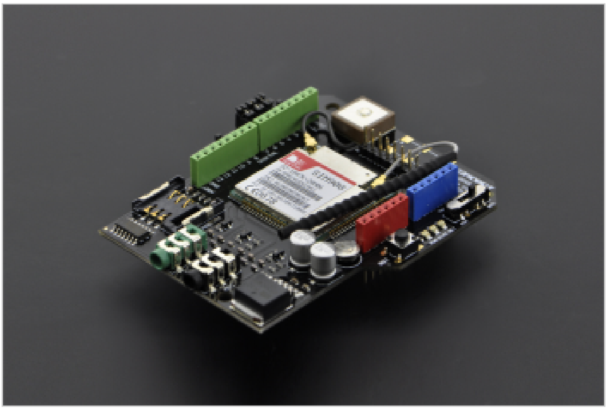
GPS/GPRS/GSM Module V3.0 (SKU:TEL0051)

From Robot Wiki



GPS/GPRS/GSM Module V3.0 (SKU:TEL0051)

Contents

Introduction

This is a GPS/GPRS/GSM shield from DFRobot. This shield with a Quad-band GSM/GPRS engine works on frequencies EGSM 900MHz/DCS 1800MHz and GSM850 MHz/PCS 1900MHz. It also supports GPS technology for satellite navigation. It's possible for your robot and control system to send messages and use the GSM network.

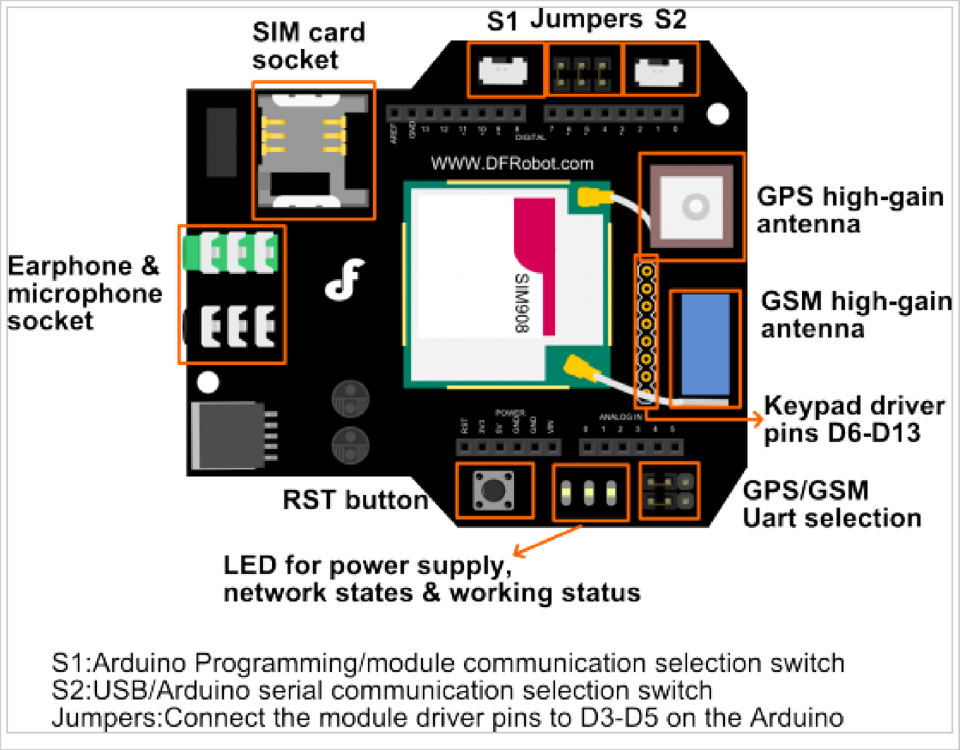
It is controlled via AT commands(GSM07.07 ,07.05 and SIMCOM enhanced AT Commands). And the design of this shield allows you to drive the GSM & GPS function directly with the computer and the Arduino Board. It includes a high-gain SMD antenna for GPS & GSM.

This GPS/GPRS/GSM shield uses an embedded SIM908 chip from SIMCom.Featuring an industry-standard interface and GPS function, the combination of both technologies allows goods, vehicles and people to be tracked seamlessly at any location and anytime with signal coverage.

Specification

- Power supply: 6-12v
- Low power consumption (100mA@7v - GSM mode)
- Quad-Band 850/900/1800/1900MHz
- GPRS multi-slot class 10
- Support GPS technology for satellite navigation
- Embeded high-gain SMD antennas for GPS & GSM
- Directly support 4*4 button pad
- USB/Arduino control switch
- Board Surface:Immersion Gold
- Size: 81x70mm

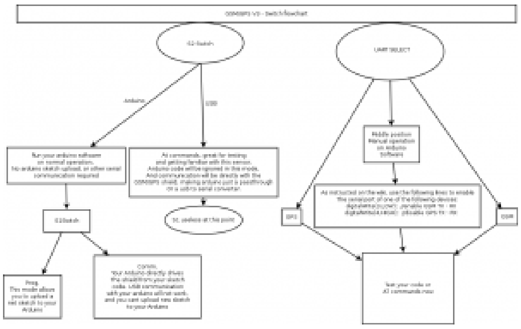
Pin Out


SIM card socket
S1 Jumpers S2
WWW.DFRobot.com
SIM908
GPS high-gain antenna
Earphone & microphone socket
GSM high-gain antenna
Keypad driver pins D6-D13
RST button
GPS/GSM Uart selection
LED for power supply, network states & working status
S1:Arduino Programming/module communication selection switch
S2:USB/Arduino serial communication selection switch
Jumpers:Connect the module driver pins to D3-D5 on the Arduino

Module driver pin jumpers: Applying the Module Pin Jumpers(J10-J12) will allocate Pin 3,4,5 for driving the module. Removing the jumpers will release the above Pins, and you could connect the driver pins to the other digital pins on your board to avoid the pin conflict.Read more (http://www.dfrobot.com/forum/index.php?topic=17186.msg21374#msg21374)

> **NOTE**: Two jumper caps of GPS/GSM UART SELECTION have been changed to a switch."Take off the jumper caps" do the same function of "slid the switch in the middle".

More details about switches:



- **Switch S1:** PC upload program to Arduino board/PC communicates with GPS/GPRS/GSM Module(Arduino programming/module communication).
- **Switch S2:** GPS/GPRS/GSM Module directly connects with PC through USB port or module communicates with Arduino board, which communicates with PC(USB/Arduino serial communication).

Tutorial

Hardware:

- Arduino Uno
- GPS/GPRS/GSM Module V3.0
- SIM Card

- Earphone & Microphone
- External power supply via the power jack

NOTE: It is recommended you to supply 7-12V@1A power via the power jack. When using the GSM mode, the module requires an external power. But the power consumption is not high,just 200mA@7v, when calling.

How to drive the GSM Mode via USB port

1. If your module is enabled, the indicator **stat** LED will light up, this means that the module is running correctly. The LED marked "NET" will be lit when network communication has being established.
2. Send the AT commands to the module by using Coolterm (http://freeware.the-meiers.org/CoolTermWin.zip)(or use the Arduino serial monitor).

NOTE: If you want to program the Arduino, please disconnect the coolterm to release the communication port.

GSM mode & GPS mode Selection

Except using UART selection jumper caps(old version) or switch(latest version), you could switch GSM and GPS function with the IO pins also. The UART selection is located on the bottom right near the PWR STAT and NET LEDs. Remove the jumper caps connected for hardware UART selection(old version) or set the switch in middle position(latest version) in order to control the GSM / GPS by software.

- **Enable GPS mode & disable GSM mode:**

?

```
digitalWrite(4,LOW);//Enable GPS mode
digitalWrite(3,HIGH);//Disable GSM mode
```

- **Enable GSM mode & disable GPS mode:**

?

```
digitalWrite(3,LOW);//Enable GSM mode
digitalWrite(4,HIGH);//Disable GPS mode
```

NOTE: You could control GPS through the GSM AT commands, without requiring to enable both independently. This way you can let the GPS enabled while using GSM network. Thus not triggering a GPS reset.

Network indication

State	SIM908 behavior
Off	SIM908 is not running
64ms On/ 800ms Off	SIM908 not registered the network
64ms On/ 3,000ms Off	SIM908 registered to the network
64ms On/ 300ms Off	PPS GPRS communication is established

NOTE: Following the steps included in the sketch below first!

?

```
// Product name: GPS/GPRS/GSM Module V3.0
// # Product SKU : TEL0051
// # Version     : 0.1

// # Description:
// # The sketch for driving the gsm mode via the USB interface

// # Steps:
// #        1. Turn the S1 switch to the Prog(right side)
// #        2. Turn the S2 switch to the USB side(left side)
// #        3. Set the UART select switch to middle one.
// #        4. Upload the sketch to the Arduino board(Make sure turn off other Serial
monitor )
// #        5. Turn the S1 switch to the comm(left side)
// #        6. RST the board

// #        wiki link-
http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)

void setup()
{
  //Init the driver pins for GSM function
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  //Output GSM Timing
  digitalWrite(5, HIGH);
  delay(2000);
  digitalWrite(5, LOW);
}
void loop()
{
  // Use these commands instead of the hardware switch 'UART select' in order to enable
each mode
  // If you want to use both GMS and GPS. enable the required one in your code and
disable the other one for each access.
  digitalWrite(3, LOW); //enable GSM TX、RX
  delay(500);
  digitalWrite(4, HIGH); //disable GPS TX、RX
  delay(500);
}
```

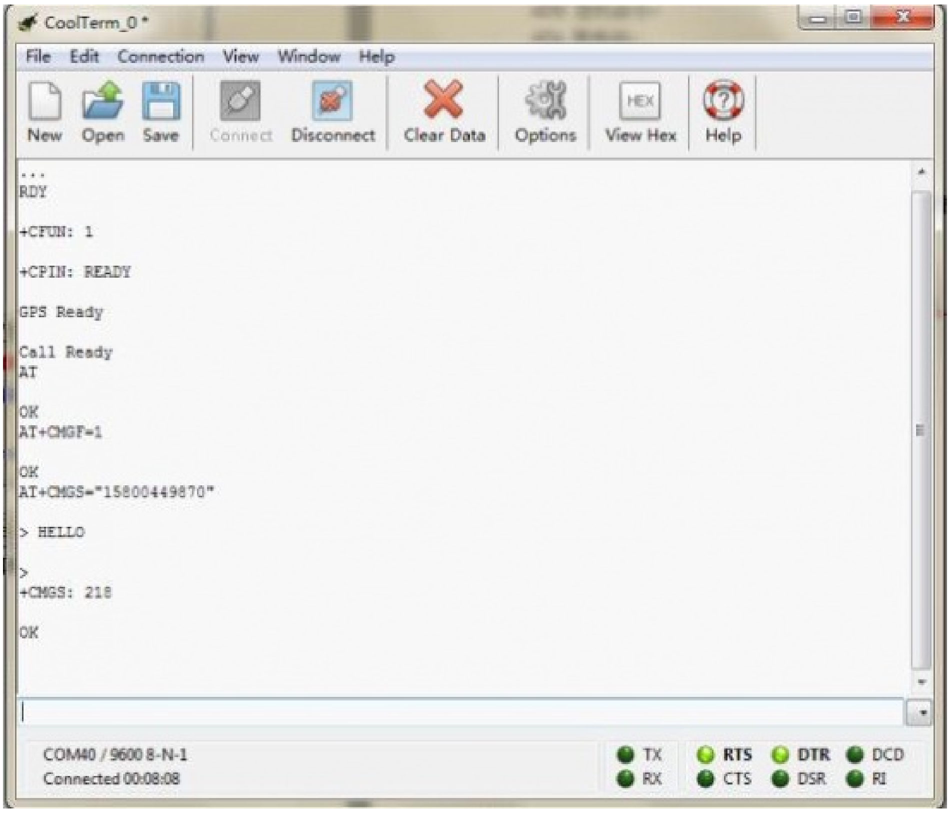
How to Send a message

Steps:

1. Send:AT
2. Send:AT+CMGF=1 (set the message to text format)
3. Send:AT+CMGS="XXXXX" （xxxx is the number of receiver ）
4. After you see '>' then type the message you want to send
5. press 'ctrl+z'(If you want to cancel, you can press ESC) Note that this step depends on the Serial software used and its configuration!

NOTE: Please do some settings as Recommended settings before connection. If you have trouble sending Ctrl+Z, please refers to Ways to send Ctrl +Z in Coolterm

Then you will see



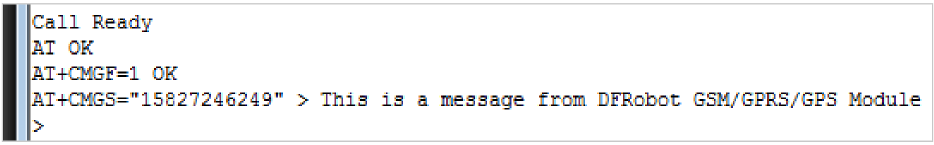
```
...
RDY

+CFUN: 1

+CPIN: READY

GPS Ready

Call Ready
AT

OK
AT+CMGF=1

OK
AT+CMGS="15800449870"

> HELLO

>
+CMGS: 218

OK
```

COM40 / 9600 8-N-1
Connected 00:08:08

After several seconds,the receiver will get a message from this shield

Ways to send Ctrl +Z in Coolterm

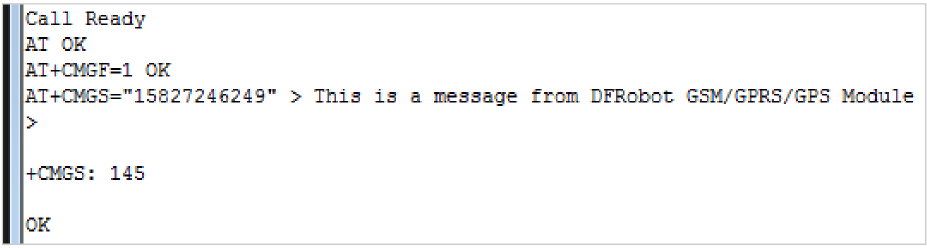
1. After typing the message text, press enter key, it will show:



```
Call Ready
AT OK
AT+CMGF=1 OK
AT+CMGS="15827246249" > This is a message from DFRobot GSM/GPRS/GPS Module
>
```

'Enter' after text finished

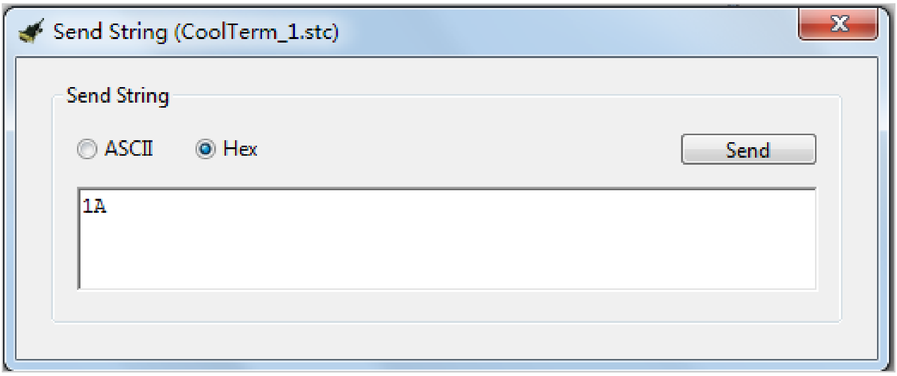
Then, in the input area, pressing Ctrl+Z(note this key shortcut depends on your version and configuration) will send out the single CTRL character successfully as below:



```
Call Ready
AT OK
AT+CMGF=1 OK
AT+CMGS="15827246249" > This is a message from DFRobot GSM/GPRS/GPS Module
>

+CMGS: 145

OK
```

After press Ctrl+Z

2. Ctrl characters can also be sent out in their hexadecimal format (this avoids problems with Coolterm version and configurations, this method can be used with other Serial software), in which hexadeciman '1A' is the value of Ctrl+Z:



```
1A
```

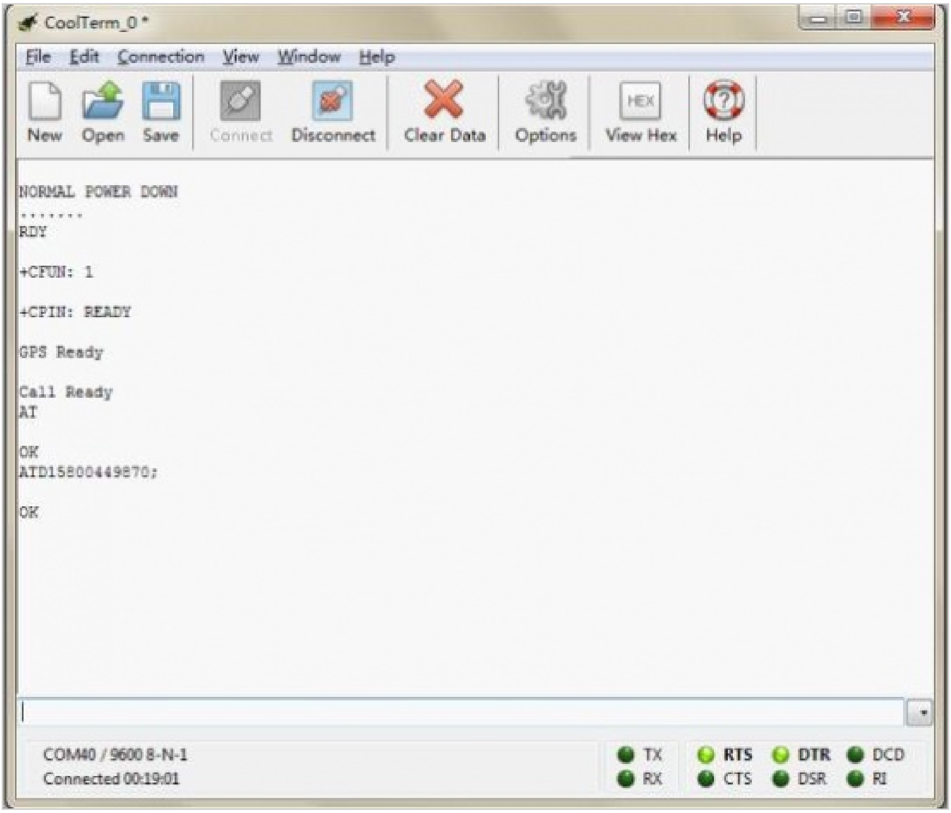
Sending Ctrl+Z with in hexadecimal format

How to Make a phone call

Steps:

1. Send:AT
2. Send:ATDXXXXXXXX;（xxxxxxx is the number of receiver,don't forget the ;）

Then you will see



```
NORMAL POWER DOWN
.......
RDY

+CFUN: 1

+CPIN: READY

GPS Ready

Call Ready
AT

OK
ATD15800449870;

OK
```

COM40 / 9600 8-N-1
Connected 00:19:01

After several seconds,the receiver will get a phone call from this shield

Some AT commands

- ATH : Hang up the phone
- ATA : Answer the phone

How to drive the GPS Mode via USB port

It is better to take the module outdoor, because you know GPS signal is not very well indoor.

?

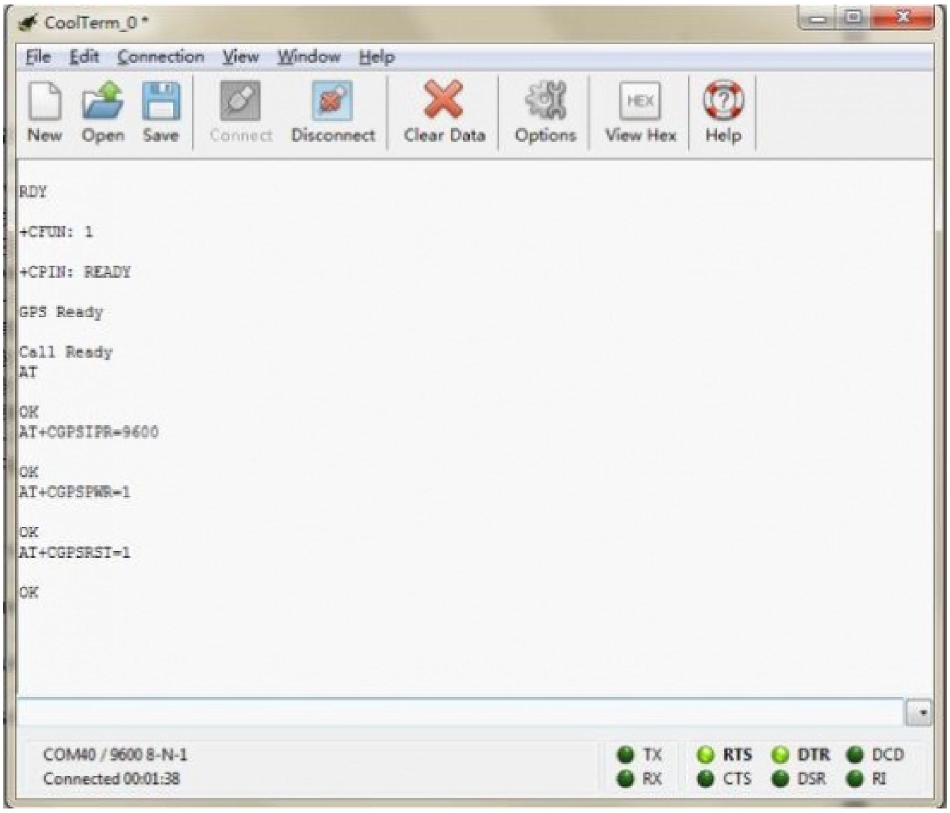
```
// Product name: GPS/GPRS/GSM Module V3.0
// # Product SKU : TEL0051
// # Version     : 0.1

// # Description:
// # The sketch for driving the gps mode via the USB interface

// # Steps:
// #        1. Turn the S1 switch to the Prog(right side)
// #        2. Turn the S2 switch to the USB side(left side)
// #        3. Set the UART select switch to middle one.
// #        4. Upload the sketch to the Arduino board(Make sure turn off other Serial
monitor )
// #        5. Turn the S1 switch to the comm(left side)
// #        6. RST the board until the START led is on

// #        wiki link-
http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)

void setup()
 {
   //Init the driver pins for GSM function
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
  //Output GSM Timing
    digitalWrite(5,HIGH);
    delay(1500);
    digitalWrite(5,LOW);
 }
 void loop()
 {
    digitalWrite(3,HIGH);//disable GSM TX、RX
    digitalWrite(4,HIGH);//disable GPS TX、RX
 }
```

After uploading the sample code,you could follow these steps to get GPS data:

Steps:
1 Turn the UART switch to **GSM** side
2 Input 4 AT command

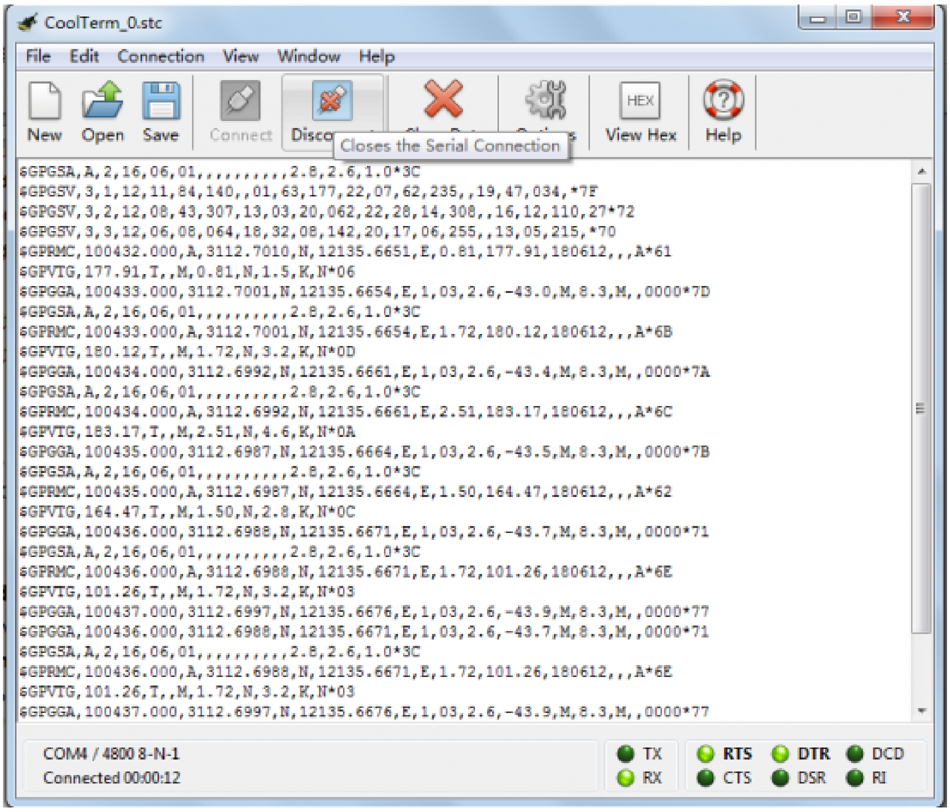
- Send:AT
- Send:AT+CGPSIPR=9600 (set the baud rate)
- Send:AT+CGPSPWR=1 （turn on GPS power supply）
- Send:AT+CGPSRST=1 （reset GPS in autonomy mode）

Then you can see


CoolTerm_0 *
File Edit Connection View Window Help
New Open Save Connect Disconnect Clear Data Options View Hex Help
COM40 / 9600 8-N-1
Connected 00:01:38
TX RX RTS CTS DTR DSR DCD RI

```
RDY

+CFUN: 1

+CPIN: READY

GPS Ready

Call Ready
AT

OK
AT+CGPSIPR=9600

OK
AT+CGPSPWR=1

OK
AT+CGPSRST=1

OK
```

3 Then turn the UART switch back to the **GPS** side.


CoolTerm_0.stc
File Edit Connection View Window Help
New Open Save Connect Disco View Hex Help
Closes the Serial Connection
COM4 / 4800 8-N-1
Connected 00:00:12
TX RX RTS CTS DTR DSR DCD RI

```
$GPGSA,A,2,16,06,01,,,,,,,,,,2.8,2.6,1.0*3C
$GPGSV,3,1,12,11,84,140,,01,63,177,22,07,62,235,,19,47,034,*7F
$GPGSV,3,2,12,08,43,307,13,03,20,062,22,28,14,308,,16,12,110,27*72
$GPGSV,3,3,12,06,08,064,18,32,08,142,20,17,06,255,,13,05,215,*70
$GPRMC,100432.000,A,3112.7010,N,12135.6651,E,0.81,177.91,180612,,,A*61
$GPVTG,177.91,T,,M,0.81,N,1.5,K,N*06
$GPGGA,100433.000,3112.7001,N,12135.6654,E,1,03,2.6,-43.0,M,8.3,M,,0000*7D
$GPGSA,A,2,16,06,01,,,,,,,,,,2.8,2.6,1.0*3C
$GPRMC,100433.000,A,3112.7001,N,12135.6654,E,1.72,180.12,180612,,,A*6B
$GPVTG,180.12,T,,M,1.72,N,3.2,K,N*0D
$GPGGA,100434.000,3112.6992,N,12135.6661,E,1,03,2.6,-43.4,M,8.3,M,,0000*7A
$GPGSA,A,2,16,06,01,,,,,,,,,,2.8,2.6,1.0*3C
$GPRMC,100434.000,A,3112.6992,N,12135.6661,E,2.51,183.17,180612,,,A*6C
$GPVTG,183.17,T,,M,2.51,N,4.6,K,N*0A
$GPGGA,100435.000,3112.6987,N,12135.6664,E,1,03,2.6,-43.5,M,8.3,M,,0000*7B
$GPGSA,A,2,16,06,01,,,,,,,,,,2.8,2.6,1.0*3C
$GPRMC,100435.000,A,3112.6987,N,12135.6664,E,1.50,164.47,180612,,,A*62
$GPVTG,164.47,T,,M,1.50,N,2.8,K,N*0C
$GPGGA,100436.000,3112.6988,N,12135.6671,E,1,03,2.6,-43.7,M,8.3,M,,0000*71
$GPGSA,A,2,16,06,01,,,,,,,,,,2.8,2.6,1.0*3C
$GPRMC,100436.000,A,3112.6988,N,12135.6671,E,1.72,101.26,180612,,,A*6E
$GPVTG,101.26,T,,M,1.72,N,3.2,K,N*03
$GPGGA,100437.000,3112.6997,N,12135.6676,E,1,03,2.6,-43.9,M,8.3,M,,0000*77
$GPGGA,100436.000,3112.6988,N,12135.6671,E,1,03,2.6,-43.7,M,8.3,M,,0000*71
$GPGSA,A,2,16,06,01,,,,,,,,,,2.8,2.6,1.0*3C
$GPRMC,100436.000,A,3112.6988,N,12135.6671,E,1.72,101.26,180612,,,A*6E
$GPVTG,101.26,T,,M,1.72,N,3.2,K,N*03
$GPGGA,100437.000,3112.6997,N,12135.6676,E,1,03,2.6,-43.9,M,8.3,M,,0000*77
```

For location of the data received, please refer to Location Mapping (GPRMC) (http://www.sanav.com/gps_tracking/webtrac-4/maps/MapLocationGPRMC.aspx)

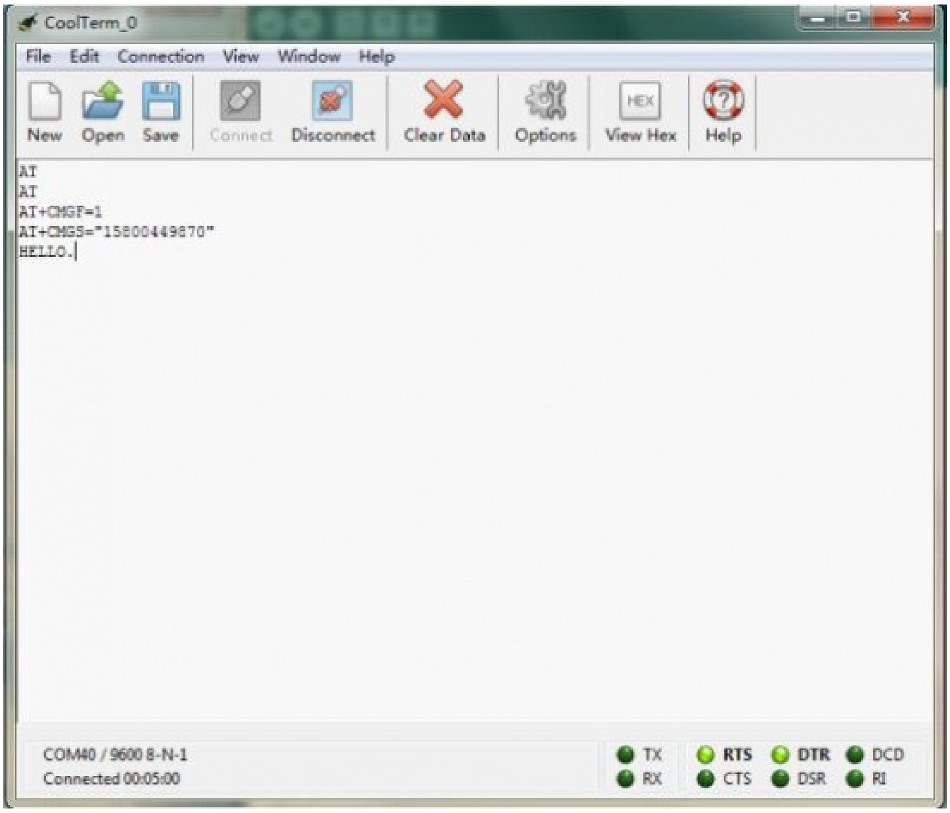
How to drive the GSM Mode via Arduino board

How to Send a message

?

```
// Product name: GPS/GPRS/GSM Module V3.0
// # Product SKU : TEL0051
// # Version     : 0.1

// # Description:
// # The sketch for driving the gsm mode via the Arduino board

// # Steps:
// #        1. Turn the S1 switch to the Prog(right side)
// #        2. Turn the S2 switch to the Arduino side(left side)
// #        3. Set the UART select switch to middle one.
// #        4. Upload the sketch to the Arduino board
// #        5. Turn the S1 switch to the comm(left side)
// #        6. RST the board

// #        wiki link-
http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)

byte gsmDriverPin[3] = {
  3,4,5};//The default digital driver pins for the GSM and GPS mode
//If you want to change the digital driver pins
//or you have a conflict with D3~D5 on Arduino board,
//you can remove the J10~J12 jumpers to reconnect other driver pins for the module!
void setup()
{
  //Init the driver pins for GSM function
  for(int i = 0 ; i < 3; i++){
    pinMode(gsmDriverPin[i],OUTPUT);
  }
  digitalWrite(5,HIGH);//Output GSM Timing
  delay(1500);
  digitalWrite(5,LOW);
  digitalWrite(3,LOW);//Enable the GSM mode
  digitalWrite(4,HIGH);//Disable the GPS mode
  delay(2000);
  Serial.begin(9600); //set the baud rate
  delay(5000);//call ready
  delay(5000);
  delay(5000);
}

void loop()
{
  Serial.println("AT"); //Send AT command
  delay(2000);
  Serial.println("AT");
  delay(2000);
  //Send message
  Serial.println("AT+CMGF=1");
  delay(1000);
  Serial.println("AT+CMGS=\"15800449871\"");//Change the receiver phone number
  delay(1000);
  Serial.print("HELLO");//the message you want to send
  delay(1000);
  Serial.write(26);
  while(1);
}
```

You can see:



```
AT
AT
AT+CMGF=1
AT+CMGS="15800449870"
HELLO.
```

COM40 / 9600 8-N-1
Connected 00:05:00

After several seconds,the receiver will get a message from this shield

How to Control your Arduino via SMS

Follow the forum discussion with more coding examples and options on this link Click Me！(http://www.dfrobot.com/forum/index.php?topic=945.msg4514#msg4514)

?

```
// Product name: GPS/GPRS/GSM Module V3.0
// # Product SKU : TEL0051

// # Description:
// # The sketch for controling the GSM/GPRS/GPS module via SMS.
// # Steps:
// #        1. Turn the S1 switch to the Prog(right side)
// #        2. Turn the S2 switch to the USB side(left side)
// #        3. Set the UART select switch to middle one.
// #        4. Upload the sketch to the Arduino board(Make sure turn off other Serial monitor )
// #        5. Turn the S1 switch to the comm(left side)
// #        6. Turn the S2 switch to the Arduino(right side)
// #        7. RST the board until the START led is on(make sure you have >6V power supply)
// #        8. Plug the long side of LED into pin 13 and short side into GND
// #        9. Start sending "LH" and "LL" to your board to turn LED on and off.

/*
 *  created:    2013-11-14
 *  by:     Grey
 *  Version:    0.3
 *  Attention: if you send the wrong SMS command to the module, just need to press RST.
 *  This version can't watch the module status via the serial monitor, it only display the Arduino command.
 *  If you want to watch the status,use the SoftwareSerial or the board with another serial port plese.
 */

byte gsmDriverPin[3] = {
  3,4,5};//The default digital driver pins for the GSM and GPS mode
//If you want to change the digital driver pins
```

```
//or you have a conflict with D3~D5 on Arduino board,
//you can remove the J10~J12 jumpers to reconnect other driver pins for the module!
int ledpin = 13;
char inchar;
void setup()
{
  //Init the driver pins for GSM function
  for(int i = 0 ; i < 3; i++){
    pinMode(gsmDriverPin[i],OUTPUT);
  }
  pinMode(ledpin,OUTPUT);
  Serial.begin(9600);                                      //set the baud rate
  digitalWrite(5,HIGH);                                     //Output GSM Timing
  delay(1500);
  digitalWrite(5,LOW);
  digitalWrite(3,LOW);                                      //Enable the GSM mode
  digitalWrite(4,HIGH);                                     //Disable the GPS mode
  delay(2000);
  delay(5000);                                              //call ready
  delay(5000);
  Serial.println("AT+CMGD=1,4");                           //Delete all SMS in box
}

void loop()
{
  if(Serial.available()>0)
  {
    inchar=Serial.read();
    if(inchar=='T')
    {
      delay(10);
      inchar=Serial.read();
      if (inchar=='I')                                     //When the GSM module get
the message, it will display the sign '+CMTI "SM", 1' in the serial port
      {
        delay(10);
        Serial.println("AT+CMGR=1");                       //When Arduino read the
sign, send the "read" AT command to the module
        delay(10);
      }
    }
    else if (inchar=='L')
    {
      delay(10);
      inchar=Serial.read();
      if (inchar=='H')                                    //Thw SMS("LH") was display
in the Serial port, and Arduino has recognize it.
      {
        delay(10);
        digitalWrite(ledpin,HIGH);                         //Turn on led
        delay(50);
        Serial.println("AT+CMGD=1,4");                    //Delete all message
        delay(500);
      }
      if (inchar=='L')                                   //Thw SMS("LH") was display
in the Serial port, and Arduino has recognize it.
      {
        delay(10);
        digitalWrite(ledpin,LOW);                         //Turn off led
        delay(50);
        Serial.println("AT+CMGD=1,4");                   //Delete all message
        delay(500);
      }
    }
  }
}
```

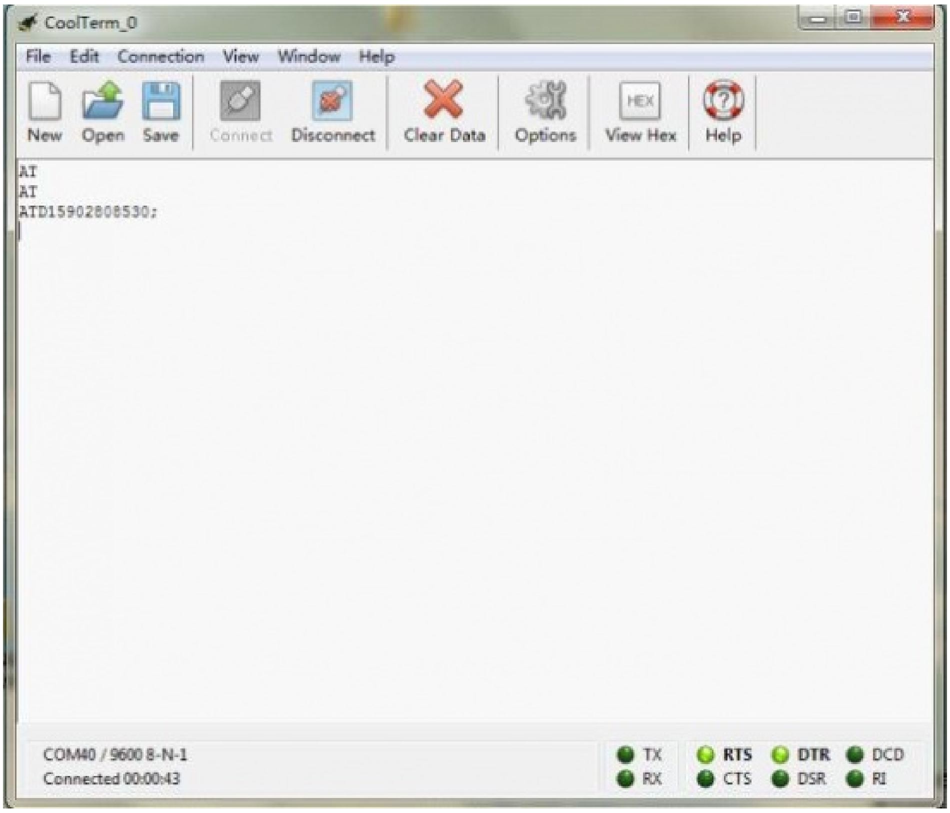
When you send the SMS "LH" to the module, it WILL turn led on; when you send the SMS "LL", it will be turned off.

How to Make a phone call

?

```
// Product name: GPS/GPRS/GSM Module V3.0
// # Product SKU : TEL0051
// # Version     : 0.1

// # Description:
// # The sketch for driving the gsm mode via the Arduino board

// # Steps:
// #        1. Turn the S1 switch to the Prog(right side)
// #        2. Turn the S2 switch to the Arduino side(left side)
// #        3. Set the UART select switch to middle one.
// #        4. Upload the sketch to the Arduino board
// #        5. Turn the S1 switch to the comm(left side)
// #        6. RST the board

// #        wiki link-
http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)

byte gsmDriverPin[3] = {
  3,4,5};//The default digital driver pins for the GSM and GPS mode
//If you want to change the digital driver pins
//or you have a conflict with D3~D5 on Arduino board,
//you can remove the J10~J12 jumpers to reconnect other driver pins for the module!

void setup()
{
 //Init the driver pins for GSM function
  for(int i = 0 ; i < 3; i++){
    pinMode(gsmDriverPin[i],OUTPUT);
  }
  digitalWrite(5,HIGH);//Output GSM Timing
  delay(1500);
  digitalWrite(5,LOW);
  digitalWrite(3,LOW);//Enable the GSM mode
  digitalWrite(4,HIGH);//Disable the GPS mode
  delay(2000);
  Serial.begin(9600); //set the baud rate
  delay(5000);//call ready
  delay(5000);
  delay(5000);
}

void loop()
{
   Serial.println("AT");//Send AT command
   delay(2000);
   Serial.println("AT");
   delay(2000);
  //Make a phone call
   Serial.println("ATD15902808530;");//Change the receiver phone number
   while(1);
}
```

You can see:



```
AT
AT
ATD15902808530;
```

COM40 / 9600 8-N-1
Connected 00:00:43

After several seconds,the receiver will get a phone call from this shield

How to drive the GPS Mode via Arduino board

- S1 - Comm
- S2 - Arduino
- S3 - UART (Middle)

?

```
// Product name: GPS/GPRS/GSM Module V3.0
// # Product SKU : TEL0051
// # Version     : 1.2

// # Description:
// # The sketch for driving the gps mode via the Arduino board

// # Steps:
// #        1. Turn the S1 switch to the Prog(right side)
// #        2. Turn the S2 switch to the Arduino side(left side)
// #        3. Set the UART select switch to middle one.
// #        4. Upload the sketch to the Arduino board
// #        5. Turn the S1 switch to the comm(left side)
// #        6. RST the board

// #        If you get 'inf' values, go outdoors and wait until it is connected.
// #        wiki link-
http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)

double Datatransfer(char *data_buf,char num)//convert the data to the float type
{                                            //*data_buf：the data
array
  double temp=0.0;                           //the number of the right of a decimal
point
  unsigned char i,j;

  if(data_buf[0]=='-')
  {
    i=1;
    //process the data array
    while(data_buf[i]!='.')
```

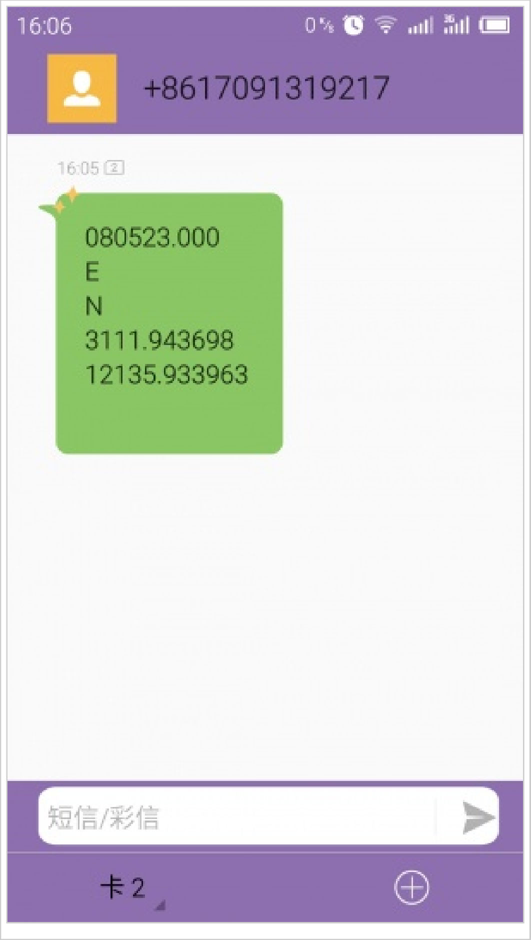
```
      temp=temp*10+(data_buf[i++]-0x30);
    for(j=0;j<num;j++)
      temp=temp*10+(data_buf[++i]-0x30);
    //convert the int type to the float type
    for(j=0;j<num;j++)
      temp=temp/10;
    //convert to the negative numbe
    temp=0-temp;
  }
  else//for the positive number
  {
    i=0;
    while(data_buf[i]!='.')
      temp=temp*10+(data_buf[i++]-0x30);
    for(j=0;j<num;j++)
      temp=temp*10+(data_buf[++i]-0x30);
    for(j=0;j<num;j++)
      temp=temp/10 ;
  }
  return temp;
}

char ID()//Match the ID commands
{
  char i=0;
  char value[6]={
    '$','G','P','G','G','A'    };//match the gps protocol
  char val[6]={
    '0','0','0','0','0','0'    };

  while(1)
  {
    if(Serial.available())
    {
      val[i] = Serial.read();//get the data from the serial interface
      if(val[i]==value[i]) //Match the protocol
      {
        i++;
        if(i==6)
        {
          i=0;
          return 1;//break out after get the command
        }
      }
      else
        i=0;
    }
  }
}

void comma(char num)//get ','
{
  char val;
  char count=0;//count the number of ','

  while(1)
  {
    if(Serial.available())
    {
      val = Serial.read();
      if(val==',')
        count++;
    }
    if(count==num)//if the command is right, run return
      return;
  }

```

```
}
void UTC()//get the UTC data -- the time
{
  char i;
  char time[9]={
    '0','0','0','0','0','0','0','0','0'
  };
  double t=0.0;

  if( ID())//check ID
  {
    comma(1);//remove 1 ','
    //get the datas after headers
    while(1)
    {
      if(Serial.available())
      {
        time[i] = Serial.read();
        i++;
      }
      if(i==9)
      {
        i=0;
        t=Datatransfer(time,2);//convert data
        t=t-30000.00;//convert to the chinese time GMT+8 Time zone
        long time=t;
        int h=time/10000;
        int m=(time/100)%100;
        int s=time%100;

//        if(h>=24)               //UTC+
//        {
//        h-=24;
//        }

         if (h<0)               //UTC-
        {
          h+=24;
        }
        Serial.print(h);
        Serial.print("h");
        Serial.print(m);
        Serial.print("m");
        Serial.print(s);
        Serial.println("s");

        //Serial.println(t);//Print data
        return;
      }
    }
  }
}
void latitude()//get latitude
{
  char i;
  char lat[10]={
    '0','0','0','0','0','0','0','0','0','0'
  };


  if( ID())
  {
    comma(2);
    while(1)
    {
      if(Serial.available())
      {
        lat[i] = Serial.read();
        i++;
      }
      if(i==10)
      {
        i=0;
        Serial.println(Datatransfer(lat,5)/100.0,7);//print latitude
        return;
      }
    }
  }
}
void lat_dir()//get dimensions
{
  char i=0,val;

  if( ID())
  {
    comma(3);
    while(1)
    {
      if(Serial.available())
      {
        val = Serial.read();
        Serial.write(val);
        Serial.println();
        i++;
      }
      if(i==1)
      {
        i=0;
        return;
      }
    }
  }
}
void longitude()//get longitude
{
  char i;
  char lon[11]={
    '0','0','0','0','0','0','0','0','0','0','0'
  };

  if( ID())
  {
    comma(4);
    while(1)
    {
      if(Serial.available())
      {
        lon[i] = Serial.read();
        i++;
      }
      if(i==11)
      {
        i=0;
        Serial.println(Datatransfer(lon,5)/100.0,7);
        return;
      }
    }
  }
}
void lon_dir()//get direction data
{
  char i=0,val;

  if( ID())
  {
    comma(5);
    while(1)
    {
      if(Serial.available())
      {
        val = Serial.read();
        Serial.write(val); //Serial.println(val,BYTE);
        Serial.println();
        i++;
      }
      if(i==1)
      {
        i=0;
        return;
      }
    }
  }
}
void altitude()//get altitude data
{
  char i,flag=0;
  char alt[8]={
    '0','0','0','0','0','0','0','0'
  };

  if( ID())
  {
    comma(9);
    while(1)
    {
      if(Serial.available())
      {
        alt[i] = Serial.read();
        if(alt[i]==',')
          flag=1;
        else
          i++;
      }
      if(flag)
      {
        i=0;
        Serial.println(Datatransfer(alt,1),1);//print altitude data
        return;
      }
    }
  }
}
void setup()
{
  pinMode(3,OUTPUT);//The default digital driver pins for the GSM and GPS mode
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  digitalWrite(5,HIGH);
  delay(1500);
  digitalWrite(5,LOW);

  digitalWrite(3,LOW);//Enable GSM mode
  digitalWrite(4,HIGH);//Disable GPS mode
  delay(2000);
  Serial.begin(9600);
  delay(5000);//GPS ready

  Serial.println("AT");
  delay(2000);
  //turn on GPS power supply
  Serial.println("AT+CGPSPWR=1");
  delay(1000);
  //reset GPS in autonomy mode
  Serial.println("AT+CGPSRST=1");
  delay(1000);

  digitalWrite(4,LOW);//Enable GPS mode
  digitalWrite(3,HIGH);//Disable GSM mode
  delay(2000);

  Serial.println("$GPGGA statement information: ");
}
void loop()
{
  while(1)
  {
    Serial.print("UTC:");
    UTC();
    Serial.print("Lat:");
    latitude();
    Serial.print("Dir:");
    lat_dir();
    Serial.print("Lon:");
    longitude();
    Serial.print("Dir:");
    lon_dir();
    Serial.print("Alt:");
    altitude();
    Serial.println(' ');
    Serial.println(' ');
  }
}
```

GPS Sample Code

This code will send GPS information to you phone via SMS. Please download the library gps_gsm_sim908 (http://www.dfrobot.com/forum/download/file.php?id=94) first.

Upload the sketch to your Arduino board, after about 5-10 minutes according to your GPS signal intensity, you could receive the GPS data like this:



080523.000
E
N
3111.943698
12135.933963

FAQ

Q1. If the module is not working properly, plz check:

A.

- SIM card should be in service
- Switches should in right position
- The external power supply should be 7-12V@1A
- Check signal range, best to be on an area with full coverage.
- Signal for GPS has best performance on a clear direct line of sight with sky, even better with less buildings around.
- GPS needs time to connect to at least 4 satellites to output data.
- AT command tester by one of our contributors. (http://www.dfrobot.com/wiki/index.php/AT_command_tester)

Q2. What's the minutes and seconds of coordinate data?

A. The raw gps information from the serial is in the form: DDmm.mmmmm. e.g.If the raw information is: 5320.12345, then it is: 53 degress and 20.12345 minutes. In order to convert it to decimal coordinates you divide the minutes by 60. i.e. 20.12345 / 60 = .33539 Finally, the decimal result: 53.33539 Forum reading, attached source code that prints google maps friendly coordinates. (http://www.dfrobot.com/forum/viewtopic.php?f=2&t=1528&p=7729#p7693)

Q3. I want to get GPS location when I send a message to the module v3.0. Anyone have solutions or suggestion? Tnx.

A. Forum reading (http://www.dfrobot.com/forum/viewtopic.php?f=8&t=1544&p=7771&hilit=GPS+DATA#p7866)

Q4. How to **send data to a web page**/ URL and after **get the anwser back in my arduino**.

A. Forum reading. (http://www.dfrobot.com/forum/viewtopic.php?f=2&t=1561#p7839)

For any questions/advice/cool ideas to share, please visit **DFRobot Forum** (http://www.dfrobot.com/forum/) or email to techsupport@dfrobot.com

More

- Version history:GPS/GPRS/GSM Module V2.0 (SKU:TEL0051) (http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V2.0_(SKU:TEL0051))

→ Go Shopping GPS/GPRS/GSM Shield V3.0 (Arduino Compatible)(SKU:TEL0051) (http://www.dfrobot.com/index.php?route=product/product&filter_name=gps&product_id=673)
→ Go Shopping DFRobot Distributor List (http://www.dfrobot.com/index.php?route=information/distributorslogo)

Retrieved from "http://www.dfrobot.com/wiki/index.php?title=GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)&oldid=33458"
Categories: Product Manual | TEL Series | Shield

- This page was last modified on 12 January 2016, at 14:19.
- This page has been accessed 97,467 times.