

Documentation: GPRS/GSM Quadband Module for Arduino Tutorial (SIM900)


Tutorial Index

- [Introduction](#)
- [Step 1: The shield \(hardware\)](#)
- [Step 2: Using GSM/GPRS module with AT commands](#)
- [Step 3: Powering the board](#)
- [Step 4: Using the shield in standalone mode - Calls](#)
 - [Originating and receiving voice calls](#)
 - [Command summary](#)
- [Step 5: Using the shield in standalone mode - Sending and receiving SMS](#)
 - [Command summary](#)
- [Step 6: Using the shield in standalone mode - FTP](#)
 - [Command summary](#)
- [Step 7: Using the shield in standalone mode - TCP and UDP](#)
 - [Single client](#)
 - [Multiple client](#)
 - [Command summary](#)
- [Step 8: Using the shield in standalone mode - HTTP](#)
 - [Command summary](#)
- [Fritzing Libraries](#)
- [Links and Documentation](#)

Introduction

Ingredients:

- 1 x [Arduino](#)
- 1 x [GPRS/GSM Quadband Module for Arduino \(SIM900\)](#)
- 1 x [GPRS antenna](#)
- 1 x SIM card
- 1 x PC

Difficulty: Medium - 

Preparation Time: 45 minutes



NOTE: If you are looking for a complete solution to use 3G, GPRS and A-GPS, you can use our [3G/GPRS shield for Arduino \(3G + GPRS\)](#) or our [Kit with Audio/Video](#)

NOTE: The codes of the tutorial have developed to work on Arduino IDE v0.22.

Step 1: The shield (hardware)

The board (shield) we are going to use in this tutorial is the [GPRS/GSM Quadband Module for Arduino \(SIM900\)](#) from Cooking hacks.

The GPRS shield is fully compatible with old Arduino USB versions, Duemilanove and Mega.

GPRS Shield diagram:



COMPARE PRODUCTS

You have no items to compare.

CART

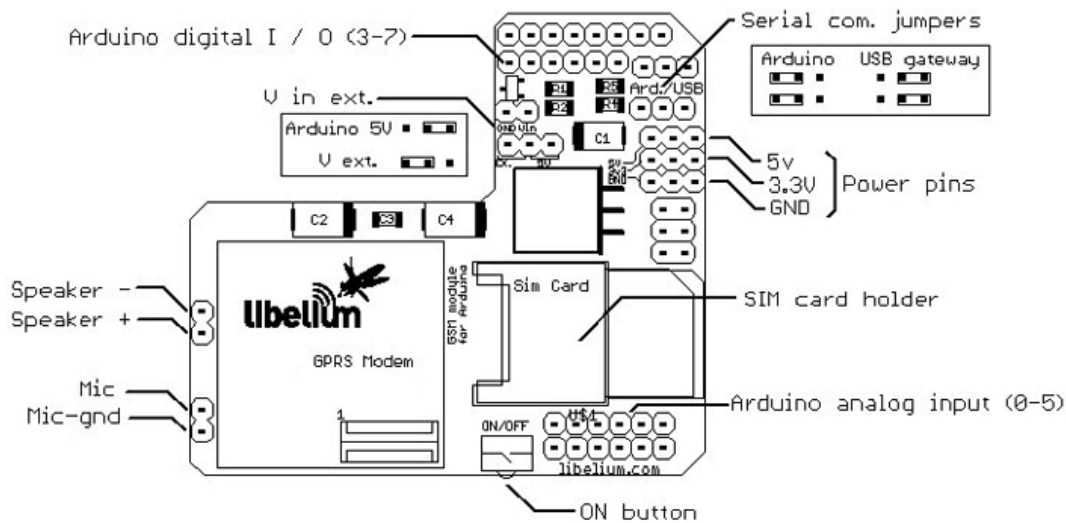
You have no items in your shopping cart.

NEWSLETTER

Sign up for our newsletter
Sign Up for Our Newsletter:



**OHS 2012
SPONSOR**



Step 2: Using GSM/GPRS module with AT commands

Important issues:

- Use capital letters for AT commands.
- Send CR (Carriage return) and LF (Line feed) after the AT command.
- Place the serial communication jumpers in the right position.
- Use an external power supply and place the power jumpers in the right position. If the shield is powered from the Arduino, the power jumper must be in Arduino 5V position. If the shield is powered from the Vin input (in the shield), the power jumper must be in Vext position.

The first thing we are going to do with the module is to connect the module to a PC directly (using an Arduino as gateway) and check the basic AT commands. In this case, serial communication jumpers have to be set on USB gateway position.

Remember take out the ATmega microcontroller from the Arduino gateway.

Basic configuration:



Connect the shield to the Arduino gateway.



Then connect the USB cable and the SIM card.

Finally plug the USB cable to the computer and open a serial port terminal to communicate via the usb port (e.g: hyperterminal (win), cutecom / gtkterm (linux)).

If you use the Arduino IDE serial monitor for sending AT commands – Be sure that you are sending CR (Carriage return) and LF (Line Feed).

Set the baudrate to 115200 bps and open the serial port, then press the ON button for two seconds. Then if you type **AT** you'll get **OK**, this means that the communication with the module is working fine. Now, with the module working you can check some AT commands to control the module, the basic commands are:

Important type commands in capital letters and with CR (carriage return) and LF (line feed)!!!

Command	Response	Description
AT	OK	If you get OK, the communication with the module is working
AT+CPIN="****"	OK	If the SIM card is locked with PIN (**** is the pin number)
AT+COPS?		Operator information

Note

Factory baudrate setting is auto-bauding by default. Baudrate can be fixed using the command AT+IPR=baudrate . Allowed baudrates: 0 (Auto-bauding) , 1200 , 2400 , 4800 , 9600 , 19200 , 38400 , 57600 and 115200 ;

All the AT commands [here](#)

Step 3: Powering the board:

Some of the USB ports on computers are not able to give all the current the module needs to work, if your module goes down when it tries to connect to the network, you can use an external power supply (12V - 2A) on the Arduino.

Remember set the Arduino power jumper to EXT!!! (if you use Diecimila or older).

How to set the power jumper in the shield?.

If you want the shield takes power from Arduino => Set the jumper to Arduino 5V position

If you want the shield takes power from an external supply => Set the jumper to V ext. position

For powering the shield from external supply, you have to use V in ext. connector (Vin + GND).

If you use a power supply with output smaller than 2 A, you should add an extra capacitor for the power.

For example, a 220 uF electrolytic capacitor between 3.3V and GND.

Step 4: Using the shield in standalone mode - Calls

Originating and receiving voice calls

The code example and the connection diagram shown below are used to originate a voice call and, pushing a button, end that

voice call. The button is connected between digital pin 12 and ground. A 10kΩ pull-up resistor is needed at this pin.

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

int led = 13;
int button = 12;
int onModulePin = 2;      // the pin to switch on the module (without press on button)

int timesToSend = 1;      // Numbers of calls to make
int count=0;

char phone_number[]="*****"; // ***** is the number to call

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

    Serial.begin(115200);      // UART baud rate
    delay(2000);
    pinMode(button, INPUT);
    pinMode(led, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    switchModule();           // switches the module ON

    for (int i=0;i < 5;i++){
        delay(5000);
    }
}

void loop(){

    while (count < timesToSend){

        Serial.print("ATD");
        Serial.print(phone_number);
        Serial.println("");

        while(digitalRead(button)==1);

        Serial.println("ATH");      // disconnects the existing call

        delay(5000);
    }
}

```

```

        count++;
    }
}

```

To make a lost call next code is used.

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

int led = 13;
int onModulePin = 2;          // the pin to switch on the module (without press on button)

int timesToSend = 1;          // Numbers of calls to make
int count = 0;
int tones = 5;                //Each tone has a duration of 6 seconds aprox.

char phone_number[]="*****"; // ***** is the number to call

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

    Serial.begin(115200);        // UART baud rate
    delay(2000);
    pinMode(led, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    switchModule();              // switches the module ON

    for (int i=0;i < 5;i++){
        delay(5000);
    }
}

void loop(){

    while (count < timesToSend){

        Serial.print("ATD");
        Serial.print(phone_number);
        Serial.println("");
        delay(6000*tones);
        Serial.println("ATH");    // cancel the call
    }
}

```

```

        delay(5000);

        count++;
    }
}

```

To receive calls the used code are this and the connection diagram is the same that the used to originate calls. Don't forget the pull-up resistor on pin 12.

Command summary

Command	Response	Description
ATD*****;		***** is the number to call.
ATA	OK	Answer an incoming call.
ATH	OK	Cancel voice calls.

Step 5: Using the shield in standalone mode - Sending and receiving SMS

The first code is used to send a SMS, the second one reads the first SMS into the memory.

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

int led = 13;
int onModulePin = 2;      // the pin to switch on the module (without press on button)

int timesToSend = 1;      // Numbers of SMS to send
int count = 0;

char phone_number[]="*****";    // ***** is the number to call

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

    Serial.begin(115200);          // UART baud rate
    delay(2000);
    pinMode(led, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    switchModule();                // switches the module ON

    for (int i=0;i < 5;i++){

```

```

        delay(5000);
    }

    Serial.println("AT+CMGF=1");          // sets the SMS mode to text
    delay(100);
}

void loop(){

    while (count < timesToSend){
        delay(1500);
        Serial.print("AT+CMGS=\"");      // send the SMS number
        Serial.print(phone_number);
        Serial.println("\"");
        while(Serial.read()!='>');
        Serial.print("Hola caracola...");    // the SMS body
        delay(500);
        Serial.write(0x1A);                //sends ++
        Serial.write(0x0D);
        Serial.write(0x0A);

        delay(5000);

        count++;
    }
}

```

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

int led = 13;
int onModulePin = 2;          // the pin to switch on the module (without press on button)

int timesToSend = 1;          // Numbers of SMS to send
int count = 0;

int n_sms,x,sms_start;
char data[256];

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

```

```
Serial.begin(115200);           // UART baud rate
delay(2000);
pinMode(led, OUTPUT);
pinMode(onModulePin, OUTPUT);
switchModule();                // switches the module ON

for (int i=0;i < 5;i++){
    delay(5000);
}

Serial.println("AT+CMGF=1");    // sets the SMS mode to text
delay(1000);
}

void loop(){

    Serial.println("AT+CMGR=1"); //Reads the first SMS
    Serial.flush();
    for (x=0;x < 255;x++){
        data[x]='\0';
    }
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
        if(data[x-1]==0x0D&&data[x-2]==' '){
            x=0;
        }
    }while(!(data[x-1]=='K'&&data[x-2]=='0'));

    data[x-3]='\0';            //finish the string before the OK

    Serial.println(data);      //shows the message

    delay(5000);
}
```

Command summary

Command	Response	Description
AT+CMGF=	OK	Specifies the input and output format of the short messages. 0 for PDU mode and 1 for text mode.
AT+CMGS		Sends a message.
AT+CMGR=*		Reads a message. * is the number of the message.

Step 6: Using the shield in standalone mode - FTP

Creating a file into the FTP server, writing it and reading it.

[Hide Code](#)

```
/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
```



```

*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see .
*
* Version 0.1
* Author: Alejandro Gállego
*/

int led = 13;
int onModulePin = 2;          // the pin to switch on the module (without press on button)

char data[1024];
int x=0;
int data_size;
char aux;

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

    Serial.begin(115200);          // UART baud rate
    delay(2000);
    pinMode(led, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    switchModule();               // switches the module ON

    for (int i=0;i < 4;i++){
        delay(5000);
    }

    Serial.println("AT+SAPBR=3,1,\"Contype\",\"GPRS\"");    //Configures GPRS connection
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+SAPBR=3,1,\"APN\",\"myAPN\"");
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+SAPBR=1,1");
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+FTPCID=1");    //Selects profile 1 for FTP operations
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+FTPSERV=\"FTP_IP_dir\"");    //Configures FTP parameters
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+FTPPORT=port_number");
    do{
        while(Serial.available()==0);
    }

```

```

}while(Serial.read()!='K');
delay(100);

Serial.println("AT+FTPPUN=\"username\"");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+FTPPW=\"password\"");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

}

void loop(){

    Serial.println("AT+FTPPUTNAME=\"filename.extension\"");    //Sets destiny file name
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+FTPPUTPATH=\"/\"");    //Sets destiny file path
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+FTPPUT=1");    //Starts session
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\n');
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\n');
    delay(100);

    data_size=0;
    while(Serial.available()==0);

    aux=Serial.read();
    do{
        data_size*=10;
        data_size+=(aux-0x30);
        while(Serial.available()==0);
        aux=Serial.read();
    }while(aux!=13);

    Serial.println("AT+FTPPUT=2,100");    //Sends data
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='+');
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\r');
    delay(100);

    Serial.print("00000000111111122222223333333444444455555556666666777777700000000111111122222233333334444");

    do{
        while(Serial.available()==0);
    }while(Serial.read()!='+');
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\r');
}

```

```

delay(100);

Serial.println("AT+FTPPUT=2,0");    //Closes the file and the session
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);
do{
    while(Serial.available()==0);
}while(Serial.read()!='+');
delay(100);

Serial.println("AT+FTPGETNAME=\"filename.extension\"");    //Sets file name to download
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+FTPGETPATH=\"/\"");    //Sets file path to download
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+FTPGET=1");    //Starts session
do{
    while(Serial.available()==0);
}while(Serial.read()!=' ');
delay(100);

Serial.println("AT+FTPGET=2,512");    //Request data
do{
    while(Serial.available()==0);
}while(Serial.read()!=' ');
do{
    while(Serial.available()==0);
}while(Serial.read()!=' ');
delay(100);

data_size=0;
while(Serial.available()==0);

aux=Serial.read();    //Gets data length
do{
    data_size*=10;
    data_size+=(aux-0x30);
    while(Serial.available()==0);
    aux=Serial.read();
}while(aux!=13);

for(x=0;x<=data_size;x++){    //Gets data string
    while(Serial.available()==0);
    data[x]=Serial.read();
}

do{
    while(Serial.available()==0);
}while(Serial.read()!=':');
delay(100);

Serial.print(data);

while(1);
}

```

Command summary

AT command	Response	Description
AT+SAPBR	OK	Configures GPRS profile.
AT+FTPCID=1	OK	Selects profile 1 for FTP.
AT+FTPSERV="*****"	OK	Sets FTP server domain name or IP address. ***** is the domain name or the IP.
AT+FTPSPORT=***	OK	Sets FTP server port. *** is the port.
AT+FTPUN="***"	OK	Sets user name for FTP server access. *** is the user name.
AT+FTPPW="***"	OK	Sets password for FTP server access. *** is the password.
AT+FTPPUTNAME="*****"	OK	Sets destiny name for the file.*** is the name of the file.
AT+FTPPUTPATH="*****"	OK	Sets destiny file path. *** is the path of the file.
AT+FTPPUT	OK	Use to put a file into the FTP server.
AT+FTPGETNAME="*****"	OK	Sets origin name for the file.*** is the name of the file.
AT+FTPGETPATH="*****"	OK	Sets origin file path. *** is the path of the file.
AT+FTPGET		Use to get a file into the FTP server.

Step 7: Using the shield in standalone mode - TCP and UDP

Single client

Sending data to a TCP server first, and then it sends to a UDP server.

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

char data[1024];
int led = 13;
int onModulePin = 2;      // the pin to switch on the module (without press on button)

int x = 0;

char name[20];

char server[ ]="";
char port[ ]="";

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

```

```
void setup(){

  Serial.begin(115200);           // UART baud rate
  delay(2000);
  pinMode(led, OUTPUT);
  pinMode(onModulePin, OUTPUT);
  switchModule();                // switches the module ON

  for (int i=0;i < 4;i++){
    delay(5000);
  }

  Serial.println("AT+CIPMUX=0");  //single connection
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='K');
  delay(100);

  Serial.println("AT+CSTT=\"myAP\"");
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='K');
  delay(100);

  Serial.println("AT+CIICR");
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='K');
  delay(100);

  Serial.println("AT+CIFSR");
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='.');
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='\n');
  delay(100);

  Serial.println("AT+CIPSTART=\"TCP\", \"IP_dir\", \"port\"");
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='K');
  delay(100);

  do{    //waits for connection
    while(Serial.available()==0);
  }while(Serial.read()!='K');
  delay(100);

  Serial.println("AT+CIPSEND");  //sending data
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='>');
  delay(100);
  Serial.print("Data test!");
  Serial.write(0x1A);  //EOL character
  Serial.write(0x0D);
  Serial.write(0x0A);
  delay(100);

  Serial.println("AT+CIPCLOSE");
  do{
    while(Serial.available()==0);
  }while(Serial.read()!='K');
  delay(100);
```

```

Serial.println("AT+CIPSTART=\"UDP\", \"IP_dir\", \"port\"");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

do{    //waits for connection
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+CIPSEND");    //sending data
do{
    while(Serial.available()==0);
}while(Serial.read()!='>');
delay(100);
Serial.print("Data test!");
Serial.write(0x1A);    //EOL character
Serial.write(0x0D);
Serial.write(0x0A);
delay(100);

Serial.println("AT+CIPCLOSE");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');

}

void loop()
{

    delay(500);
    digitalWrite(led,HIGH);
    delay(500);
    digitalWrite(led,LOW);

}

```

Multiple client

SIM900 allows to use 8 connections simultaneously. Here is the example code with a UDP and TCP connections.

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

```

```

char data[1024];
int led = 13;
int onModulePin = 2;          // the pin to switch on the module (without press on button)

int x = 0;

char name[20];

char server[ ]="";
char port[ ]="";

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

    Serial.begin(115200);          // UART baud rate
    delay(2000);
    pinMode(led, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    switchModule();               // switches the module ON

    for (int i=0;i < 4;i++){
        delay(5000);
    }

    Serial.println("AT+CIPMUX=1");    //multiple connection
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+CSTT=\"myAPN\"");
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+CIICR");
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+CIFSR");
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='. ');
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\n');
    delay(100);

    Serial.println("AT+CIPSTART=0,\"TCP\", \"IP_dir\", \"port\");    //TCP connection in id 0
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    do{    //waits for connection
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+CIPSTART=1,\"UDP\", \"IP_dir\", \"port\");    //UDP connection in id 1

```

```
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

do{    //waits for connection
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+CIPSEND=0"); //sending data
do{
    while(Serial.available()==0);
}while(Serial.read()!='>');
delay(100);
Serial.print("Data test!");
Serial.write(0x1A);    //EOL character
Serial.write(0x0D);
Serial.write(0x0A);
delay(100);

Serial.println("AT+CIPSEND=1"); //sending data
do{
    while(Serial.available()==0);
}while(Serial.read()!='>');
delay(100);
Serial.print("Data test!");
Serial.write(0x1A);    //EOL character
Serial.write(0x0D);
Serial.write(0x0A);
delay(100);

Serial.println("AT+CIPCLOSE=0");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+CIPCLOSE=1");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');

}

void loop()
{

    delay(500);
    digitalWrite(led,HIGH);
    delay(500);
    digitalWrite(led,LOW);

}
```

Command summary

AT command	Response	Description
AT+CIPMUX=	OK	Selects single connection (0) or multiple connection (1)
AT+CSTT="myAPN"	OK	Sets APN
AT+CIICR		Brings up wireless connection
AT+CIFSR		Get local IP address
AT+CIPSTART		Establishes a connection with a server.

AT+CIPSEND		Sends data when the a connection is established.
AT+CIPCLOSE		Closes the connection

Step 8: Using the shield in standalone mode - HTTP

SIM900 can launch a HTTP operation like GET or POST. Here is an example with GET operation:

[Hide Code](#)

```

/*
 * Copyright (C) 2012 Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see .
 *
 * Version 0.1
 * Author: Alejandro Gállego
 */

char data[1024];
int data_size;
int led = 13;
int onModulePin = 2;      // the pin to switch on the module (without press on button)
char aux;
int x = 0;

void switchModule(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}

void setup(){

    Serial.begin(115200);          // UART baud rate
    delay(2000);
    pinMode(led, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    digitalWrite(led,LOW);
    switchModule();                // switches the module ON

    for (int i=0;i < 5;i++){
        delay(5000);
    }

    Serial.println("AT+SAPBR=3,1,\"Contype\",\"GPRS\");    //Sets GPRS parameters
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+SAPBR=3,1,\"APN\",\"myAPN\");
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');

```

```

delay(100);

Serial.println("AT+SAPBR=1,1");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);
Serial.println("AT+HTTPINIT");
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.println("AT+HTTPPARA=\"CID\",1");    //Sets GPRS profile to use with HTTP
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);
}

void loop()
{

    Serial.println("AT+HTTPPARA=\"URL\", \"www.libelium.com\"");    //Sets URL direction
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='K');
    delay(100);

    Serial.println("AT+HTTPACTION=0"); //Connects with the HTTP server, this action can need to wait a several time
    x=0;

    do{
        while(Serial.available()==0);
    }while(Serial.read()!=' ');
    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\r');
    delay(100);

    Serial.println("AT+HTTPREAD=0,200");    //Reads 200 bytes from byte 0

    do{
        while(Serial.available()==0);
    }while(Serial.read()!=':');

    data_size=0;
    while(Serial.available()==0);

    aux=Serial.read();
    do{
        data_size*=10;
        data_size+=(aux-0x30);
        while(Serial.available()==0);
        aux=Serial.read();
    }while(aux!='\r');

    do{
        while(Serial.available()==0);
    }while(Serial.read()!='\n');

    for(x=0;x<=data_size;x++){
        while(Serial.available()==0);
        data[x]=Serial.read();
    }
}

```

```
Serial.println("AT+HTTPTERM");    //Closes the opened HTTP session
do{
    while(Serial.available()==0);
}while(Serial.read()!='K');
delay(100);

Serial.print(data);    //Shows data

while(1);

}
```

Command summary

AT command	Response	Description
AT+SAPBR	OK	Configures GPRS profile
AT+HTTINIT	OK	Initializes HTTP service
AT+HTTTPARA	OK	Configures HTTP parameters
AT+HTTTPACTION=0	OK	Sets HTTP Method Action , GET in this chase.
AT+HTTTPREAD		Reads HTTP data
AT+HTTPTERM	OK	Closes the opened HTTP session.

Fritzing Libraries

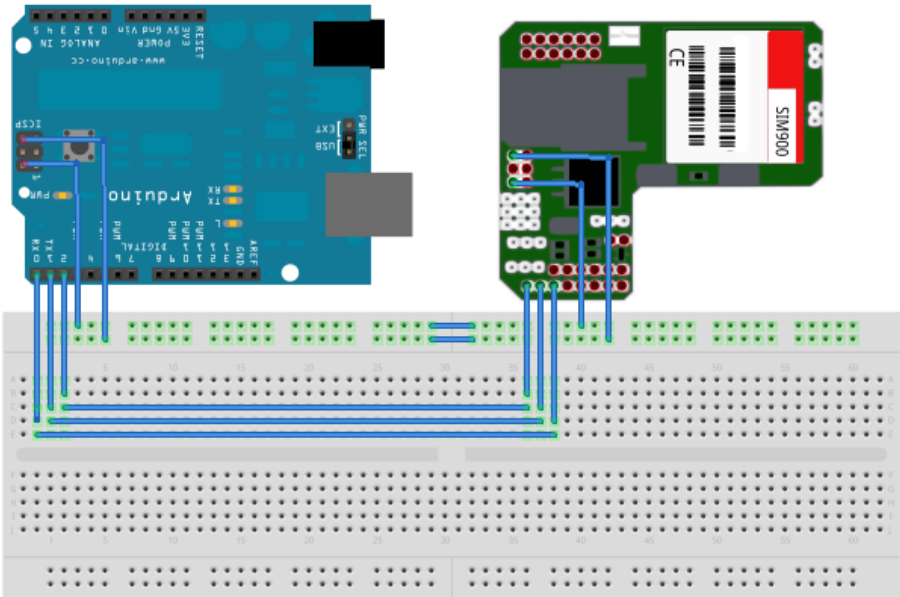


GPRS/GSM Quadband Module for Arduino (SIM900)

Download

[GPRS/GSM Quadband Module for Arduino \(SIM900\)](#) offers GPRS connection to your Arduino board. It includes the SIM900 communication module from SIMCom.

You can download our Fritzing libraries from [this area](#).



Links and Documentation

- [Schematic of the shield](#)
- [AT command manual](#)
- [SIM900 Module Info](#)

NOTE: If you are looking for a complete solution to use 3G, GPRS and A-GPS, you can use our [3G/GPRS shield for Arduino \(3G + GPRS\)](#) or our [Kit with Audio/Video](#)