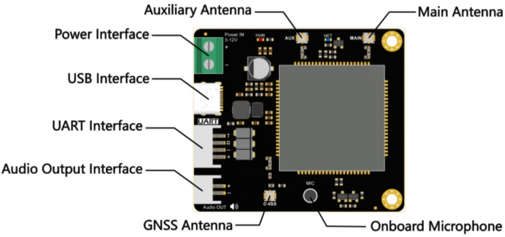
## 3. Application

- IoT data transmission and remote control.
- Mobile device monitoring.
- Logistics and transportation.
- Warehousing and theft prevention.
- Surveillance and positioning.

## 4. Technical Specifications:

- Power Supply Voltage:
  - Terminal connector: 5-12V
  - TYPE-C: 5V
- UART Interface:
  - Voltage level: 3.3V-5V
  - Physical interface: PH2.0-4P Gravity sequence standard
- Audio Output Interface: PH2.0-2P
- Frequency Band Applicable Regions: Global
- Supported Bands:
  - LTE-TDD: B34/B38/B39/B40/B41
  - LTE-FDD: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/B66
  - UMTS/HSDPA/HSPA+: B1/B2/B4/B5/B6/B8/B19
  - GSM/GPRS/EDGE: 850/900/1800/1900MHz
- 4G Communication Rate: CAT1 (Uplink speed 5Mbps, Downlink speed 10Mbps)
- GNSS Satellite Positioning: GPS, BD, GLONASS
- 4G Antennas:
  - MAIN antenna IPEX male connector ×1
  - AUX antenna IPEX male connector ×1
- GNSS Antenna:
  - IPEX antenna male connector ×1
- Voice Microphone: ×1
- SIM Card Slot: Push-lock type
- LED Indicators:
  - Power status: Red
  - Network status: Blue
- Operating Temperature: -40°C ~ 85°C
- Product Dimensions: 60mm×52mm

## 5. Functional Diagram



| Interface Name | Description |
|---|---|
| MAIN | The primary antenna interface, connects to GSM antenna |
| AUX | The auxiliary antenna interface, also connects to GSM antenna |
| GNSS | Connects to the GPS positioning antenna |
| UART | UART interface (interface voltage level 3.3V-5V) |
| USB-Type C | USB device interface, used for communication and parameter configuration with a computer |
| Power IN | External power supply required when using the UART interface (5V-12V) |
| Audio OUT | External 8Ω 1.0W speaker |
| SIM Card | Insert a SIM card |

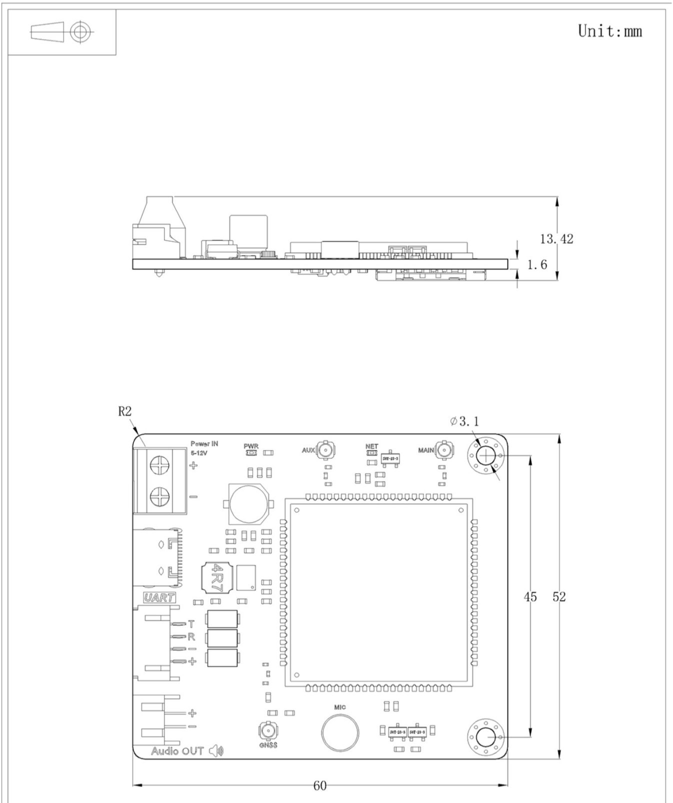
**Important Notes:**

1.When using the PH2.0 socket for module communication, an external power supply of 5V-12V is required.

2.Net - Network signal indicator:

- Steady blue light: Searching for a network.
- Rapid blue flashes (once every 1 second): Network registration completed.

## 6. Dimensional Diagram



## 7. Common AT Commands

| Command | Significance | Return Value |
|---|---|---|
| AT | AT test command | OK |
| ATE | ATE1 sets echo, ATE0 disables echo | OK |
| AT+CGMI | Query module manufacturer | OK |
| AT+CGMM | Query module model | OK |
| AT+CGSN | Query product serial number | OK |
| AT+CSUB | Query module version and chip | OK |
| AT+CGMR | Query firmware version number | OK |
| AT+IPR=9600 | Set the module's temporary hardware serial port baud rate to 9600 | +IPR: OK |
| AT+IPR? | Query the current hardware serial port baud rate | +IPR: Baud Rate |
| AT+CRESET | Reset the module (the module will restart and automatically register on the network) | OK |
| AT+CSQ | Query network signal quality, returns signal value | +CSQ: 14,99 OK |
| AT+CPIN? | Query SIM card status, returns READY if the SIM card is recognized | +CPIN: READY |
| AT+COPS? | Query the current operator, will return operator information after normal connection | +COPS: OK |
| AT+CREG? | Query network registration status | +CREG: OK |

**Note: This AT command is case-insensitive.**

## 8. Sample Applications

**Hardware Preparation:**

- SIM7600G 4G Module x1
- Windows 11 PC x1
- SIM Card x1
- USB Data Cable x1

**Software Preparation:**

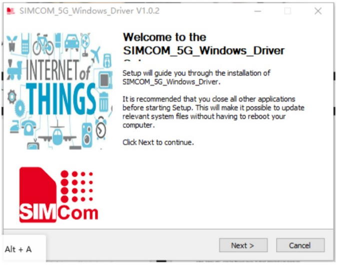
Driver Installation:

- SIMCOM Driver [Download Link]

Install the driver, unzip the package, and locate the extracted files.

SIMCOM_Windows_USB_Drivers_V1.0.2.exe

Select "Next"



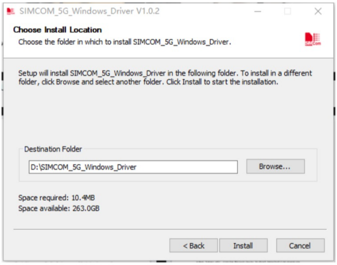
**Welcome to the SIMCOM_5G_Windows_Driver**

Setup will guide you through the installation of SIMCOM_5G_Windows_Driver.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Please install the driver file onto the disk of your choice.



**Choose Install Location**

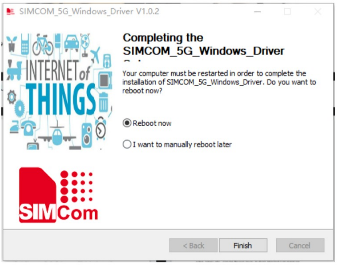
Choose the folder in which to install SIMCOM_5G_Windows_Driver.

Setup will install SIMCOM_5G_Windows_Driver in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: D:\SIMCOM_5G_Windows_Driver

Space required: 10.4MB
Space available: 263.0GB

Restart the system, and the driver installation will be successful.



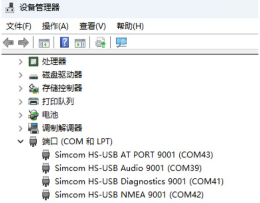
**Completing the SIMCOM_5G_Windows_Driver**

Your computer must be restarted in order to complete the installation of SIMCOM_5G_Windows_Driver. Do you want to reboot now?

● Reboot now

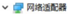
○ I want to manually reboot later

Using Windows 10 as an example, after successfully installing the driver following the aforementioned method, connect the USB-type C interface on the module and wait for the appearance of the "Simcom Port" in the computer device management interface.



- 处理器
- 磁盘驱动器
- 存储控制器
- 打印队列
- 电池
- 调制解调器
- 端口 (COM 和 LPT)
  - Simcom HS-USB AT PORT 9001 (COM43)
  - Simcom HS-USB Audio 9001 (COM39)
  - Simcom HS-USB Diagnostics 9001 (COM41)
  - Simcom HS-USB NMEA 9001 (COM42)

The "Simcom Network Adapter Driver" will appear in the network adapter interface.



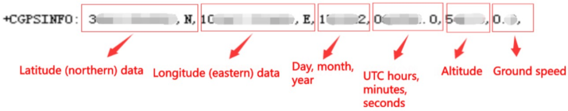
Intel(R) Wi-Fi 6 AX201 160MHz
Simcom HS-USB WWAN Adapter 9001 #3
WAN Miniport (IKEv2)
WAN Miniport (IP)
WAN Miniport (IPv6)
WAN Miniport (L2TP)
WAN Miniport (Network Monitor)
WAN Miniport (PPPOE)
WAN Miniport (PPTP)
WAN Miniport (SSTP)

**(1)Retrieve GPS positioning data.**

When utilizing GNSS positioning, first attach the GPS antenna and place the module outdoors, then power it on, and wait for one minute before initiating the positioning function.

| Command | Significance | Response |
|---|---|---|
| AT+CGPS=1 | Activate GPS | OK |
| AT+CGPSINFO | Print GPS information to the serial port | Positioning information |
| AT+CGPS=0 | Deactivate GPS | OK |

The GPS positioning data printed to the serial port appears as follows:



**Please note: The latitude and longitude data printed to the serial port cannot be directly used for positioning; it needs to be converted using a formula.**

Assuming the read data is **+CGPSINFO:1234.567891,N,12345.678912,E,141023,012523.0,455.1,0.0,**

Conversion formula:

N (Latitude) 1234.567891

(1) 1234.567891÷100=12 (rounded down)

(2) 34567891÷60=576.13152

The latitude coordinate in degree format is N 12.57613152°

E (Longitude) 12345.678912

(1) 12345.678912÷100=123 (rounded down)

(2) 45678912÷60=761.315.2

The longitude coordinate in degree format is E 123.7613152°

Open a latitude and longitude lookup website, and input the calculated latitude and longitude data in order to obtain the positioning location.

**Note:* *Upon issuing the command to print GPS information, if you fail to retrieve the GPS positional data, kindly reissue the GPS print command persistently until successful acquisition of location information is achieved.

**(2)Initiating and Receiving Phone Calls**

| Command | Significance | Response |
|---|---|---|
| AT+CNUM | Query the local number (not all SIM cards support this command) | +CNUM OK |
| AT+CSDVC | AT+CSDVC=1: Switch to headphone output AT+CSDVC=3: Switch to speaker output | OK |
| AT+CLVL=? | Query the volume range | OK |
| AT+CLVL=2 | Set the volume to 2 | OK |
| ATD; | Dial a specified target mobile number | OK |
| AT+CHUP | Hang up the call | OK |
| AT+CLIP=1 | Enable caller ID display | OK |
| ATA | Answer the call | OK |

**Demonstration of Making and Receiving Phone Calls:**

1.Before making or receiving a phone call, use the **AT+CSDVC=3** command to switch to speaker output.

2.Send the **AT+CLVL=5** command to set the speaker volume to an appropriate level.

3.Send the **ATD138xxxxxxxx;** command to dial the target user's phone number. *_Note:_ *Remember to add a semicolon at the end of the command.

**VOICE CALL: BEGIN:** Indicates the call has started. After hanging up, you will see **VOICE CALL END: xxxxx** and **NO CARRIER,** indicating the end of the call.

**RING:** Indicates an incoming call. Use the **ATA** command to answer the call.

**VOICE CALL BEGIN:** Indicates that the call has been answered.

4.During the call, use the **AT+CHUP** command to hang up.

### (3)Transmitting Text Messages

| Command | Significance | Response |
|---|---|---|
| AT+CMGF=1 | Set SMS mode to TEXT mode | OK |
| AT+CMGS="phone number" | Set the recipient's phone number | Return ">," to send the required content |
| AT+CMGR=SMS number | Read the N+1th message | Read the message |
| AT+CMGD=SMS number | Delete the N+1th message | OK |
| AT+CPMS? | Check message storage and number of messages | "SM," number of messages, maximum message limit |

**Demonstration of Sending Messages and Querying:**

1.Send the **AT+CMGF=1** command to set the SMS mode to TEXT mode.

2.Then, input **AT+CMGS="138xxxxxxxx"** to send a message to the target phone. You will receive the ">" prompt, then input the message you want to send. *_Note:_ *When dispatching the contents of the message, omit the carriage returns and line breaks.

3.After composing and sending the message, confirm it and check "SendHEX" to send the message in 16-bit hexadecimal. Send **1A** (used to instruct the module to perform the sending operation), and receive **+CMGS: ok**, indicating successful transmission.

These are the steps for sending SMS messages. If you want to inquire about the number of existing messages and the message limit, you can use the **AT+CPMS?** command. Note: The number of existing messages starts from 0.

You can also input **AT+CMGR=X** to read the specific content of the X+1th message or use **AT+CMGD=X** to delete the X+1th message.

### (4)MQTT Connection

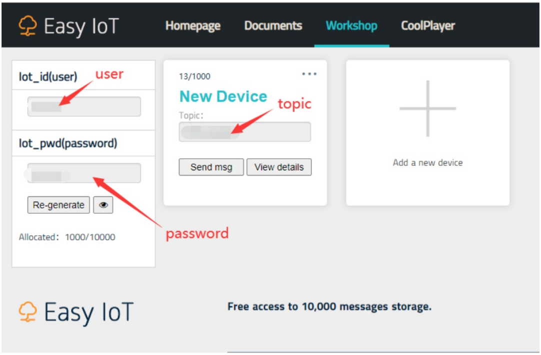
| Command | Significance | Response |
|---|---|---|
| AT+CMQTTSTART | Initiate the MQTT service | OK<br>+CMQTTSTART: 0 |
| AT+CMQTTACCQ=0,"client test ID",, | Acquire an MQTT client<br>"client test ID" can be customized; indicates the server type (0: TCP, 1: SSL/TLS); indicates the MQTT protocol version (3: MQTT version 3.1, 4: MQTT version 3.1.1) | OK |
| AT+CMQTTCONNECT=0,"",,,"user_name","password" | Connect to the MQTT server<br>must start with "tcp://". If does not include a port, the default port is 1883, e.g., tcp://116.247.119.165:5141; is the interval between receiving two messages by the client (1 second to 64800 seconds); indicates two session flags when the client disconnects from the server, defaulting to 1; | OK |
| AT+CMQTTSUB=0,,0 | Subscribe to a message from the server<br>input the length of the topic data (1 to 1024 bytes) | ><br>OK |
| AT+CMQTTTOPIC=0, | Input the message publishing topic<br>input the length of the topic data (1 to 1024 bytes) | ><br>OK |
| AT+CMQTTPAYLOAD=0, | Input the message body for publishing<br>input the length of the message data (1 to 10240 bytes) | ><br>OK |
| AT+CMQTTPUB=0,, | Publish the message to the server<br>(0: at most once, 1: at least once, 2: exactly once); (60 seconds to 180 seconds) | OK |
| AT+CMQTTDISC=0,120 | Disconnect from the server<br>Here, set the disconnect timeout value to 120 seconds, within the range of 60 seconds to 180 seconds | OK<br>+CMQTTDISC: 0,0 |

**Demonstration of Sending Messages to the Target Server:**

This example demonstrates sending a predefined message to the Easy IoT platform using AT commands with the SIM7600G-H module.

Enter Easy IoT. If you do not possess an account, you may commence by registering one.

Upon successful login, proceed to open your workspace. Retrieve your credentials: **username**, **password**. Then, proceed to add a novel device, denoted as a **Topic**, and meticulously document the aforementioned **three pivotal parameters**.



1.Open the SSCOM assistant and send the **AT+CMQTTSTART** command to initiate the module's MQTT service.

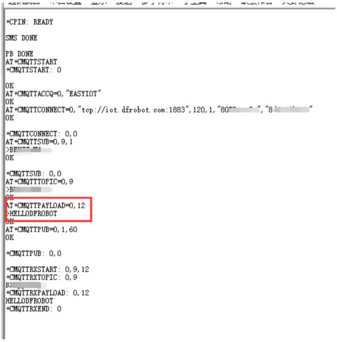
2.Send the **AT+CMQTTACCQ=0,"EASYIOT"** command (set the connection mode to TCP and customize the username as "EASIOT").

3.Send the **AT+CMQTTCONNECT=0,"tcp://iot.dfrobot.com:1883",120,1,"user","password"** command (TCP, set the server address to Easy IoT, set the message interval to 120 seconds [maximum 64800], enter the account [user] and password [password] obtained from the Easy IoT platform, and clear user information when the device is offline).

4.Send the **AT+CMQTTSUB=0,9,1** command (parameter 1 is the client parameter, parameter 2 sets the number of message publications, parameter 3 sets the publication timeout interval in seconds) and wait for the ">" prompt. Then send the **topic** obtained from the Easy IoT platform.

5.Send the **AT+CMQTTTOPIC=0,9** command (set the publishing topic, with parameter 1 as the client parameter and parameter 2 as the topic data length). Wait for the ">" prompt and then send the **topic** obtained from the Easy IoT platform.

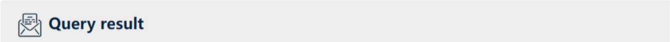
6.Send the **AT+CMQTTPAYLOAD=0,12** command (set the message content, similar to AT+CMQTTTOPIC, and directly send characters afterward; ensure the length matches). Wait for the ">" prompt and then send the message content (HELLODFROBOT).

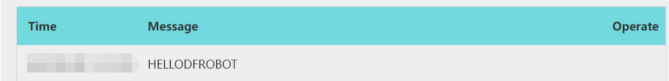


```
+CPIN: READY

SMS DONE

PB DONE
AT+CMQTTSTART
+CMQTTSTART: 0

OK
AT+CMQTTACCQ=0,"EASYIOT"
OK
AT+CMQTTCONNECT=0,"tcp://iot.dfrobot.com:1883",120,1,"80[redacted]","8[redacted]"
OK

+CMQTTCONNECT: 0,0
AT+CMQTTSUB=0,9,1
>BF[redacted]
OK

+CMQTTSUB: 0,0
AT+CMQTTTOPIC=0,9
>B[redacted]
OK
AT+CMQTTPAYLOAD=0,12
>HELLODFROBOT
OK
AT+CMQTTPUB=0,1,60
OK

+CMQTTPUB: 0,0

+CMQTTRXSTART: 0,9,12
+CMQTTRXTOPIC: 0,9
B[redacted]
+CMQTTRXPAYLOAD: 0,12
HELLODFROBOT
+CMQTTRXEND: 0
```

7.Send the **AT+CMQTTPUB=0,1,60** command (publish the message, parameter 1 is the client parameter, parameter 2 sets the number of message publications, and parameter 3 sets the publication timeout interval in seconds).

8.Finally, access the Easy IoT workspace, check the detailed information of the created topic, and verify that the message (HELLODFROBOT) sent from the module has been received.





| Time | Message | Operate |
|---|---|---|
| [redacted] | HELLODFROBOT | |

**Note: Line breaks should also be entered when inputting topics and messages.**

**(5)Audio Playback**

| Command | Significance | Response |
|---|---|---|
| AT+CREC=1,"e:/rec.wav" | Start recording a WAV audio and save it to the local path e:/rec.wav. "1" represents the local path. | +CREC: 1<br>OK |
| AT+CREC=0 | Stop recording WAV audio | +CREC: 0<br>OK<br>+RECSTATE: crec stop |
| AT+CCMXPLAYWAV="E:/rec.wav",2 | Play a WAV audio file from the local path. "2" represents the local path. | +WAVSTATE: wav play<br>OK<br>+WAVSTATE: wav play stop |
| AT+CCMXSTOPWAV | Stop playing WAV audio files | +CCMXSTOPWAV:<br>OK<br>+WAVSTATE: wav play stop |
| AT+CCMXPLAY="E:/rec.mp3",0,0 | Play an audio file from the local path. "0,0" indicates default local playback and playback once. | +CCMXPLAY:<br>OK<br>+AUDIOSTATE: audio play<br>+AUDIOSTATE: audio play stop |
| AT+CCMXSTOP | Stop playing audio files | +CCMXSTOP:<br>OK<br>+AUDIOSTATE: audio play stop |

**Recording and Playing Back WAV Audio - Demonstration**

1. Enter **AT+CSDVC=3** to switch to speaker output. **Note:* *You need to set this command each time after booting if you want to use speaker output.
2. Send **AT+CLVL=3** to set an appropriate speaker volume.
3. Input **AT+CREC=1,"e:/rec.wav"** to start recording WAV audio and save it to the local path e:/rec.wav. A return of **+CREC: 1, OK** indicates successful recording, while **ERROR** indicates recording failure.**Note:* *Use only English characters when sending AT commands.
4. Send **AT+CREC=0** to stop recording WAV audio. To play the previously recorded WAV audio, use the command **AT+CCMXPLAYWAV="E:/rec.wav",2**. If you want to stop playback, you can use the **AT+CCMXSTOPWAV** command.

   **Note: Please place the audio files into the directory designated as "e:/".**

**(6)Power Saving Mode**

| Command | Significance | Response |
|---|---|---|
| AT+CFUN=0 | Minimum functional mode. The serial port can still be used, but functions related to RF and SIM card, as well as some AT commands, cannot be used. | OK |
| AT+CFUN=1 | Full functional mode with maximum power. | OK |

**(7)Sleep Mode**

In sleep mode, the module can still receive paging, SMS, and voice calls from the network.

| Command | Significance | Response |
|---|---|---|
| AT+CSCLK=0 | UART always on input. | OK |
| AT+CSCLK=1 | UART enters sleep mode when DTR is high and exits sleep mode when DTR is low. | OK |
| AT+CSCLK? | View the current UART sleep state. | +CSCLK: Status OK |

## 9. Arduino Example Code

In addition to the methods mentioned earlier for controlling the SIM7600G module via PC and onboard USB virtual serial port, you can also use Arduino to send AT commands to debug and operate the SIM7600G module.

In this example, we will modify the baud rate of the 7600G module and the main controller.

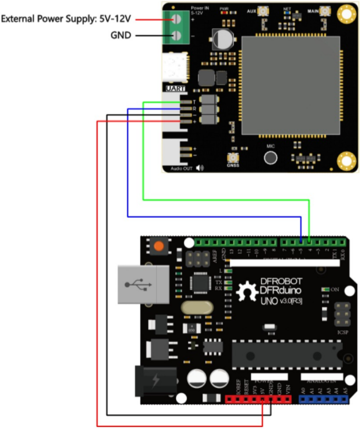
Hardware Preparation:

- SIM7600G 4G Module x1
- Windows 11 PC x1
- SIM card x1
- DFRduino UNO R3 x1
- 5V-12V external power supply

Software Preparation:

- Latest version of Arduino IDE Download Arduino IDE

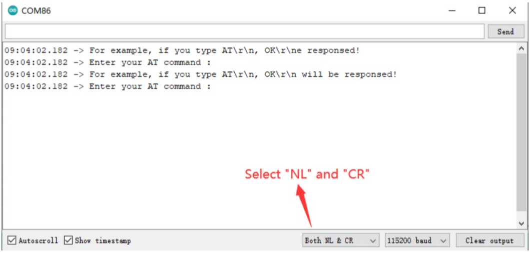
Connection Diagram and Steps:



### Steps

1. Insert the SIM card into the expansion board and connect it to the DFRduino UNO R3 mainboard as shown in the diagram. Connect an external power supply as well.
2. Wait for the SIM card to register on the network. The NET status indicator LED should start flashing rapidly (once every 1 second).
3. Download the program below to the DFRduino UNO R3 and open the serial monitor, setting the baud rate to 115200.

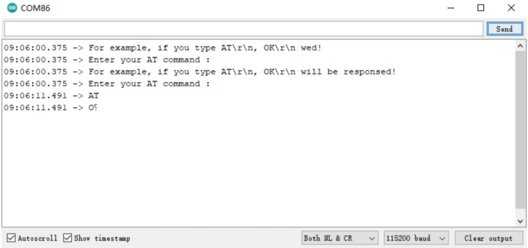
```
#include "SoftwareSerial.h"
   SoftwareSerial softSerial(/*rx =*/4, /*tx =*/5);

void setup() {

  Serial.begin(115200); // Initialize the hardware UART
  softSerial.begin(115200); // Initialize the software UART
  Serial.println("For example, if you type AT\\r\\n, OK\\r\\n will be responsed!");
  Serial.println("Enter your AT command :");
}

void loop() {
  if (softSerial.available()) {
    char data = softSerial.read(); // Read data from the software UART
    Serial.write(data); // Transmit to the hardware UART
  }

  if (Serial.available()) {
    char data = Serial.read(); // Read data from the hardware UART
    softSerial.write(data); // Transmit to the software UART
  }
}
```



```
09:04:02.182 -> For example, if you type AT\r\n, OK\r\ne responded!
09:04:02.182 -> Enter your AT command :
09:04:02.182 -> For example, if you type AT\r\n, OK\r\n will be responsed!
09:04:02.182 -> Enter your AT command :
```

Select "NL" and "CR"

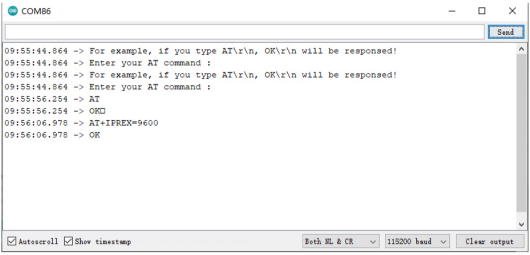
Transmitting AT commands, receiving "OK" as the serial response, signifies successful communication.



```
09:06:00.375 -> For example, if you type AT\r\n, OK\r\n wed!
09:06:00.375 -> Enter your AT command :
09:06:00.375 -> For example, if you type AT\r\n, OK\r\n will be responsed!
09:06:00.375 -> Enter your AT command :
09:06:11.491 -> AT
09:06:11.491 -> O⸮
```

Please note that when using an Arduino UNO for serial communication, the module's default communication rate is 115200. The oscillation frequency of the external crystal can result in significant baud rate errors, causing data corruption. It is necessary to set both the module's baud rate and the program's baud rate to 9600.

To achieve this, send the command "AT+IPREX=9600" to set the module's baud rate to 9600 after a restart.



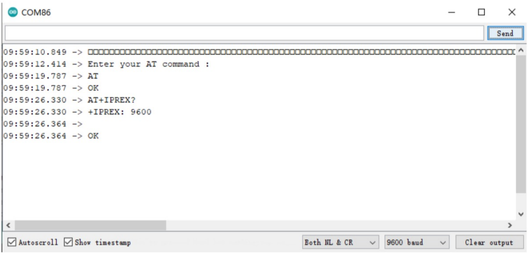
```
09:55:44.864 -> For example, if you type AT\r\n, OK\r\n will be responsed!
09:55:44.864 -> Enter your AT command :
09:55:44.864 -> For example, if you type AT\r\n, OK\r\n will be responsed!
09:55:44.864 -> Enter your AT command :
09:55:56.254 -> AT
09:55:56.254 -> OK□
09:56:06.978 -> AT+IPREX=9600
09:56:06.978 -> OK
```

Subsequently, modify the UNO's serial communication rate and re-upload the program.

```
#include "SoftwareSerial.h"
   SoftwareSerial softSerial(/*rx =*/4, /*tx =*/5);

void setup() {

  Serial.begin(9600); // Initialize the hardware UART
  softSerial.begin(9600); // Initialize the software UART
  Serial.println("For example, if you type AT\\r\\n, OK\\r\\n will be responsed!");
  Serial.println("Enter your AT command :");
}

void loop() {
  if (softSerial.available()) {
    char data = softSerial.read(); // Read data from the software UART
    Serial.write(data); // Transmit to the hardware UART
  }

  if (Serial.available()) {
    char data = Serial.read(); // Read data from the hardware UART
```

```
    softSerial.write(data); // Transmit to the software UART
  }
}
```

Using a baud rate of 9600, send "AT" and "AT+IPREX?" to the serial port, which should return "OK" and the current baud rate value of 9600, confirming the successful modification of the communication rate.



```
09:59:10.849 -> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
09:59:12.414 -> Enter your AT command :
09:59:19.787 -> AT
09:59:19.787 -> OK
09:59:26.330 -> AT+IPREX?
09:59:26.330 -> +IPREX: 9600
09:59:26.364 ->
09:59:26.364 -> OK
```

☑ Autoscroll ☑ Show timestamp

Both NL & CR

9600 baud

## 10. Internet Access on Raspberry Pi Raspbian

### RNDIS Dial-Up Internet

**Hardware Preparation:**

- SIM7600G 4G Module ×1
- Raspberry Pi 5 ×1

**Operation Steps:**

- First, insert the SD card with the Raspberry Pi system and the SIM card into the Raspberry Pi and the SIM7600G 4G Module slot respectively
- Then power on and open the Raspberry Pi terminal

**Software Preparation**

1. Install the Linux platform serial port debugging tool minicom

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

If an error message "Cannot find the source of the specified version of lrzsz and minicom" occurs

You can update the software source list, run the following command in the terminal, and then try to install minicom

```
sudo apt-get update
sudo apt-get upgrade
```

2. After installation, run minicom:

```
minicom -D /dev/ttyUSB2
```

Copy

Note: The default baud rate of the serial port debugging tool and the SIM7600G 4G Module is 115200

If an error message "minicom: Cannot open file /dev/ttyUSB2: Device or resource busy" appears, disconnect and reconnect the SIM7600G 4G Module, and wait for the network indicator light to light up before trying to run minicom with the command

**Note:** minicom can enter the help menu by pressing Ctrl+A and then Z

In minicom, if the entered command is not displayed, it may be because the local echo (Local Echo) is turned off, you can turn on the local echo through the following steps

1. In minicom, press "Ctrl+A" to enter command mode.
2. Then press "Z" to display the help menu.
3. In the help menu, press "E" to switch the echo mode.
4. Next, send a command through minicom to change the PID of the module and wait for the module to restart

```
AT+CUSBPIDSWITCH=9011,1,1
```

4. After restarting, use the ifconfig command to check whether the usb0 network card is recognized

```
ifconfig
```

At this time, you can see the network card and address of usb0

5. Get IP address

```
sudo dhclient -v usb0
```

6. Finally, test whether the usb0 wireless network card can access the Internet normally by pinging Baidu:

```
ping -I usb0 www.baidu.com
```

Note: If you need to switch to use the Windows system, or restore the default settings of the module, you can send the corresponding command through the serial port:

```
AT+CUSBPIDSWITCH=9001,1,1
```

## 11. Internet Access on LattePanda

**Hardware Preparation:**

- SIM7600G 4G Module ×1
- LattePanda ×1
- 11.6-inch FHD Touch Screen ×1
- Keyboard ×1

Operation steps:

- Insert the SIM card into the SIM7600G 4G Module slot, then connect the module and LattePanda via USB, and also connect the keyboard and touch screen to LattePanda.
- Power on and wait for LattePanda to boot
- Under network status, install the SIMCOM driver according to the method in the sample application
    - SIMCOM driver [Download address]
- After installation, restart LattePanda and check the Simcom port in the device management interface to confirm that the driver of the 4G module has been successfully installed.
- Finally, find the 4G module as a new network connection in "Change Adapter Settings", right-click and select "Connect", and you can use the 4G module as a network card for LattePanda.

## 12. Querying Result Codes and Their Meanings

| Detailed Result Code | Meaning |
|---|---|
| OK | Command executed successfully with no errors |
| CONNECT | Connection successful |
| RING | Ring detected (incoming call received) |
| NO CARRIER | No established connection or connection disconnected |
| ERROR | Invalid command or command failed |
| NO DIALTONE | No dial tone, unable to dial, incorrect mode |
| BUSY | Currently in a call or server is busy |
| NO ANSWER | Connection timed out |

### More Document Downloads

- TEL0161_SIM7600G 4G Module_Schematic.pdf
- AT Command Set
- TEL0161_2D_CAD.pdf
- TEL0161_2D_DXF.rar
- TEL0161_3D_STP.rar

## FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum**.

Downloaded from Arrow.com.