



3G/GPRS/GSM Shield Datasheet



Tinysine Electronics @ 2017
Version 1.1

INTRODUCTION

TinySine 3G shield has better coverage, GSM backwards-compatibility and even sports a built-in GPS module for geolocation & asset tracking. This all-in-one cellular phone module with that lets you add location-tracking, voice, text, SMS and data to your project in a single shield.

This shield packs a surprising amount of technology into its little frame. At the heart is a powerful GSM cellular module with integrated GPS. On its own, this shield can't do anything. It requires a microcontroller to drive it! We suggest using an Arduino or any 3-5V microcontroller with a UART can send and receive commands over the RX/TX pins.

SPECIFICATIONS

3G/GPRS specifications:

- Quad-band 850MHz GSM, 900MHz EGSM, 1800MHz DCS, 1900MHz PCS - connect onto any global GSM network with any 2G SIM.
- American Version(SIM5320A): dual-band UMTS/HSDPA 850/1900MHz WCDMA + HSDPA
- European Version(SIM5320E): dual-band UMTS/HSDPA 900/2100MHz WCDMA + HSDPA
- Fully-integrated GPS that can be controlled and query over the same serial port
- Make and receive voice calls using a headset or an external 8 Ω speaker + electret microphone
- Send and receive SMS messages
- Send and receive GPRS data (TCP/IP, HTTP, etc.)
- AT command interface can be used with 300, 600, 1200, 4800, 9600, 19200, 39400, 57600, 115200, 230K, 461K, 961K, 3.2M, 3.7M and 4.0Mbps
- Native USB support - plug it into a computer and you'll get serial ports for AT commands, GPS NMEA as well as a modem (note we've only tried out the AT&NMEA ports on Windows)

GPS specifications

- 16 acquisition channels
- GPS L1 C/A code
- Sensitivity
 - Tracking: -157dBm
 - Cold starts: -144dBm
- Time-To-First-Fix
 - Cold starts: 100s (typ.)
 - Hot starts: 1s (typ.)
- Accuracy: approx 2.5 meters

Overview





The indicator LEDs

TinySine 3G Shield has three indicator LEDs for the 3G Shield power, SIM5320x power and net status.

3G Shield power(P)

This LED is used to indicate the power status of the 3G Shield. If the external power supply is connected to the arduino board, then the 3G Shield will get power, this LED will light up.

SIM5320x power(S)

This LED is used to indicate the power status of the SIM5320x. After the SIM5320x module is powered on, the status LED will light up.

Net Status(W)

This LED is used to indicate the net status. The LED will blink slowly or quickly according to different states.

LED Status	Module Status
Always ON	Searching Network/Call Connecting
200ms ON,200ms OFF	Data Transmit
800ms ON,800ms OFF	Registered Network
Always OFF	Power OFF/Sleep

SIM5320x Power Button

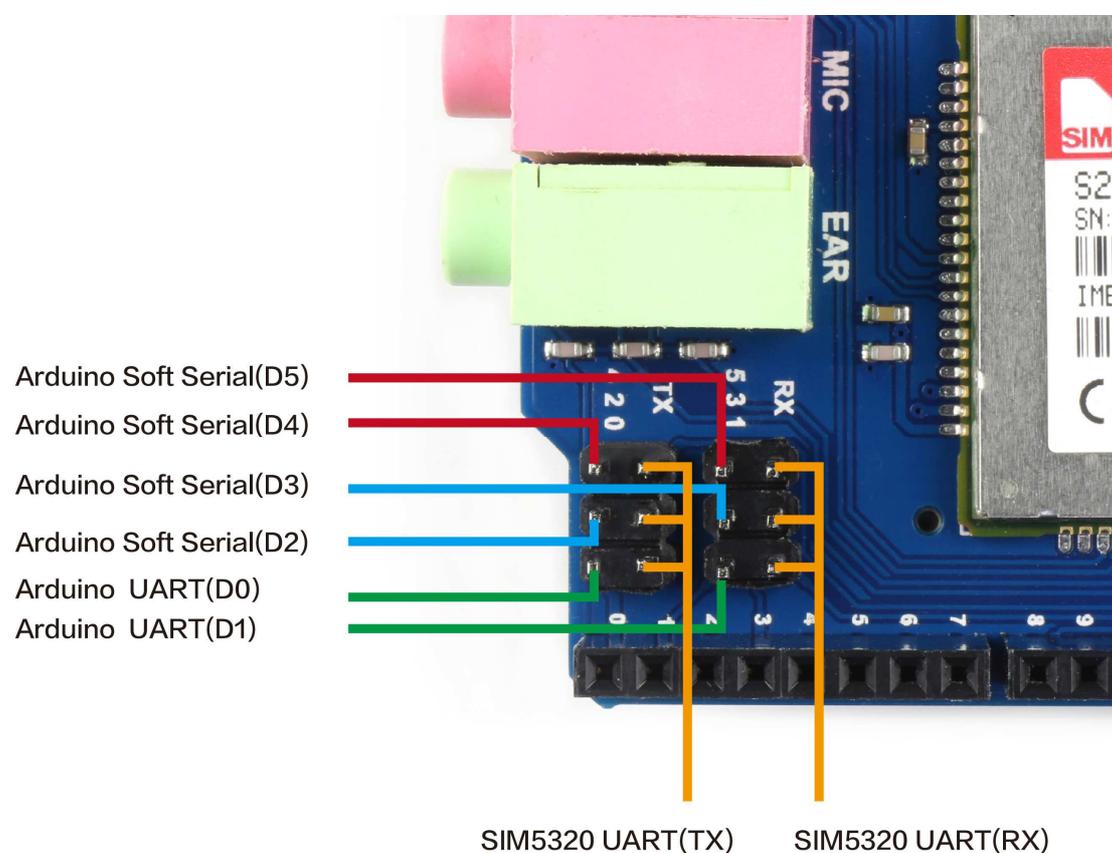
Turn On

Press the button for one second to turn on the SIM5320x module.

Turn Off

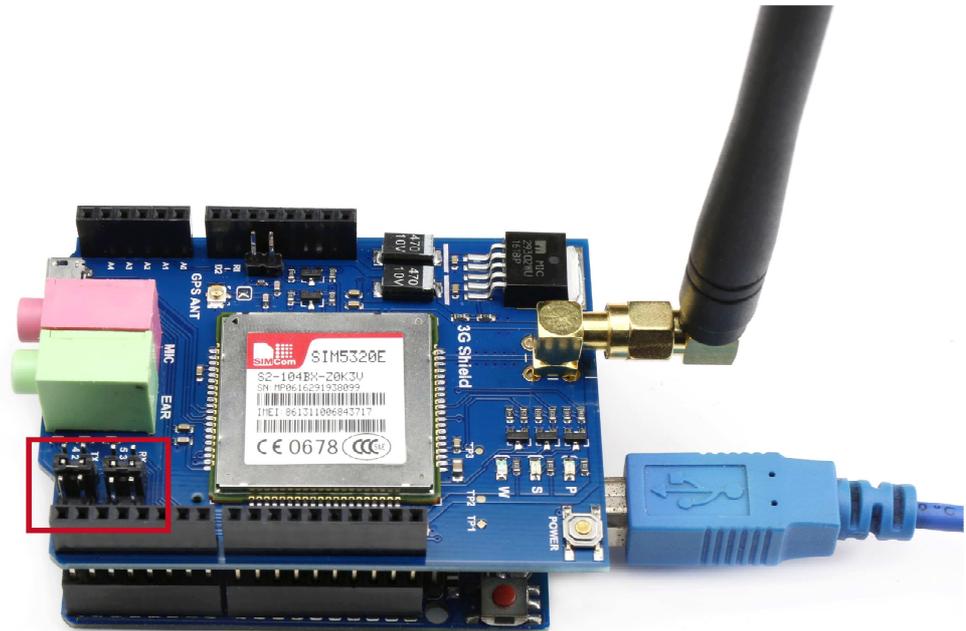
Press the button for about 3 seconds the SIM5320x module will be turned off.

Setting Jumpers



We provide three serial ports jumpers on 3G shield: UART(D0,D1), Softserial1(D2,D3) and Softserial2(D4,D5). The Softserial1 is for default code setting.

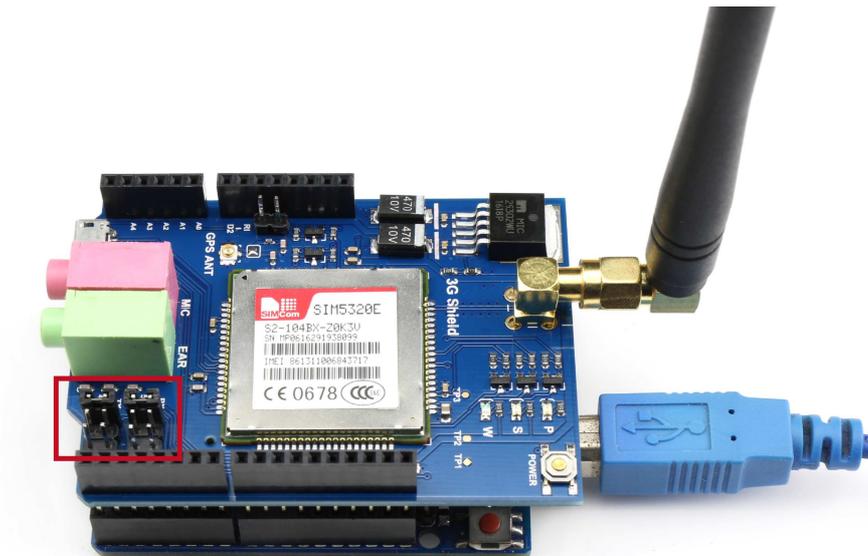
Hardware connection with Arduino UNO



The sample code that we provide uses D2,D3 as the soft serial. You don't need to change any code if you use the default settings.

Use D4,D5 as the soft serial

If you want to use D4,D5 on the Arduino board as soft serial port to communicate with the 3G shield, you need to set the jumper to Soft Serial2 (D4,D5) position and modify the code as follows.



```

FONA3G_setbaud $
#include "Adafruit_FONA.h"
#define FONA_TX 4
#define FONA_RX 5
#define FONA_RST 9

// this is a large buffer for replies
char replybuffer[255];

// We default to using software serial. If you want to use hardware serial
// (because softserial isnt supported) comment out the following three lines
// and uncomment the HardwareSerial line
#include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;

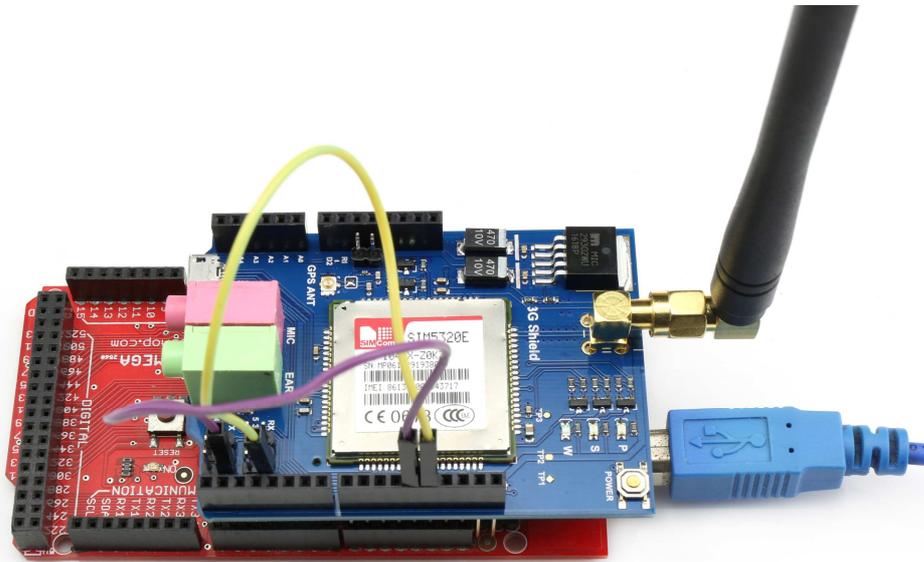
```

Hardware connection with Arduino MEGA

There are only some of the pins on the Mega and Mega 2560 supporting interrupts, thus you can only use the following ports as RX ports:

10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69).

Here we use D12,D13 as the softserial port. You need to connect the wires in the following ways and modify the code as follows.



```

FONA3G_setbaud $

#include "Adafruit_FONA.h"

#define FONA_TX 12
#define FONA_RX 13
#define FONA_RST 9

// this is a large buffer for replies
char replybuffer[255];

// We default to using software serial. If you want to use hardware serial
// (because softserial isnt supported) comment out the following three lines
// and uncomment the HardwareSerial line
#include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;

```

How to use

Install Library (Here use Arduino 1.6.9 version)

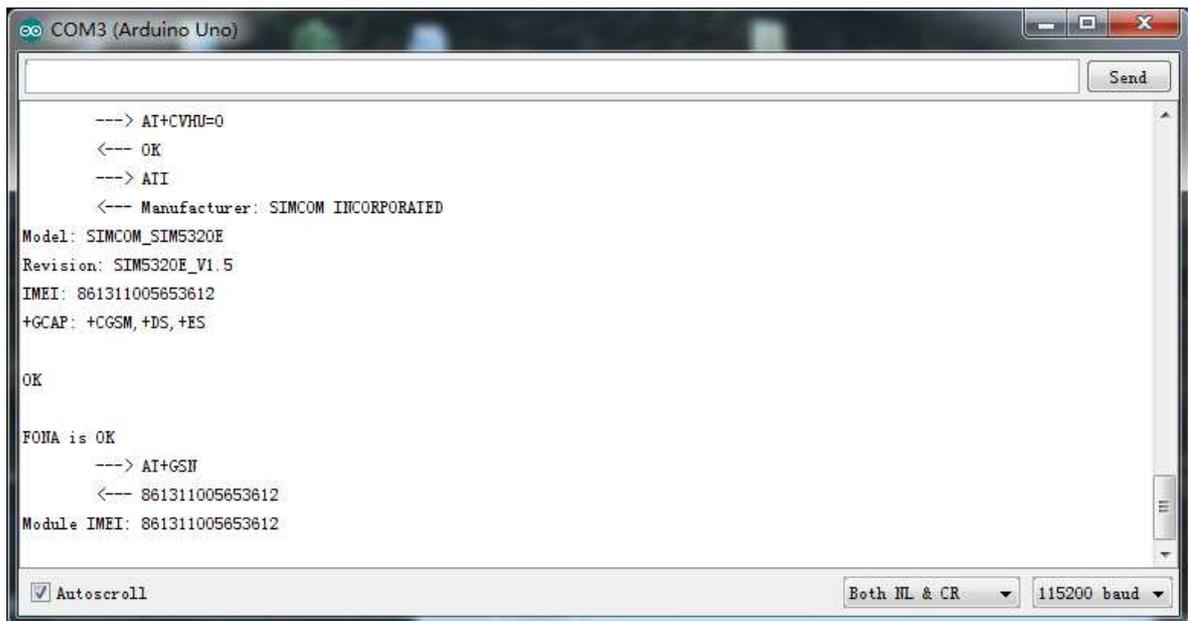
Before reading sensor data, you need to Install Library. “Classic” installation of the library is as follows:

1. Download Arduino Library and unzip it.
2. Check whether the Adafruit_FONA folder contains Adafruit_FONA.cpp and Adafruit_FONA.h. Then place it alongside your other Arduino libraries, typically in your (home folder)/Documents/Arduino/Libraries folder.
3. Re-start the Arduino IDE if it is currently running.

3G Baud Adjustment

If you have a FONA 3G, the first time you use it you may need to run the FONA3G_setBaud example to set the baud rate manually to 4800bps rather than 115200.

1. Put the 3G shield on the Arduino board(Here we use Arduino UNO), then plug in the USB cable from your computer to the Arduino board.
2. We'll use software serial to talk to the module, we choose TX—2,RX—3 on the 3G shield. And press the small POWER button for about 4 seconds to power on.
3. Select your board type and serial port from the Tools menu, open up File->Examples->Adafruit_FONA->FONA3G_setbaud and upload to your Arduino wired up to the module.
4. Once finishing uploading to your Arduino, open up the serial console at 115200 baud speed to begin the tester sketch. Wait a minute and When the serial port monitor prints IMEI, then the baud rate setting completes.



```

COM3 (Arduino Uno)
---> AT+CVHU=0
<--- OK
---> ATII
<--- Manufacturer: SIMCOM INCORPORATED
Model: SIMCOM_SIM5320E
Revision: SIM5320E_V1.5
IMEI: 861311005653612
+GCAP: +CGSM,+DS,+ES

OK

FONA is OK
---> AT+GSII
<--- 861311005653612
Module IMEI: 861311005653612
  
```

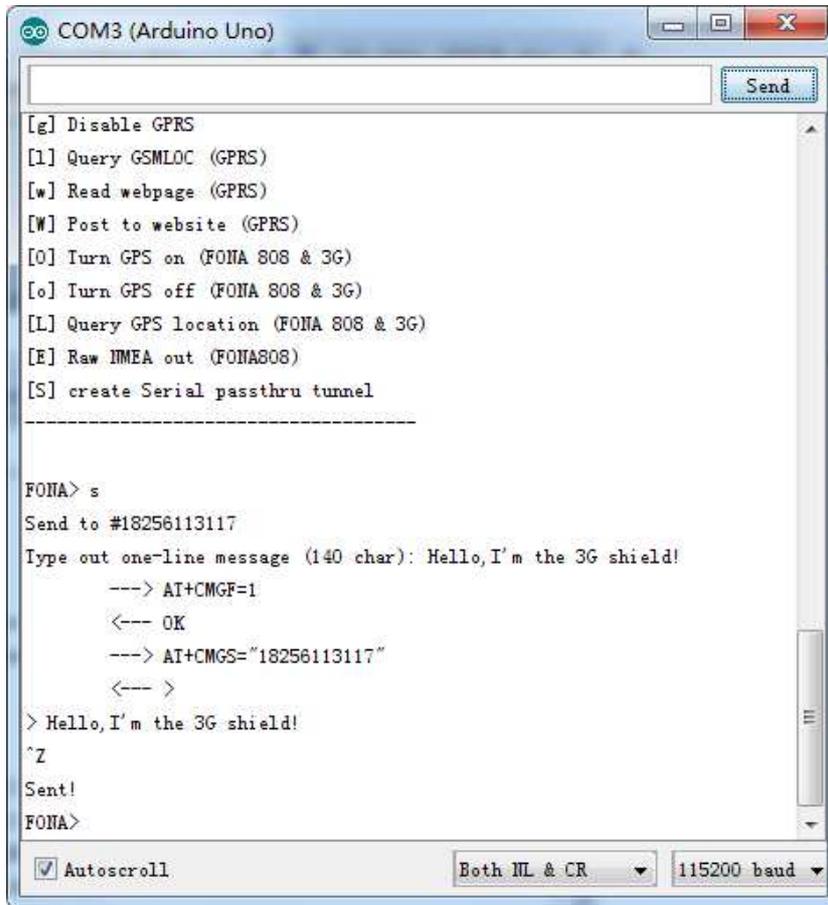
Send SMS

In order to use the phone parts of 3G shield you will need a 2G or 3G Mini SIM card. Luckily, nearly any cell phone shop can sell you a SIM card. You can get a pre-paid or post-paid SIM but we kinda like the pre-paid kind.

Note: Micro SIMs won't fit - so make sure it's a "Mini" SIM. Sometimes these are just called plain "SIM" cards since the huge-size SIMs are rarely used. Mini SIMs are 1" x 0.6" / 25mm x 15mm. It is by far the most common size.

1. Insert a Mini SIM card in the SIM card holder on the back, and connect the GSM antenna.
2. Put the 3G shield on the Arduino board, then plug in the USB cable from your computer to the Arduino board.
3. We'll use software serial to talk to the module, and we choose TX—2,RX—3 on the 3G shield. And press the small POWER button for about 4 seconds to power on.
4. Select your board type and serial port from the Tools menu, Open up File->Examples->Adafruit_FONA->FONAtest and upload to your Arduino wired up to the module.
5. Once finishing uploading to your Arduino, open up the serial console at 115200 baud speed to begin the tester sketch.
6. Wait a minute and enter the "s" in the send column and click "send". Serial monitor

will print output “Send to #”, then enter the phone number you want to send and click “send”. Finally enter the message you want to send and click ”send”.



```

COM3 (Arduino Uno)
[ ] Send
[g] Disable GPRS
[l] Query GSMLOC (GPRS)
[w] Read webpage (GPRS)
[W] Post to website (GPRS)
[O] Turn GPS on (FONA 808 & 3G)
[o] Turn GPS off (FONA 808 & 3G)
[L] Query GPS location (FONA 808 & 3G)
[E] Raw IMEA out (FONAS08)
[S] create Serial passthru tunnel
-----

FONA> s
Send to #18256113117
Type out one-line message (140 char): Hello,I'm the 3G shield!
    ---> AT+CMGF=1
    <--- OK
    ---> AT+CMGS="18256113117"
    <--- >
> Hello,I'm the 3G shield!
^Z
Sent!
FONA>
Autoscroll Both NL & CR 115200 baud
  
```

Note: Make sure you also have Both NL & CR for the serial command sender option. This means when you send data to the Arduino via the console, it will put a newline/return at the end.

GPS

GPS does work without a SIM, however, in that case you cannot send or receive texts, calls, etc. And we can get a fix with a passive antenna but it takes a long time. So we strongly recommend using an active antenna with the GPS.

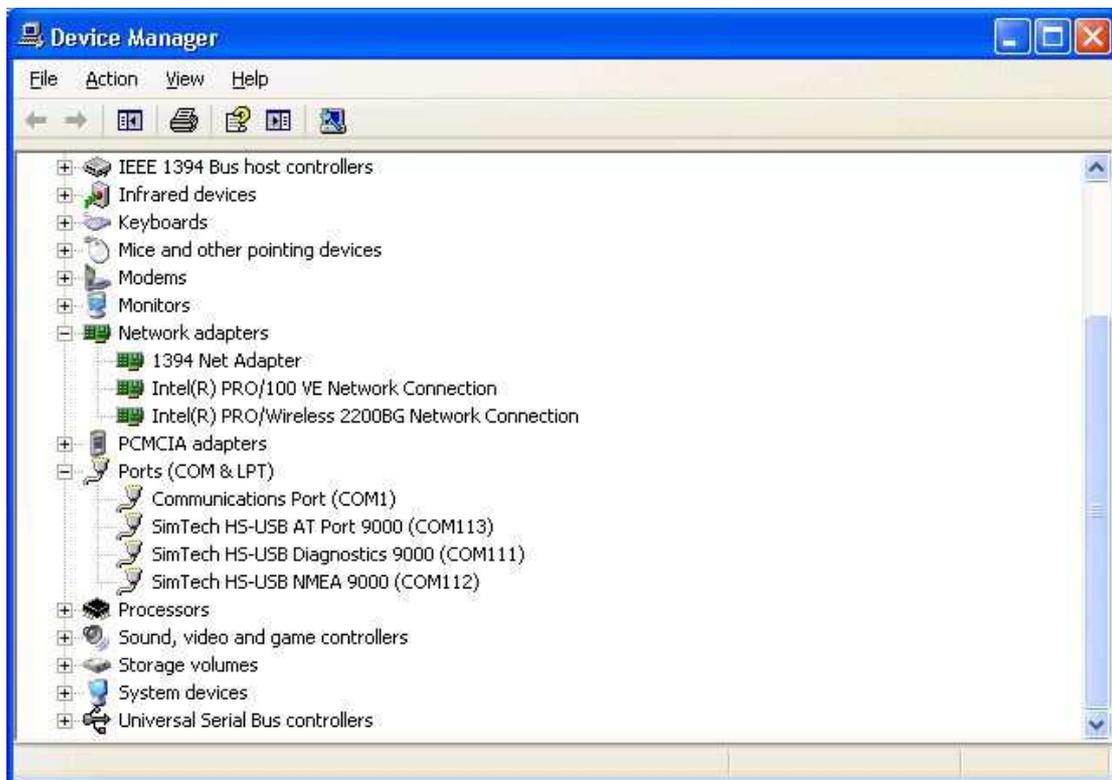
Note: For more rapid and accurate positioning, please put the GPS antenna outside or around the place where there is no obstacle to prevent it from locating the satellite.

1. Connect the GPS antenna to the 3G shield and put it on the Arduino board, then plug in the micro USB cable from your computer to the 3G shield.

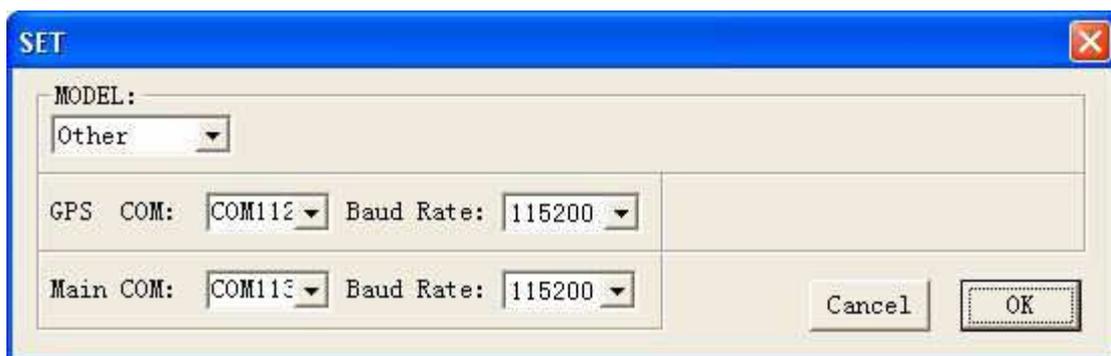
2. Power the Arduino board with a 9VDC power supply and press the small POWER button for 4 seconds to power on. Then the computer will recognize a multi-type USB connection. If you download the drive for the first time, please choose to install the appropriate driver according to your computer system.

Open device manager there are 4 devices created:

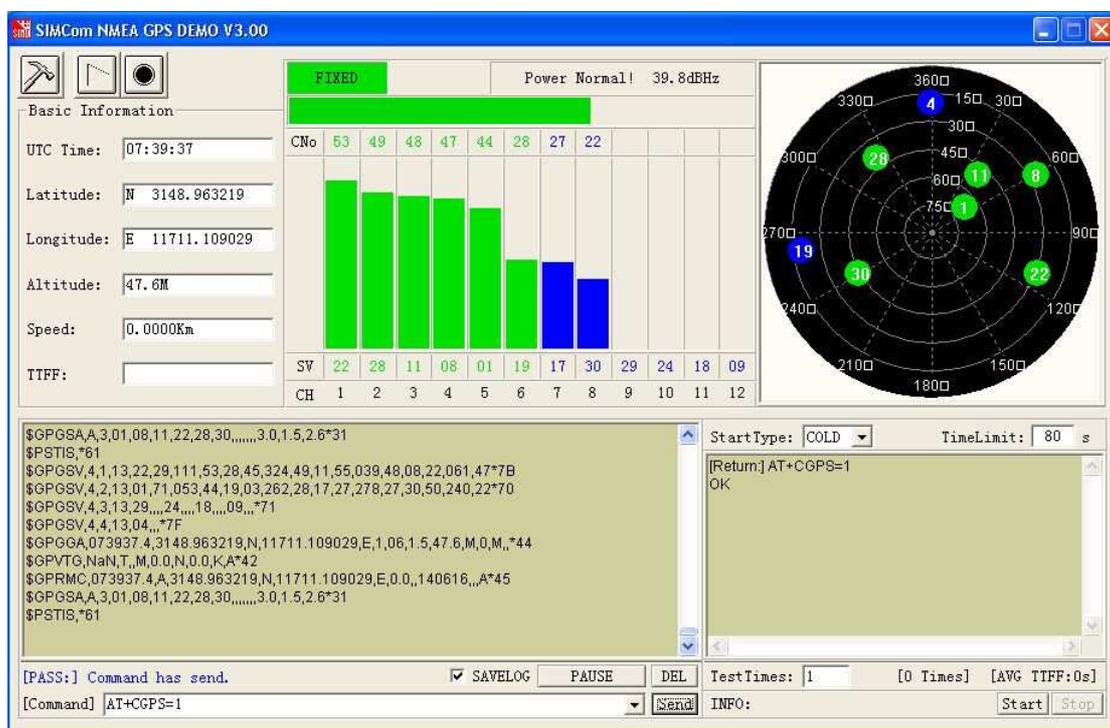
- A communication port
- A Sim Tech HS-USB diagnostics 9000
- A Sim Tech HS-USB NMEA 9000 (this is the raw GPS NMEA output)
- A Sim Tech HS-USB AT port 9000 (this is how you can send commands to the module!)



3. Download the SIMCom NMEA GPS DEMO and open it, click  to set the corresponding port number (Here GPS COM is Sim Tech HS-USB NMEA 9000 and Main COM is Sim Tech HS-USB AT port 9000). Please choose 115200 for Baud Rate and choose “Other” for MODEL .



4. Enter the “AT+CGPS=1” in the command column and click “send”, then after a while you can view the longitude, latitude, altitude etc. of the current location.

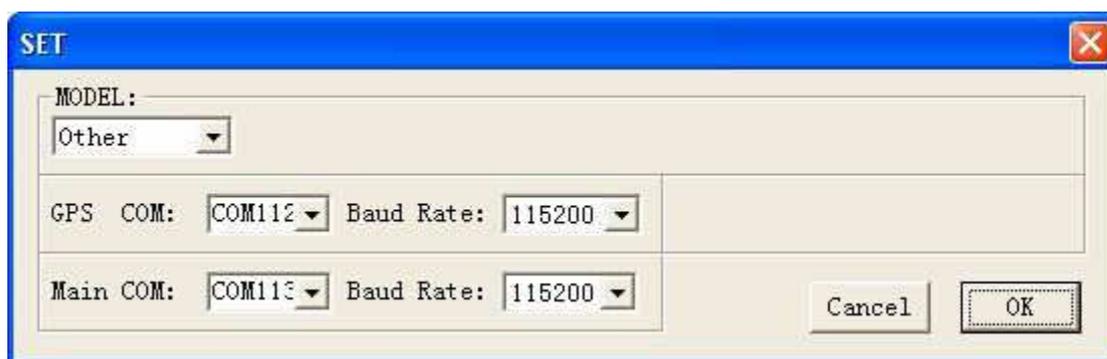


Query registration network

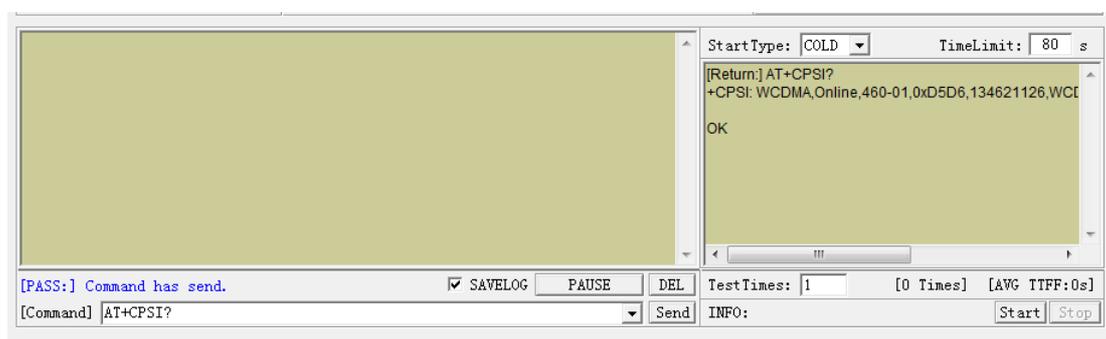
If you want to query the current registered network ,you can get it by sending AT commands. Here are the steps:

1. Insert a Mini SIM card in the SIM card holder on the back, and connect the GSM antenna.
2. Put the 3G shield on the Arduino board, then plug in the USB cable from your computer to the Arduino board and plug in the micro USB cable from your computer to the 3G shield.

3. open the SIMCom NMEA GPS DEMO and click  to set the corresponding port number (Here GPS COM is Sim Tech HS-USB NMEA 9000 and Main COM is Sim Tech HS-USB AT port 9000). Please choose 115200 for Baud Rate and choose “Other” for MODEL .



4. Enter the “AT+CPSI?” in the command column and click “send”, then in the lower right corner of the interface you will see the following information.



+CPSI:WCDMA,Online,460-01,0xD5D6,134621126,WCDMAIMT

2000,334,10713,0,7.5,85,21,30,500

If it returns the above information then it indicates the currently registered network is WCDMA.

+CPSI: GSM,Online,460-01 0x5502,23581,114 EGSM 900,-69,0,27-27

If it returns the above information then it indicates the currently registered network is GSM.