SIM900 Quad-Band GPRS shield with Micro SD card slot

Introduction

This shield is designed based on SIM900 Quad-band GSM/GPRS module(support 850/900/1800/1900MHz network)which can be used throughout the world. An SD card slot also is included. Working with Arduino, you can design your project quickly.

Key Features

SIM900 Features

- Quad-Band 850/ 900/ 1800/ 1900 MHz
- Low power consumption: 1.0mA(sleep mode&BS-PAMFRMS=9)
- Operation temperature: -40°C to +85 °C
- GPRS class 10: max. 85.6 kbps (downlink)
- Embedded TCP/UDP protocol
- RTC backup
- PWM
- ADC

Features of this shield

- Compatible with Arduino
- SD card slot, can be disabled by the jumper
- Two in one Earphone Socket
- Software and Hardware Serial port:Can communicate with Arduino through the Software serial port(D2/D3)or the Hardware serial port(D0/D1)
- FTDI interface. You can use the PC or other host which have an USB port(through FT232RL board) to debug it.
- Battery slot for RTC. This is useful if you want the time of the module doesn't lose when it is Power Off.
- Software or Hardware Power ON/OFF. You can Power On/Off SIM900 through an IO of Arduino or the "PWRKEY" button on the board.

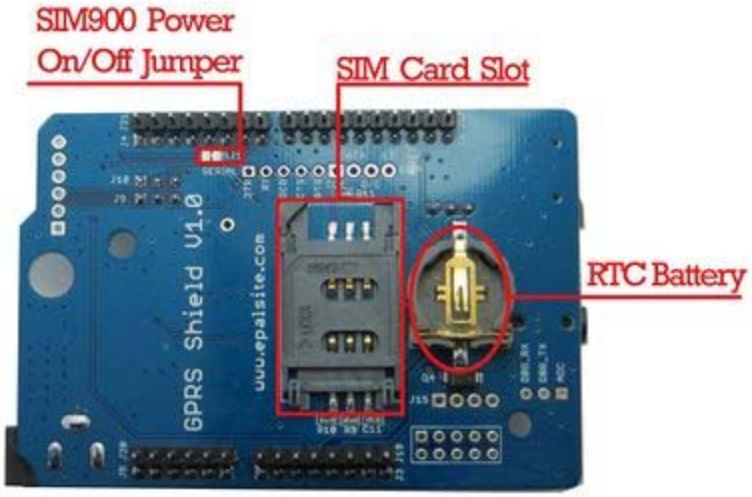
Parameters

Parameter	Condition	Min.	Typ.Max.	Max.	Uints
Vin		5		12	V
Current	Sleep Mode			1.5	mA
	Voice Call			250	
	GPRS Data Mode	76		440	
	During Tx Burst			2	A
Dimension		55*81			mm

For exact parameters please refer to the SIM900 datasheet.

Usage Instruction

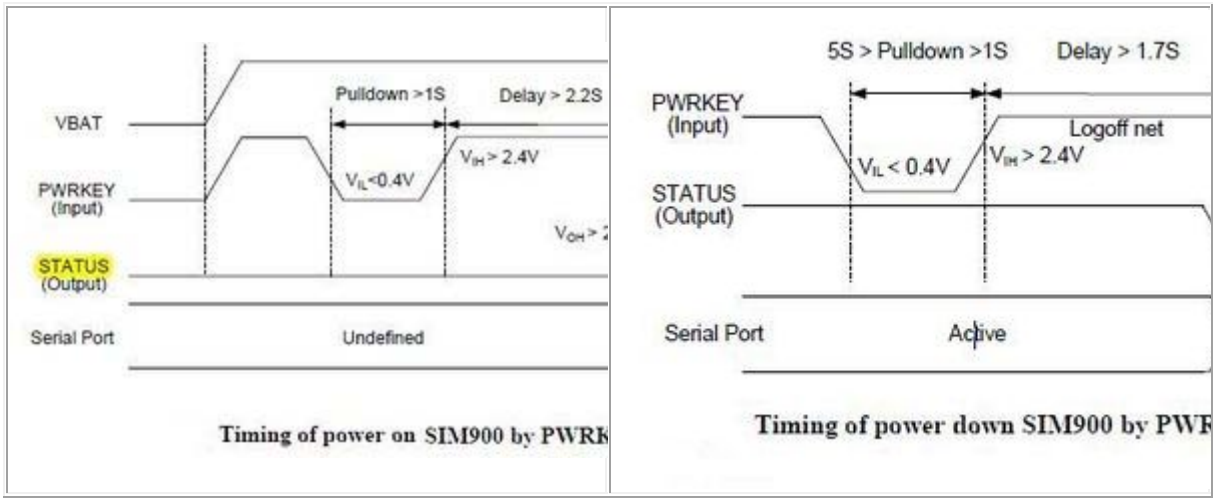
Board Overview



Explanation:

- **Power Jack:** Connected to 5V-12V adapter, this provide the power to SIM900 shield. The adapter can also be connected to Arduino, this way the Power Jack should be left open.If you have any doubt about this, please check the schematic.
- **Micro SD card(TF card)Slot:** Used with Micro SD card. Can use the jumper labeled with "SD_ENABLE" to enable or disable the SD card slot. If it is disabled, then the SD card will have no effect to the IOs(SPI port which are D10-D13)of Arduino.
- **Earphone Socket:** This is a two in one 3.5mm socket(microphone and speaker in one socket).Iphone 5's earphone is compatible.
- **Software or Hardware UART select:** Uart Selection. SIM900 GPRS shield can communicate with Arduino through the Hardware Serial port(D0/D1) or the Software Serial port(D2/D3). We can use the jumper to do the selection. The detailed signal are as following.
    - D0: Hardware serial RX of Arduino
    - D1: Hardware serial TX of Arduino
    - D2: Software serial RX of Arduino

    - D3: Software serial TX of Arduino
    - D7: Used as Software Power ON/OFF button for SIM900
- **FT232RL Interface:**You can use the PC or other host which have an USB port(through FT232RL board) to debug SIM900 shield through the FT232RL USB to Serial converter.
- **PWRKEY BUTTON:** SIM900 Power ON/OFF button. For this shield, we have two methods to Power ON/OFF SIM900. One method is to use the PWRKEY, this is called hardware Power ON/OFF. Another method is to use arduino to control this through D7.The power on/down scenarios for SIM900 is illustrated as following figure.



- **Arduino Reset Button:** This button is connected to the Reset pin of Arduino, can be used to reset Arduino directly.

- **LED Indicators:** There are two LEDs on the boards. One is named "PWR" which shows if the Power for SIM900 is on. The other led is named "NET" which shows the net status of the SIM900 shield.

LEDs(color)	Status	Description
PWR(Red)	ON	Power of the GPRS Shield is on
	OFF	Power of the GPRS Shield is off
Netlight(Green)	64ms On/800ms Off	SIM900 has not registered to a network
	64ms On/3000ms Off	SIM900 has registered to a network
	64ms On/300ms Off	GPRS communication
	OFF	SIM900 is not running

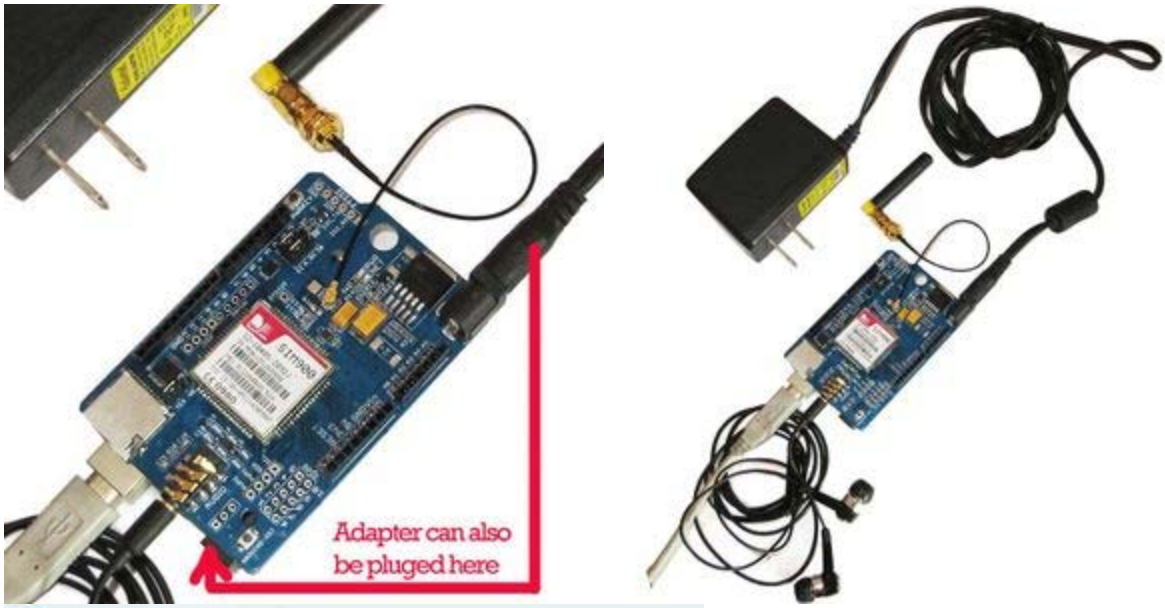
Step by Step Usage

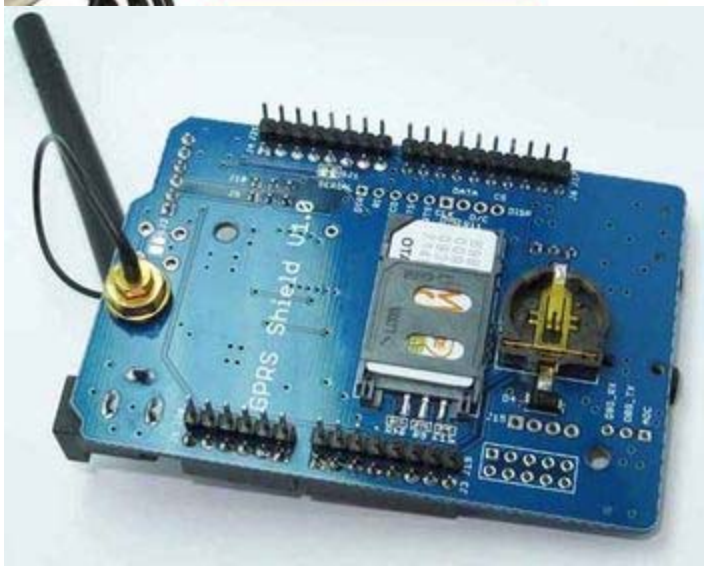
We have two ways to control this board. One way is using Arduino. Another way is using FT232RL USB to Serial converter to control the board through your computer.Below describes these two ways.

Using with Aruduino

We can use Arduino to send AT commands to SIM900, this way we can control SIM900 to do any works it can, like making a phone call, sending a short message,sending some data to a server through GPRS network. Just plug this shield on arduino. Please make sure to connect them correctly.

Hardware Connection





Description: The above pictures shows how to connect the hardware before use. Please note Arduino is at the bottom of the SIM900 shield.

1. The adapter should be 5V-12V output. 2A current capability is recommended. Please note the adapter can also be plugged on the Arduino. But don't plug two adapters at the same time: one to SIM900 shield and one to Arduino. This may cause some damage.
2. Arduino is connected to the computer through the USB cable.
3. Uart Selection. SIM900 GPRS shield can communicate with Arduino through the Hardware Serial port(D0/D1) or the Software Serial port(D2/D3). We can use the jumper to do the selection.

4. Other connection like the Antenna, the earphone, the SIM card should be done before use if needed.

Example1-Make a phone call

User can use AT command “AT+IPR=x” to set a fixed baud rate and save the configuration to non-volatile flash memory.After the configuration is saved as fixed baud rate, the Code “RDY” should be received from the serial port every time when SIM900 is powered on. For details, please refer to the "SIM900_AT Command Manual"

Now we will make a phone call using Arduino and SIM900 GPRS shield. In this example, we use the Hardware Serial to send the code to Arduino from the Computer. And using the software serial to send AT command from Arduino to SIM900 Shield. The steps are as following.

1. Connect hardware as the pictures shows in the Hardware Connection section.
    - Insert the SIM card.
    - Plug in your earphone to the TWO in ONE earphone socket.
    - Connect the Antenna.
    - Make sure you have set the Serial Port correctly through the Jumper. Just D2 to A_RX, and D3 to A_TX. This way the RX of SIM900 will be connected to D3 of Arduino, and the TX pin of SIM900 will be connected to D2 of Arduino.
    - Plug SIM900 shield on Arduino correctly.
    - Connect Arduino with computer through an USB cable.
    - Connect a 5V-12V adapter to the Power Jack of SIM900 or Arduino.
2. Download the following code to Arduino through Arduino IDE.

```
 /*Note:This code is used for Arduino 1.0 or later*/
#include <SoftwareSerial.h>

SoftwareSerial  Sim900Serial(2, 3);

void setup()
{
  Sim900Serial.begin(115200);               // the GPRS baud rate
  delay(2000);
  Sim900Serial.println("AT+IPR=19200");     // Set the baud rate
  delay(500);
  Sim900Serial.begin(19200);               // the GPRS baud rate
  delay(1000);
}
void loop()
{
  Sim900Serial.println("ATD**********;");//'*'instead the phone number you
want to dial
  while(1);
}
```

- Press the PWRKEY button to power on the SIM900 shield. You can see if it has connected to the Network through the NET led.
- When SIM900 is connected to the network, then press the ARDUINO_RST button to reset Arduino to make it resent the command. If no problem, then the phone number you're dialing will give a feedback.

Example2-Set the GPIO of the SIM900 GPRS Shield

In the following sketch we use arduino to set the shield to output a HIGH or LOW voltage on the GPIO ports. This is realized by send AT command "AT+SGPIO=..."

```
 /*Note:This code is used for Arduino 1.0 or later*/
#include <SoftwareSerial.h>
SoftwareSerial Sim900Serial(2, 3);
void setup()
{
  Sim900Serial.begin(115200);               // the GPRS baud rate
  delay(500);
  Sim900Serial.println("AT+IPR=19200");
  delay(500);
  Sim900Serial.begin(19200);               // the GPRS baud rate
  delay(1000);
}
void loop()
{
    while(1)
  {
     Sim900Serial.println("AT+SGPIO=0,1,1,1");// set GPIO 1 PIN to 1
     delay(1000);
     Sim900Serial.println("AT+SGPIO=0,1,1,0");// set GPIO 1 PIN to 0
     delay(1000);
  }
}
```

Example3-Sending a SMS message

The following example shows a simple example of how to send a SMS. In this example, we send the text"hello" to a target phone number.

```
/*Note:This code is used for Arduino 1.0 or later*/
#include <SoftwareSerial.h>
SoftwareSerial Sim900Serial(2, 3);
void setup()
{
  Sim900Serial.begin(115200);               // the GPRS baud rate
  delay(500);
  Sim900Serial.println("AT+IPR=19200");
  delay(500);
  Sim900Serial.begin(19200);               // the GPRS baud rate
  delay(1000);
  Serial.begin(9600);               // the Hardware serial rate
 Serial.println("Please type 's' to send  SMS");
}
void loop()
{
   if (Serial.available())
    switch(Serial.read())
   {
     case 's':
       SendTextMessage();
       break;
    }
   if (Sim900Serial.available())
    Serial.write(Sim900Serial.read());
}
```

```
void SendTextMessage()
{
  Sim900Serial.print("AT+CMGF=1\r");    //Sending the SMS in text mode
  delay(100);
  Sim900Serial.println("AT + CMGS = \"***********\"");//The target phone
number
  delay(100);
  Sim900Serial.println("hello");//the content of the message
  delay(100);
  Sim900Serial.println((char)26);//the ASCII code of the ctrl+z is 26
  delay(100);
  Sim900Serial.println();
}
```

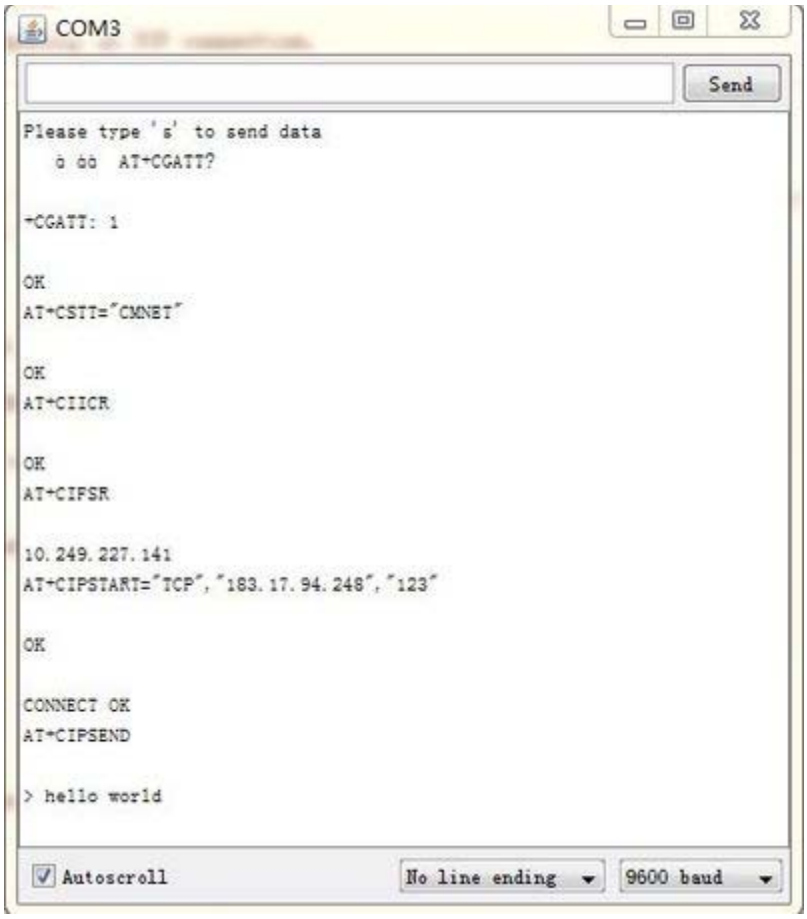
Example4-Send data to internet through a TCP connection

The following simple shows the flow of sending data to internet by opening an TCP connection.First make sure you've already power on SIM900 shield. Then download the following code to your arduino(But first you should change the IP and the TCP server port to your own before downloading). The steps are as following:

1. Power on GPRS shield and arduino.
2. Change the code to meet your fact. Like changing the IP address and the TCP server port.
3. Open Serial Monitor of Arduino, then type 's' to continue the process
4. Check your server to see if it has get the data transferred by the GPRS shield.

```
 /*Note:This code is used for Arduino 1.0 or later*/
#include <SoftwareSerial.h>
SoftwareSerial Sim900Serial(2, 3);
#define sendtext "hello world"
#define ip_address "AT+CIPSTART=\"TCP\",\"183.17.94.248\",\"123\""
//"183.17.94.248"is the ip you want to send the data to, "123" is the TCP
server's port, please change them based on your requirement.
byte buffer[64]; // buffer array for data recieve over serial port
int count=0;     // counter for buffer array
void setup()
{
  Sim900Serial.begin(115200);               // the SoftSerial baud rate
  delay(500);
  Sim900Serial.println("AT+IPR=19200");    //set the baud of SIM900
  delay(500);
  Sim900Serial.begin(19200);               // the SoftSerial baud rate
  delay(1000);
  Serial.begin(9600);               // the Hardware serial rate
  Serial.println("Please type 's' to send data"); //Type 's' TO send the
data to internet
}
void loop()
{
   if (Serial.available())
    switch(Serial.read())
   {
       case 's':
      {
        GPRS_SendText();                   // Send data to internet
        break;
      }
```

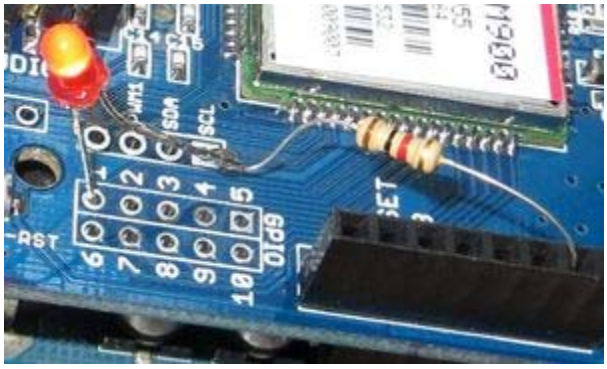
```
   }
}

boolean Sim900_SendATCmd(char *pCmd)
{
  Sim900Serial.println(pCmd);
  delay(500);
  GetBuffer();
}
void GPRS_SendText()
{
  Sim900_SendATCmd("AT+CGATT?");
  delay(1000);
  Sim900_SendATCmd("AT+CSTT=\"CMNET\"");//Set the APN
  delay(1000);
  Sim900_SendATCmd("AT+CIICR");// Bring up GPRS connection
  delay(1000);
  Sim900_SendATCmd("AT+CIFSR");// Get local IP address
  delay(1000);
  Sim900_SendATCmd(ip_address);
  delay(1000);
  Sim900_SendATCmd("AT+CIPSEND");//Send data to remote server
  delay(100);
  Sim900Serial.print(sendtext);//The text you want to send
  delay(100);
  Sim900Serial.println((char)26);//the ASCII code of the ctrl+z is 26
  delay(1000);
  GetBuffer();
}
void clearBufferArray()              // function to clear buffer array
{
  for (int i=0; i<count;i++)
    { buffer[i]=NULL;}                  // clear all index of array with
command NULL
}
void GetBuffer()
{
  if (Sim900Serial.available())              // if date is comming from
softwareserial port ==> data is comming from gprs shield
  {
    while(Sim900Serial.available())          // reading data into char
array
    {
      buffer[count++]=Sim900Serial.read();     // writing data into array
      if(count == 64)break;
  }
      Serial.write(buffer,count);            // if no data transmission
ends, write buffer to hardware serial port
      clearBufferArray();              // call clearBufferArray function to
clear the storaged data from the array
    count = 0;
  }
}
```



Example5-Using SMS to control the output state of the GPIO of SIM900 GPRS Shield

In this sketch we use another phone to send a SMS to SIM900 GPRS Shield to make it output a HIGH or LOW voltage on the GPIO. Steps:

1. Open the Serial Monitor of arduino IDE
2. You can connect the GPIO 1 of SIM900 shield to a led or something else that can show the state of it.
3. Use your second phone to send SMS with content of "ON" or "OFF" to the SIM900 GPRS shield, but don't forget to set the phone number used to send the command in the sample code.
4. The GPIO 1 should output HIGH or LOW voltage based on the SMS.



```
/*Note:This code is used for Arduino 1.0 or later*/
#include <SoftwareSerial.h>
#include <string.h>
SoftwareSerial Sim900Serial(2, 3);
byte buffer[64]; // buffer array for data recieve over serial port
int count=0;     // counter for buffer array
#define phonenumber "+8618612345678"  //Change to your phonenumber, the
phonenumber should be the same as the format received by the SMS, may
include your country code.
void setup()
{
  Sim900Serial.begin(19200);               // the SIM900 baud rate
  Serial.begin(19200);             // the Serial port of Arduino baud rate.
  delay(500);
  Sim900_Inti();
}

void loop()
{
  if (Sim900Serial.available())              // if date is comming from
softwareserial port(data is comming from gprs shield)
  {
    while(Sim900Serial.available())          // reading data into char
array
    {
      buffer[count++]=Sim900Serial.read();     // writing data into array
      if(count == 64)break;
  }
    Serial.write(buffer,count);            // if no data transmission ends,
write buffer to hardware serial port
    Cmd_Read_Act();                        //Read the 'COMMAND' sent to
SIM900 through SMS
    clearBufferArray();              // call clearBufferArray function to
clear the storaged data from the array
    count = 0;                       // set counter of while loop to zero


  }
  if (Serial.available())            // if data is available on
hardwareserial port ==> data is comming from PC or notebook
    Sim900Serial.write(Serial.read());       // write it to the GPRS shield
}
void clearBufferArray()              // function to clear buffer array
{
  for (int i=0; i<count;i++)
    { buffer[i]=NULL;}                  // clear all index of array with
command NULL
}
void Sim900_Inti(void)
{
  Sim900Serial.println("AT+CMGF=1");
  delay(500);
  Sim900Serial.println("AT+CNMI=2,2");
   delay(500);
}
void Cmd_Read_Act(void)          //This function read the SMS sent to
SIM900 shield, then act based on the command.
{
  char buffer2[64];
  char comparetext[25];    //take out the first part of the SMS which
include the phoneunmber
```

```
  for (int i=0; i<count;i++)
  { buffer2[i]=char(buffer[i]);}
  memcpy(comparetext,buffer2,25); //take out the first part of the SMS
which include the phoneunmber
  if (strstr(comparetext,phonenumber))
  {
  if (strstr(buffer2,"ON"))           //If there are word 'ON' in the SMS,
turn on GPIO 1 of SIM900
    {
      Sim900Serial.println("AT+SGPIO=0,1,1,1");// set GPIO 1 PIN to 1
    }
    if (strstr(buffer2,"OFF"))        //If there are word 'OFF' in the SMS,
turn off GPIO 1 of SIM900
    {
      Sim900Serial.println("AT+SGPIO=0,1,1,0");// set GPIO 1 PIN to 0
    }
  }
}
```

RTC Usage

The SIM900 shield uses an CR1220 battery. For SIM900, we can use AT command:"AT+CCLK" to control the RTC.

1. AT+CCLK=<time>. This command is used to set time in this format:"yy/MM/dd,hh:mm:ss±zz",where characters indicate year (two last digits),month, day, hour,minutes,seconds and time zone (indicates the difference, expressed in quarters of an hour, between the local time and GMT; range -47...+48). E.g. 6th of May 2010, 00:01:52 GMT+2 hours equals to "10/05/06,00:01:52+08"
2. AT+CCLK?. This command is used to read out the time from SIM900

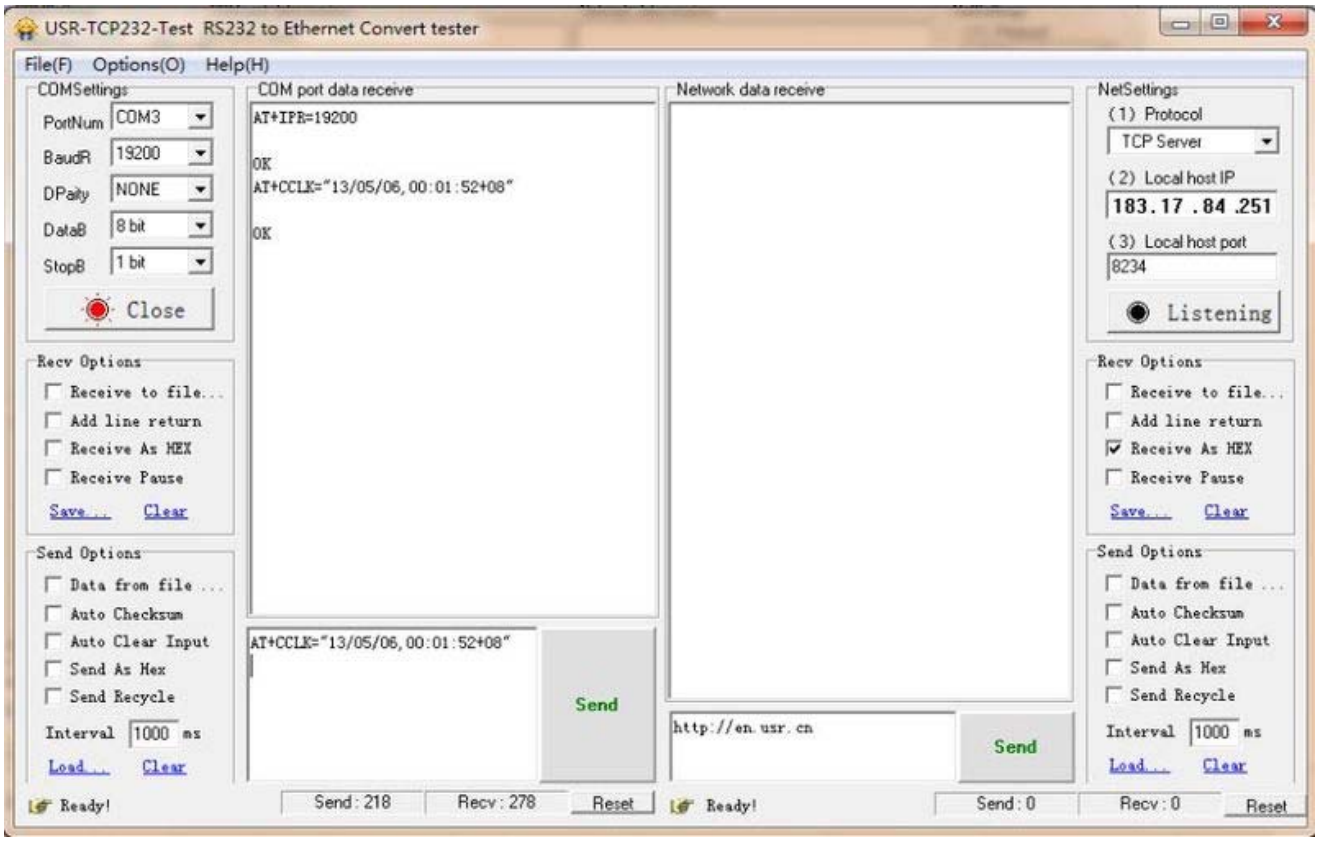
In the following example, we use arduino as a bridge between PC and SIM900 shield, then we use the software Media:TCP232.rar to debug the RTC

```
/*Note:This code is used for Arduino 1.0 or later*/
#include <SoftwareSerial.h>
SoftwareSerial GPRS(2, 3);
unsigned char buffer[64]; // buffer array for data recieve over serial port
int count=0;     // counter for buffer array
void setup()
{
  GPRS.begin(19200);               // the GPRS baud rate
  Serial.begin(19200);             // the Serial port of Arduino baud rate.
}

void loop()
{
  if (GPRS.available())              // if date is comming from
softwareserial port ==> data is comming from gprs shield
  {
    while(GPRS.available())          // reading data into char array
    {
      buffer[count++]=GPRS.read();     // writing data into array
      if(count == 64)break;
    }
    Serial.write(buffer,count);            // if no data transmission ends,
write buffer to hardware serial port
```

```
    clearBufferArray();              // call clearBufferArray function to
clear the storaged data from the array
    count = 0;                       // set counter of while loop to zero
   }
    if (Serial.available())            // if data is available on
hardwareserial port ==> data is comming from PC or notebook
    GPRS.write(Serial.read());       // write it to the GPRS shield
}
void clearBufferArray()              // function to clear buffer array
{
  for (int i=0; i<count;i++)
    { buffer[i]=NULL;}                  // clear all index of array with
command NULL
}
```

Please test the RTC by following the following steps.

1. Plug SIM900 shield on arduino, then connect arduino and PC.
2. Download the up code to arduino. Then open TCP232
3. Set TCP232 based on the following picture



4. Send AT command to SIM900 through TCP232 like the up picture. If you can't get any feedback, then maybe the problem is the baud rate. You can send "AT+IPR=19200" first to have a try.

Using with FT232RL module