

New 3G + GPS shield for Arduino

Technical Documentation

The new 3G shield for Arduino enables the connectivity to high speed WCDMA and HSPA cellular networks in order to make possible the creation of the next level of worldwide interactivity projects inside the new "Internet of Things" era.

1.	Introduction.....	4
2.	Features.....	5
3.	Kit and accessories	6
4.	Features.....	7
4.1.	Electrical features.....	7
4.2.	Radio features	7
4.3.	Extra features	7
5.	Hardware.....	8
5.1.	3G Shield diagram	8
5.2.	3G Shield PCB	8
6.	Antennas.....	9
6.1.	Connections.....	9
7.	Bandwidth	10
8.	Using 3G module with AT commands	10
9.	Using 3G module like a modem	12
10.	Audio	13
10.1.	Audio features	13
10.2.	Audio connectors.....	13
10.3.	Recording and playing sound	20
10.4.	Command summary	23
11.	Camera	23
11.1.	Camera features	23
11.2.	Video recording	26
11.3.	Taking photos	28
11.4.	Video call	30
11.5.	Command summary	31
12.	GPS.....	31
12.1.	Stand-alone mode	32
12.2.	S-GPS mode	35
12.3.	A-GPS mode.....	38
12.4.	Command summary	39
13.	SMS.....	39
13.1.	Sending and receiving SMS.....	39

13.2.	Command summary	42
14.	E-mails	43
14.1.	Sending an e-mail	43
14.2.	Receiving an e-mail.....	47
14.3.	Command summary	50
15.	SD and system memory.....	50
16.	FTP and FTPS.....	52
16.1.	FTP	52
16.2.	FTPS	59
16.3.	Command summary	67
17.	TCP and UDP	68
17.1.	UDP client	68
17.2.	TCP client	69
17.3.	Multiclient	71
17.4.	TCP server	73
17.5.	Command summary	76
18.	HTTP AND HTTPS	76
18.1.	http	76
18.2.	HTTPS.....	78
18.3.	Command summary	80

1. Introduction

The module counts also with an internal GPS what enables the location of the device outdoors and indoors combining standard NMEA frames with mobile cell ID triangulation using both assisted-mobile (A-GPS) and mobile-based (S-GPS) modes.

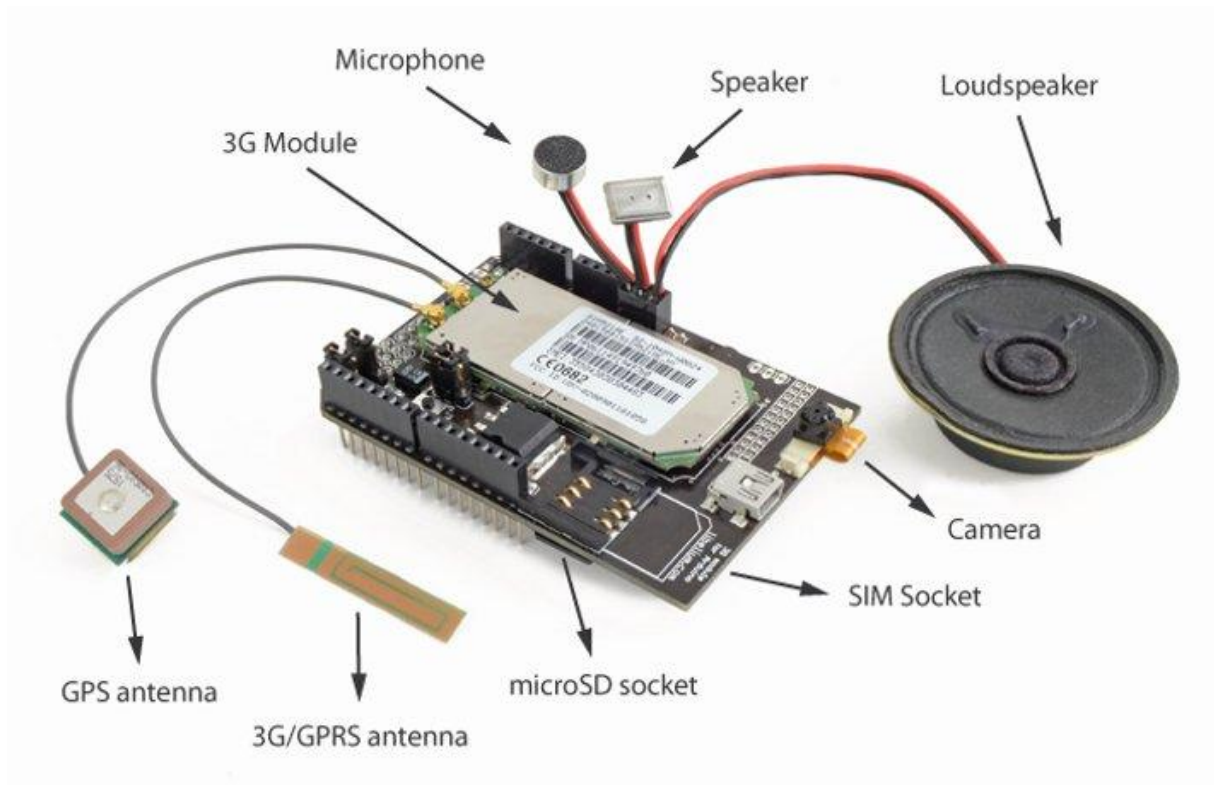
Other interesting accessories which can be connected to the module are a video camera which enables the record of video in high resolution (640x480), an audio kit including microphone, speaker, hands free and headphones sets and a SD socket to save directly all the data coming from the 3G network or recorded from the video camera. You can even reproduce audio files stored in the SD card (like a mp3 player!).

You can also use it as a standard 3G modem at full speed (~7.2Mbps download, ~5.5Mbps upload) just connecting it through its specific mini-USB socket to your laptop (Linux, Windows, MacOS).

The new communicating module is specially oriented to work with Internet servers implementing internally several application layer protocols which make easier to send the information to the cloud. We can make HTTP and HTTPS (secure mode) navigation, downloading and uploading content to a web server. In the same way FTP and FTPS (secure mode) protocols are also available which is really useful when your application requires handling files. You can even send and receive mails directly from Arduino using the SMTP and POP3 clients implemented internally.

With the SD Card socket so you can handle a complete FAT16 file systems and store up to 32GB of information. This specially useful as the 3G module can work at full speed (~7.2Mbps download, ~5.5Mbps upload) when working with the SD files directly without need of Arduino for data or files management.

The GPS module also makes possible perform geolocation services even in indoors as it can work in A-GPS and S-GPS modes, so the location given by the GPS through NMEA sentences is completed with the cell information provided by both the 3G module and external Internet Geoposition Servers which helps you to get the most accurate location in each case.



2. Features

- WCDMA and HSPA 3G networks compatibility
- Internal GPS for Assisted A-GPS and Supported S-GPS modes
- Video Camera (640x480) for video and photo recordings available
- Audio Kit including microphone, speaker, hands free and headphones available
- SD file system up to 32GB
- Works as a standard 3G modem in Linux/Windows/MacOS (~7.2Mbps download, ~5.5Mbps upload)
- Talk directly to web servers by HTTP/HTTPS (secure)
- Upload and download files directly by FTP/FTPS (secure)
- Send and receive mails by POP3/SMTP
- Play compressed audio files

3. Kit and accessories

Does make sense a 3G module for a low speed microcontroller device such as Arduino?

Definitively yes.

On the one hand you can use the Arduino UART at full speed ~ 115kbps, so the communication of data will be at the same bandwidth as if you had the Arduino connected directly to the USB directly (obviously with more initial latency). With the GPRS modules available in the market until now the speed limit was established by the GPRS network ~ 20kbps, so now we get 6 times more download performance than before. The upload speed the increment three times from 10kbps (GPRS) to 35kbps (3G).

On the other hand there is a bunch of actions can be made directly from the 3G module without having to go through the Arduino microcontroller. This means you will control all the action from the main program running in Arduino although the data may flow from the 3G module to the SD card or from the video camera to the 3G module without going through the UART avoiding a possible bottle neck and ensuring maximum speed performance given by your 3G carrier (~7.2Mbps download, ~5.5Mbps upload at max). Some of these independent actions include:

- Uploading and downloading files to HTTP and FTP servers directly from or to the SD
- Recording and uploading video and photos to the Internet using the video camera connected directly to the module
- Recording ambient audio and sending to a web server

These files may be also stored in the SD card of the 3G and in order to manage them after.

As well as this, the module offers theoretically also the possibility of performing Video-Calls. We say "theoretically" as the tests performed in Cooking Hacks we got a carrier error, the same we got when trying to make a Video-Call with our Smartphones so, although everything points it is is a carrier problem we cannot ensure the Video-Call functionality (if you get please let us know!).

As pointed before the 3G module can also be used as and standalone 3G/GPRS USB modem! It works in Linux, Windows and MacOS, just plug it through its specific USB connector to your laptop and follow the steps located in the current manual.

You can control also the kind of network service and the quality of the signal received (RSSI) from the surrounding cells you are using even change among them: GSM - GPRS - WCDMA - HSPA depending on the coverage of the zone. Get detailed information about the frequency band you are using to connect to the carrier and the quality of the signal.

4. Features

4.1. Electrical features

The 3G module used is the SIM5218 from SIMCOM. The power supply of SIM5218 is from a single voltage source of VBAT= 3.4v-4.2v. In some case, the ripple in a transmit burst may cause voltage drops when current consumption rise to typical peaks of 2A. So the power supply must be able to provide sufficient current up to 2A. The maximum rating are VBAT=4,4v and 3A peak. The maximum value of voltage for digital or analogy pins is 2.8v.

4.2. Radio features

SIM5218 is a quad-band GSM/GPRS/EDGE and UMTS engine that works on frequencies of GSM 850MHz, EGSM 900 MHz, DCS 1800 MHz, PCS1900 MHz, and WCDMA 2100M/1900M/850M.

Output power:

- UMTS 850/900/1900/2100: 0.25W
- GSM850/GSM900: 2W
- DCS1800/PCS1900: 1W

NOTE: when flight mode is activated RF functions are closed. The mode can be changed by putting the jumper in position (see 3G shield diagram below).

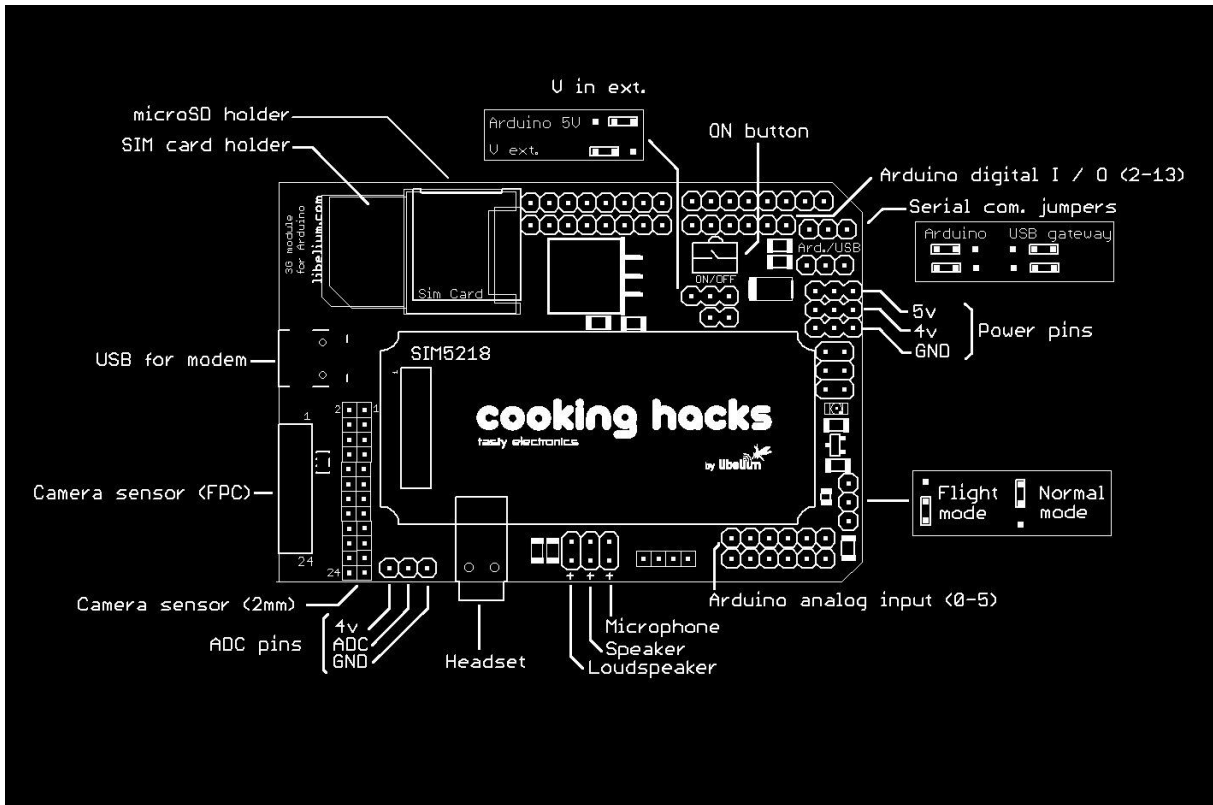
NOTE FOR US USERS: We have tested the new 3G shield with the AT&T network which supports natively the GSM and 3G protocols. With other carriers may also work although we haven't tried and thus we cannot ensure it. For this reason we recommend to use AT&T SIM cards.

4.3. Extra features

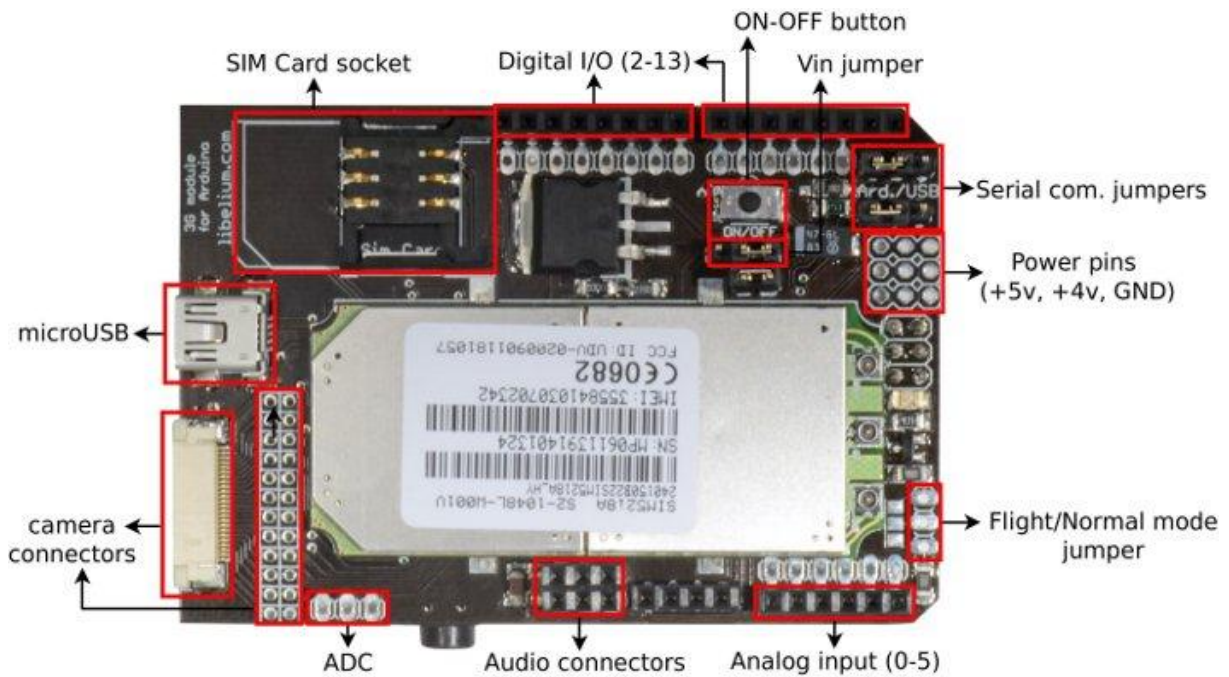
- USIM interface: support SIM cards: 3V & 1.8V
- High speed USB interface for modem
- Three audio channels include two microphones inputs and three audio outputs.
- A camera interface is provided in FPC connector and 2mm pitch holes.
- I2C interface is provided.
- 8-bit ADC input with an input voltage range between 2.65v and GND
- 2.8v LDO power output
- 4 bit microSD card interface
- 3 modes of GPS (stand alone, S-GPS y A-GPS)
- FTP and FTPS for managing files
- TCP and UDP sockets
- POP3 and SMTP for e-mail

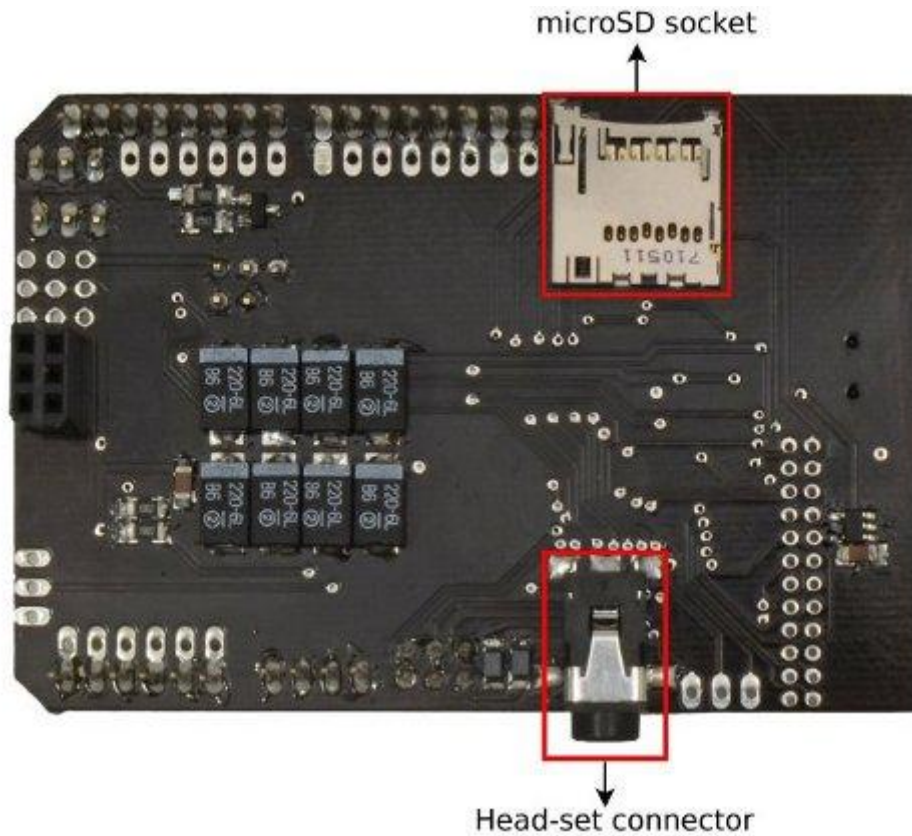
5. Hardware

5.1.3G Shield diagram



5.2.3G Shield PCB

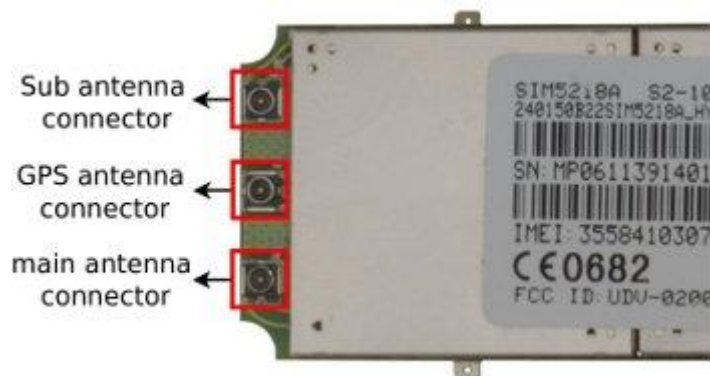




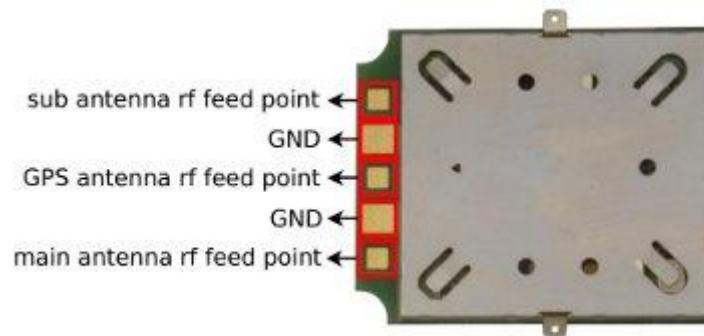
6. Antennas

6.1.Connections

SIM5218 has 3 UFL connectors: two for diversity of 3G mobile carriers and one for GPS antenna. The impedance of the RF interface is 50Ω. It is recommended use just the "main" antenna socket for the mobile connection unless you experience coverage or performance problems. In this case two antennas allowing diversity may be placed.



Also, the module allows to solder the antenna to the pad, or attach it via contact springs.



7. Bandwidth

3G module connected with Arduino allows downlinks rates over 115200 bauds (~11.5KBps), the maximum UART's speed and uplinks rates over 30000 bauds (~3KBps). Using the module as 3G USB modem we got speeds of 2Mbps (~222KBps) for downlink and 0.7Mbps (~77KBps) for uplink and. The connection speed may vary depending on the state of the network, the quality of the signal and the carrier.

8. Using 3G 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 SIM card and the USB cable.

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: If your 3G module doesn't answer to AT commands, maybe it is configured in a different baudrate. The command for change it temporarily is AT+IPR. Try different baudrates (300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 3200000, 3686400, 4000000) until you get to communicate. Once you can communicate, you can configure it in the baudrate you want.

9. Using 3G module like a modem



If you want to use the 3G module such a modem, please refer to the next link:

http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/3G_modem_tutorial.pdf

10. Audio

10.1. Audio features

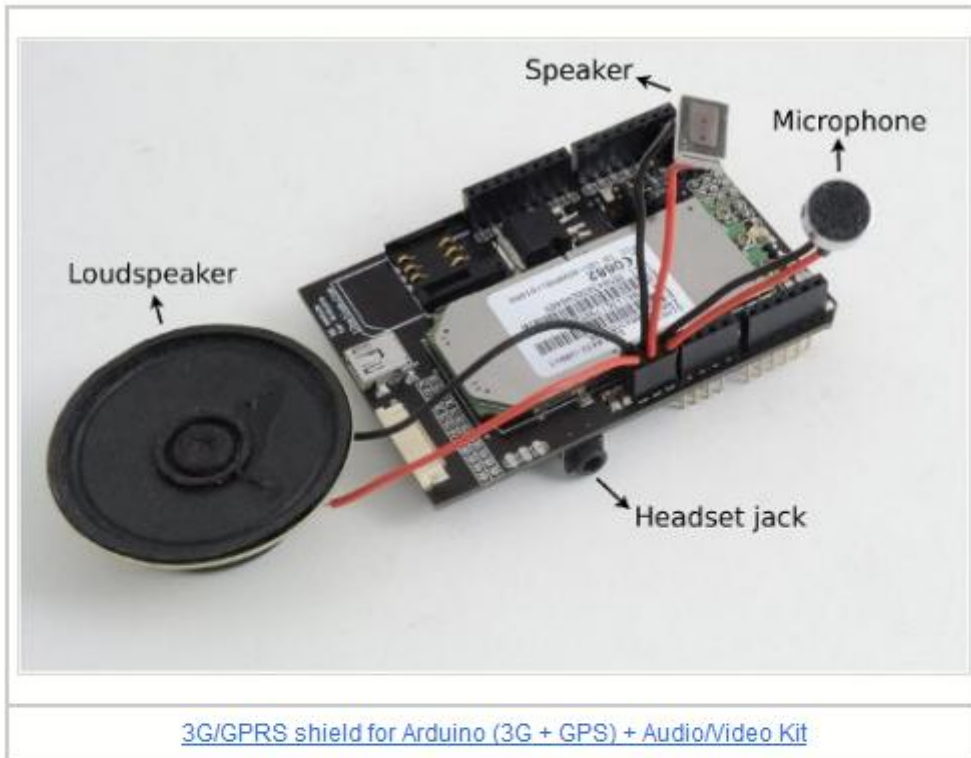
SIM5218 module provides two analog inputs for microphone and three analog outputs for two speakers and headphones. Depending on the selected audio channel (normal, headset and hands-free) a specific input and output are activated. The different channels are selected with the command AT+CSDVC.

Normal channel (AT+CSDVC=1) uses the main microphone and a speaker. Headset channel (AT+CSDVC=2) uses the jack connector. Hand-free channel (AT+CSDVC=3) uses the main microphone and the loudspeaker.

Both microphones don't need an external power source because they are powered directly by the module. You can use some extra commands in your codes like noise suppression, echo canceller or the gain of the mic amps. Please, refer to the AT command document list at the end of this article for more information.

10.2. Audio connectors

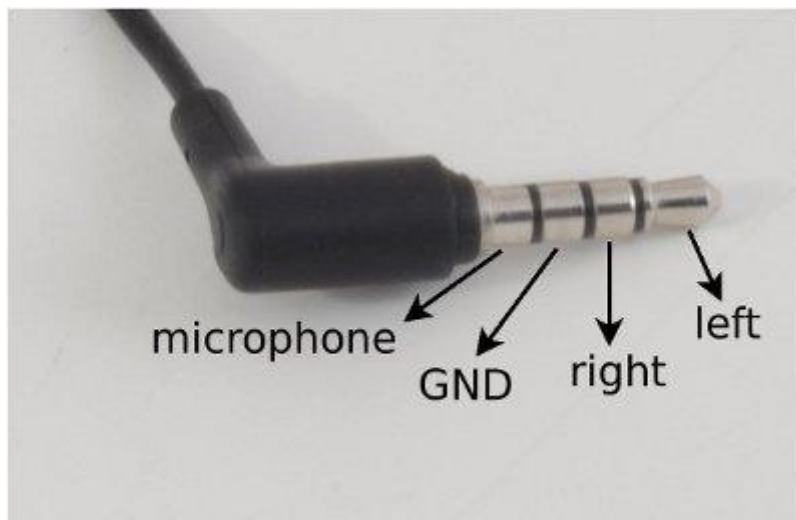
Audio connectors are situated at the bottom of the board. The two speakers and the main microphone are connected to board using pin connectors



Headset connectors are situated at the bottom too, between Arduino board and 3G board.



This connector has 4 terminals that correspond with the positive terminal of the microphone, left and right headphones and common ground. The order of connections is shown below :



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.

```

/*
 * 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

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){

        for(int x=0;x < 9;x++){           //generates the tones of the phone
number
            Serial.print("AT+CPTONE=");
            Serial.println(phone_number[x]);
        }
    }
}

```



```

        delay(400);
    }

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

    while(digitalRead(button)==1);

    Serial.println("AT+CHUP");           // disconnects the existing
call

    delay(5000);

    count++;
}
}

```

To make a lost call next code is used.

```

/*
 * 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 = 2;                //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){
        delay(1500);
        for(int x=0;x < 9;x++){           //generates the tones of the phone
number
            Serial.print("AT+CPTONE=");
            Serial.println(phone_number[x]);
            delay(400);
        }

        Serial.print("ATD");
        Serial.print(phone_number);
        Serial.println(";");
        delay(6000*tones);
        Serial.println("AT+CHUP");        // 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.

```

/*
 * 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

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

boolean check_response(char *str){

    int i = 0;
    int j = 0;
    char b[10];

    // reading the message and cutting
    // initials and finals
    while(Serial.available() > 0){
        if(j==0 || j==1){
            Serial.read();
            j++;
        }
        else if( Serial.peek() == 13 ){
            b[i] = 0;
            Serial.read();
        }
        else if( Serial.peek() == 10 ){
            Serial.read();
        }
        else {
            b[i++] = Serial.read();
        }
    }

    // comparing
    int cont=0;
    int cont2=0;
    boolean ok=false;
    while(b[cont]!=-1){
        if (b[cont]==str[cont2]){
            while(str[cont2]!=-1){
                if (b[cont+cont2]!=b[cont2])
                {
                    cont2++;
                    ok=true;
                }
                else {
                    ok=false;
                    break;
                }
            }
        }
        if (ok) break;
        cont++;
    }
}

```

```

    if(ok){
        return true;
    }

    return false;
}

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(!check_response("RING")); //waits for a call
    Serial.println("ATA");          //answers an incoming call
    delay(10000);

    while(digitalRead(button)==1);

    Serial.println("AT+CHUP");     // disconnects the existing call
}

```

The channel may be changed during a call. Use the command AT+CSDVC to select the channel that you want.

10.3. Recording and playing sound

With the next code we can record a sound, stores it and before play it. You can select the place to store the sound clip: the local storage or a SD card. Once saved the sound can be sent to an FTP or FTPS as you'll see later.

```

/*
 * 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 x = 0;

char name[20];

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

void setup(){

    for(x=0;x < 20;x++){
        name[x]='\0';
    }

    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+FSLOCA=0");
    // store media files to local storage space, put 1 to store media files
to storage card
    while(Serial.read()!='K');

    Serial.println("AT+CNSM=1");   // Enables noise supression
    while(Serial.read()!='K');

}

void loop(){

    //RECORDING A SOUND CLIP
    delay(1500);
    while(Serial.available() !=0){
        Serial.read();
    }
    Serial.println("AT+CQCPREC=0,amr"); // starts to record a sound
clip
    Serial.flush();
    while(Serial.read() !='/');
}

```

```

while (Serial.read() != '/');

x=0;
do{
  while (Serial.available() == 0);
  name[x]=Serial.read();
  x++;
}while(x < 19);

while (Serial.read() != 'K');
Serial.println(name);
delay(10000);

Serial.println("AT+CQCPAUSE"); // pauses record sound
while (Serial.read() != 'K');
delay(2000);

Serial.println("AT+CQCPRESUME"); // resumes record sound
while (Serial.read() != 'K');
delay(10000);

Serial.println("AT+CQCPSTOP"); // stops to record a sound clip
while (Serial.read() != 'K');

//PLAY A SOUND CLIP
Serial.println("AT+FSCD=Audio");
Serial.flush();
while (Serial.read() != 'K');

Serial.print("AT+CCMXPLAY=\""); //plays an audio file
Serial.print(name);
Serial.println("\"");
Serial.flush();
while (Serial.read() != 'K');
delay(7000);

Serial.println("AT+CCMXPAUSE"); // pauses playing audio file
while (Serial.read() != 'K');
delay(2000);

Serial.println("AT+CCMXRESUME"); // resumes playing audio file
while (Serial.read() != 'K');
delay(8000);

Serial.println("AT+CCMXSTOP"); // Stops playing audio file
while (Serial.read() != 'K');

while(1);
}

```

Also, you can use the 3G module like a mp3 player! First, you must save your music in mp3, amr or qcp format. You can use Audacity (linux) or Mobile Media Converter (Windows or Linux) to do this. Then, save the music in the Audio folder on microSD card. Now, put microSD card into the shield and enjoy the music!

NOTE: You will need to **update your firmware** to play mp3. The update is available [here](#). Uncompress and run it in a PC with Windows (emulated Windows not supported).

A voice call can be recorded using these same commands too.

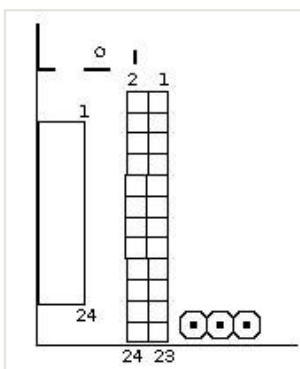
10.4. Command summary

Command	Response	Description
AT+CPTONE=**	OK	Plays a DTMF tone or complex tone on local voice channel device . ** is the number of the tone.
ATD*****;		***** is the number to call.
ATA	OK	Answer an incoming call.
AT+CHUP	OK	Cancel voice calls.
AT+CNSM	OK	Enable/disable noise suppression.
AT+CQCPREC=*,&&&		Starts recording sound clips . Answers with the path and the name of the clip. * is the path and &&& is the format.
AT+CQCPPAUSE	OK	Pauses record sound.
AT+CQCPRESUME	OK	Resumes record sound.
AT+CQCPSTOP	OK	Stops record sound.
AT+CCMXPLAY="*****"	OK	Plays an audio file . ***** is the name of the file.
AT+CCMXPAUSE	OK	Pauses playing audio file.
AT+CCMXRESUME	OK	Resumes playing audio file.
AT+CCMXSTOP	OK	Stops playing audio file.

11. Camera

11.1. Camera features

This module allows connecting a camera for video recording and taking photos. The maximum resolution for images are 640x480 (VGA) and 320x240 (QVGA) with 15 fps for video. Once saved the video or image file can be sent to an FTP or FTPS as you'll see later. The pin diagram shows below:



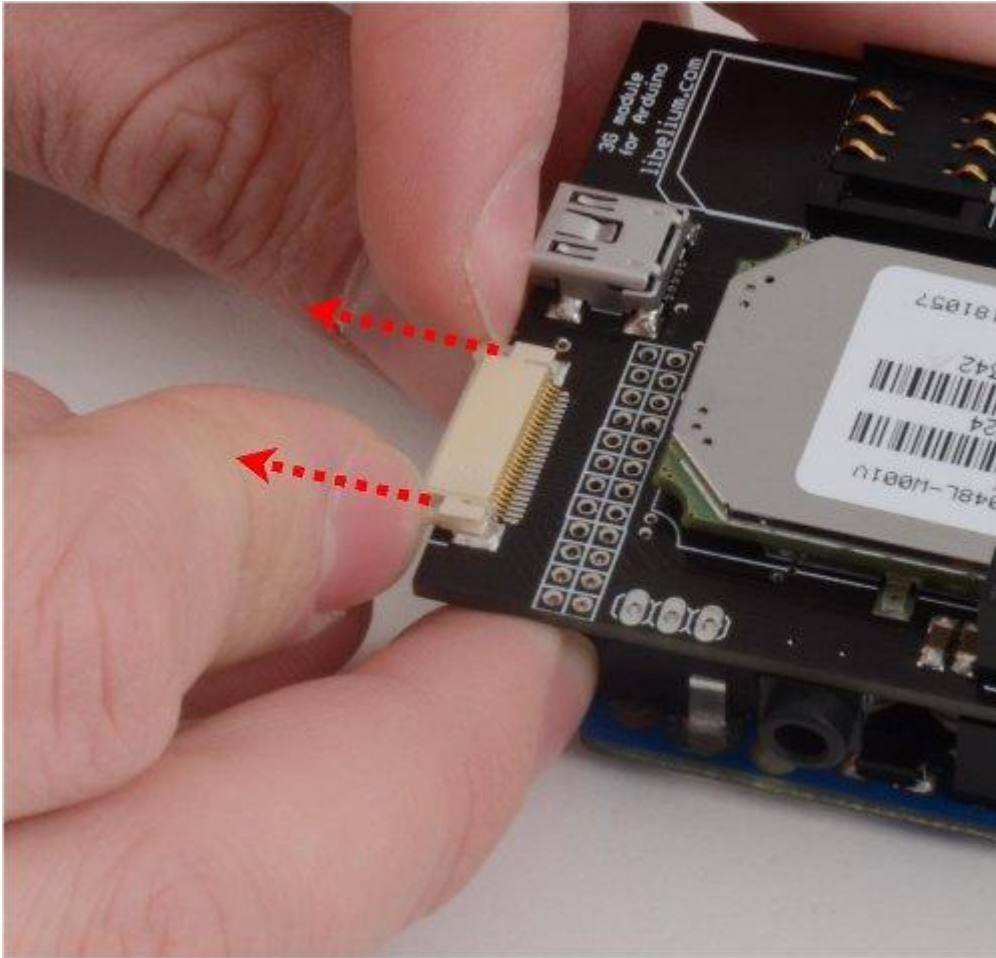
Pin n°	Name	Pin n°	Name	Pin n°	Name
1	NC	9	HSYNC	17	PCLK
2	AGND	10	DVDD	18	DATA6
3	SDA	11	DOVDD	19	DATA2
4	AVDD	12	DATA9	20	DATA5
5	SCL	13	MCLK	21	DATA3
6	RESET	14	DATA8	22	DATA4
7	VSYNC	15	AGND	23	DATA1
8	PWDN	16	DATA7	24	DATA0



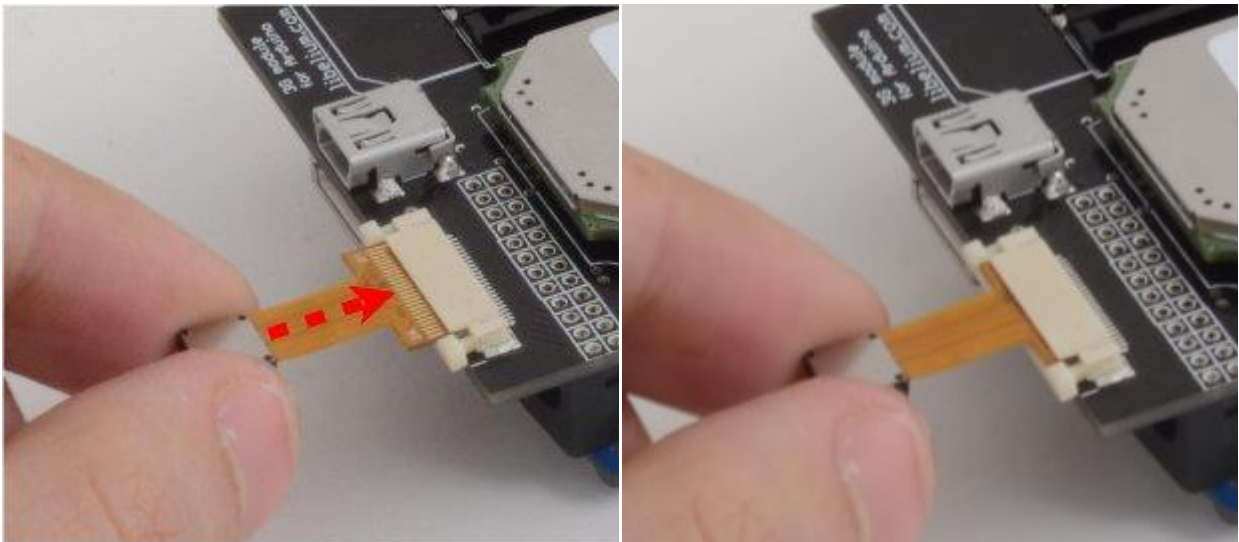
Mounting the camera



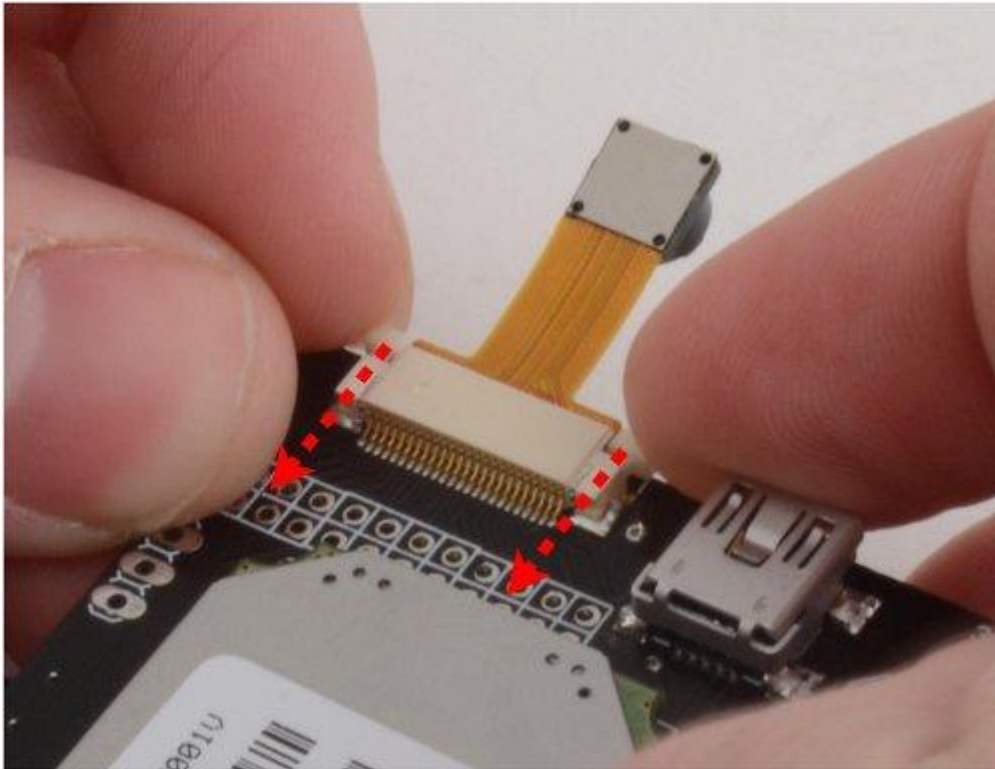
Follow these simple steps to mount the camera into the 3G Arduino board.



The first step is to open the socket. To do this, pull carefully out the sides of the connector.



Inserts the camera with metallic contacts facing up



At the final step, push in the laterals of the connector

11.2. Video recording

To record video, and take images, you must start the camera in the first step. Then you can configure some parameters like resolution, fps, rotation or zoom. The following code allows recording a video file in mp4 format. Remember, if you want to store the video file in a SD card, you must to use AT+FSLOCA command.

```
/*
 * 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 x = 0;

char name[20];

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+CCAMS");    //starts the camera
    while(Serial.read()!='K');

    Serial.println("AT+CCAMSETD=320,240");           //sets QVGA (320*240)
resolution
    while(Serial.read()!='K');

    Serial.println("AT+CCAMSETF=2");           //sets 15 frames per second
    while(Serial.read()!='K');

}

void loop(){

    delay(1500);

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

    Serial.println("AT+CCAMRS");           //starts to record a video
    Serial.flush();                       //video is saved into C:/Video
    while(Serial.read() !='/');
    while(Serial.read() !='/');

    x=0;
    do{
        while(Serial.available() ==0);
        name[x]=Serial.read();
        x++;
    }while(x < 19);

    while(Serial.read() !='K');

```

```

Serial.println(name);
delay(10000);

Serial.println("AT+CCAMRP"); // pauses record
while(Serial.read()!='K');
delay(2000);

Serial.println("AT+CCAMRR"); // resumes record
while(Serial.read()!='K');
delay(10000);

Serial.println("AT+CCAMRE"); // stops to record
while(Serial.read()!='K');

Serial.println("AT+CCAME"); // stops the camera
while(Serial.read()!='K');

while(1);
}

```

11.3. Taking photos

Taking photos is very easy, here is the example 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 x = 0;

char name[20];

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+CCAMS");    //starts the camera
    while(Serial.read()!='K');

    Serial.println("AT+CCAMSETD=640,480");    //sets VGA (640*480)
resolution
    while(Serial.read()!='K');

    Serial.println("AT+FSLOCA=0");    //stores the image file in the 3G
module
    while(Serial.read()!='K');

}

void loop(){

    delay(1500);
    while(Serial.available() !=0){
        Serial.read();
    }
    Serial.println("AT+CCAMTP");    //takes a picture, but not saved it
    while(Serial.read()!='K');

    Serial.println("AT+CCAMEP");    // saves the picture into C:/Picture
    Serial.flush();
    while(Serial.read() !='/');
    while(Serial.read() !='/');

    x=0;
    do{
        while(Serial.available() ==0);
        name[x]=Serial.read();
        x++;
    }while(x < 19);

    while(Serial.read() !='K');
    Serial.println(name);

    Serial.println("AT+CCAME");    // stops the camera
    while(Serial.read() !='K');

    while(1);

}

```

11.4. Video call

The SIM5218 allows video calls, but to carry them out correctly the operator and the network must be able to allow it. The example code is below. Cooking Hacks not ensure that the video call functions correctly.

```
/*
 * 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 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+CCAMS");   //starts the camera
    while(Serial.read()!='K');

    Serial.println("AT+CCAMSETD=320,240"); //sets QVGA (320*240)
resolution
    while(Serial.read()!='K');

    Serial.println("AT+CCAMSETF=2"); //sets 15 frames per seconds
    while(Serial.read()!='K');
```

```

}

void loop(){

    delay(1500);

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

    Serial.print("AT+VPMAKE="); //inits a videocall
    Serial.println(phone_number);
    delay(30000);

    Serial.println("AT+VPEND"); // ends the videocall
    while(Serial.read() != 'K');

    Serial.println("AT+CCAME"); // stops the camera
    while(Serial.read() != 'K');

    while(1);

}

```

11.5. Command summary

Command	Response	Description
AT+CCAMS	OK	Starts camera.
AT+CCAME	OK	Stops camera.
AT+CCAMSETD=xxx,yyy	OK	Sets dimension of camera. xxx is the width and yyy is the height.
AT+CCAMSETF=*	OK	Sets the frames per second. * is the frame rate option.
AT+CCAMRS	OK	Starts video recording . Also responds with the path and name of the file.
AT+CCAMRP	OK	Pauses the record.
AT+CCAMRR	OK	Resumes video record.
AT+CCAMRE	OK	Stops video record.
AT+CCAMTP	OK	Takes a picture.
AT+CCAMEP	OK	Saves a picture taken by last AT+CCAMTP. Also responds with the path and name of the file.
AT+VPMAKE		Makes a video call.
AT+VPEND		Ends a video call.

12. GPS

SIM5218 supports both A-GPS and S-GPS and provides three operating modes: mobile-assisted mode, mobile-based mode and standalone mode. A-GPS is include mobile-assisted and mobile-based mode.

In mobile-assisted mode, when a request for position location is issued, available network information is provided to the location server (e.g., Cell-ID) and assistance is requested from the location server. The location server sends the assistance information to the handset. The handset/mobile unit measures the GPS observables and provides the GPS measurements along with available network data (that is appropriate for the given air interface technology) to the location server. The location server then calculates the position location and returns results to the requesting entity.

In mobile-based mode, the assistance data provided by the location server encompasses not only the information required to assist the handset in measuring the satellite signals, but also the information required to calculate the handset's position. Therefore, rather than provide the GPS measurements and available network data back to the location server, the mobile calculates the location on the handset and passes the result to the requesting entity.

In standalone (autonomous) mode, the handset demodulates the data directly from the GPS satellites. This mode has some reduced cold-start sensitivity, and a longer time to first fix as compared to the assisted modes. However, it requires no server interaction and works out of network coverage.

This combination of GPS measurements and available network information provides:

- High-sensitivity solution that works in all terrains: indoor, outdoor, urban, and rural
- High availability that is enabled by using both satellite and network information

Therefore, while network solutions typically perform poorly in rural areas and areas of poor cell geometry/density, and while unassisted, GPS-only solutions typically perform poorly indoors, the SIM5218 GPS solution provides optimal time to fix, accuracy, sensitivity, availability, and reduced network utilization in both of these environments, depending on the given condition.

12.1. Stand-alone mode

In this mode, the GPS obtains position, altitude,... with only the signal of the satellites, making it the slowest of the three modes. If you use the AT+CGPSINFO command, the module bring directly latitude, longitude, date, UTC time, altitude and speed. Here is the example 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[255];
char latitude[11],longitude[12];
char date[6],UTC_time[8];
char speed_OG[6],altitude[6];

int x,y;

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+CGPS=1,1"); // starts GPS session in stand-
alone mode
    delay(100);
}

void loop(){

    delay(5000);
    Serial.println("AT+CGPSINFO"); // request GPS info
    Serial.flush();

    x=0;
    do{
        do{
            digitalWrite(led,HIGH);
        }
        while(Serial.available()==0);
        digitalWrite(led,LOW);
        data[x]=Serial.read();
        x++;
    }
    while(Serial.read()!='K');

    x=24;
    y=0;
}

```

```

if(data[x]!=','){
  Serial.print("Latitude: ");    //shows actual latitude
  do{
    latitude[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
  }
  while(data[x]!=',');
  x++;
  Serial.print(" ");
  Serial.println(data[x]);    //north or south

  x+=2;
  y=0;
  Serial.print("Longitude: ");    //shows actual longitude
  do{
    longitude[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
  }
  while(data[x]!=',');
  x++;
  Serial.print(" ");
  Serial.println(data[x]);    //west or east

  x+=2;
  y=0;
  Serial.print("Date: ");    //shows date
  do{
    date[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
  }
  while(data[x]!=',');
  x++;

  y=0;
  Serial.print("\r\nUTC time: ");    //shows UTC time
  do{
    UTC_time[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
  }
  while(data[x]!=',');
  x++;

  y=0;
  Serial.print("\r\nAltitude: ");    //shows altitude
  do{
    UTC_time[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
  }
  while(data[x]!=',');
  Serial.println(" m");
  x++;
}

```

```

    y=0;
    Serial.print("Speed: ");    //shows speed
    do{
        speed_OG[y]=data[x];
        Serial.print(data[x]);
        y++;
        x++;
    }
    while(data[x]!=0x0D);
    Serial.println(" knots");
}
else{
    Serial.println("GPS information not available, please wait...");
}
}
}

```

If you want to get the NMEA sentences, you must to use the command AT+CGPSSWITCH=1 and configure the UART port at 57600 bauds. Supported NMEA sentences include GSV, GGA, RMC, GSA, and VTG.

12.2. S-GPS mode

For the S-GPS the module connects to a GPS server and the module calculates the position. Here is the example 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[255];
char latitude[11], longitude[12];
char date[6], UTC_time[8];
char speed_OG[6], altitude[6];

```

```

int x,y;

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+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!check_response("OK"));

    Serial.println("AT+CGPSURL=\"supl.google.com:7276\");           // sets
GPS server
    delay(100);
    while(Serial.read()!='K');

    Serial.println("AT+CGPSSSL=0");           // without certificate
    delay(100);
    while(Serial.read()!='K');

    Serial.println("AT+CGPS=1,2");           // starts GPS session in based
mode
    delay(100);
    while(Serial.read()!='K');
}

void loop(){

    delay(5000);
    Serial.println("AT+CGPSINFO"); // request GPS info
    Serial.flush();
    for (x=0;x< 255;x++){
        data[x]='\0';
    }
    x=0;
    do{
        do{
            digitalWrite(led,HIGH);
        }
        while(Serial.available()==0);
    }
}

```

```

    digitalWrite(led,LOW);
    data[x]=Serial.read();
    x++;
}
while(Serial.read()!='K');

x=24;
y=0;
if(data[x]!=','){
  Serial.print("Latitude: ");    //shows actual latitude
  do{
    latitude[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
    if(y==2){
      Serial.print(0xF8);
    }

    if(y==4){
      Serial.print(".");
    }
  }
  while(data[x]!=',');
  x++;
  Serial.print(" ");
  Serial.println(data[x]);    //north or south

  x+=2;
  y=0;
  Serial.print("Longitude: ");    //shows actual longitude
  do{
    longitude[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
    if(y==3){
      Serial.print(0xF8);
    }

    if(y==5){
      Serial.print(".");
    }
  }
  while(data[x]!=',');
  x++;
  Serial.print(" ");
  Serial.println(data[x]);    //west or east

  x+=2;
  y=0;
  Serial.print("Date: ");    //shows date
  do{
    date[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
  }
  while(data[x]!=',');
  x++;

```

```

y=0;
Serial.print("\r\nUTC time: ");    //shows UTC time
do{
    UTC_time[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
}
while(data[x]!='\n');
x++;

y=0;
Serial.print("\r\nAltitude: ");    //shows altitude
do{
    UTC_time[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
}
while(data[x]!='\n');
Serial.println(" m");
x++;

y=0;
Serial.print("Speed: ");    //shows speed
do{
    speed_OG[y]=data[x];
    Serial.print(data[x]);
    y++;
    x++;
}
while(data[x]!=0x0D);
Serial.println(" knots");
}
else{
    Serial.println("GPS information not available, please wait...");
}
}

```

12.3. A-GPS mode

For the A-GPS the module connects to a GPS server for calculate the position. Here is the example code:

MS-assisted	Server	Module
Location server sends aiding data that is valid for the current fix	Sending data	
Module sends code phases		Code phases
Server calculates position	Calculate position	
MS-based	Server	Module
Location server sends aiding data that is valid for the current fix	Send aiding data	
Module calculates		Calculate position
Stand alone	Server	Module
Module demodulates data form GPS satellite		Demodulates GPS satellite data
Module calculates position		Calculate position

12.4. Command summary

Command	Response	Description
AT+CGSOCKCONT=**,&&&	OK	** is the protocol and &&& Access Point Name.
AT+CGPS=*,&	OK	* sets on (1) or off (0) and & is the GPS mode.
AT+CGPSINFO		Gets current position information.
AT+CGPSURL=****	OK	Sets AGPS default server URL . *** is the URL.
AT+CGPSSSL=*	OK	Select transport security, used certificate (*=1) or not (*=0).
AT+CGPSSWITCH	OK	Choose the output port for NMEA sentence.

13. SMS

13.1. Sending and receiving SMS

The first code is used to send a SMS, the second one reads the SIM message memory and shows the last SMS.

```

/*
 * 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("\"");
        delay(1500);
        Serial.print("Hola caracola..."); // the SMS body
        delay(500);
        Serial.write(0x1A); //sends ++
        Serial.write(0x0D);
        Serial.write(0x0A);

        delay(5000);

        count++;
    }
}

```

```

/*
* 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(100);
}

void loop(){

    while (count < timesToSend){
        delay(1500);
        Serial.println("AT+CPMS=\"SM\", \"SM\", \"SM\");           //selects SIM
memory
        Serial.flush();
        for (x=0;x< 255;x++){
            data[x]='\0';

```

```

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

    }while (!(data[x-1] == 'K' && data[x-2] == 'O'));

    n_sms = ((data[32] - 0x30) * 10) + (data[33] - 0x30);

    Serial.print(n_sms, DEC);
    Serial.println(" SMS stored in SIM memory. Showing last SMS:");
    // shows the total number of SMS and the last SMS

    Serial.print("AT+CMGR="); // Reads the last SMS
    Serial.println(n_sms-1, DEC);
    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] == 'O'));

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

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

    delay(5000);

    count++;
}
}

```

13.2. 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+CPMS=***		Selects memory storages. *** is the memory type.
AT+CMGR=*		Reads a message. * is the number of the message.

14. E-mails

14.1. Sending an e-mail

To send e-mail a SMTP server is used, so we need an e-mail account. The code example is the next:

```
/*
 * 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 x;

//server and account data
char server[ ]="";
char user_name[ ]="";
char password[ ]="";
char port[ ]="";

//information about sender, direccctions and names
char sender_add[ ]="";
char sender_name[ ]="";

char to_add[ ]="";
char to_name[ ]="";

char cc_add[ ]="";
char cc_name[ ]="";

char bcc_add[ ]="";
char bcc_name[ ]="";

//
char subject[ ]="";
char body[ ]="";

char response[128];

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(){

    Serial.print("AT+SMTPSRV=\"");
    //configures the SIM5218 with server name and port
    Serial.print(server);
    Serial.print "\",");
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.print("AT+SMTPAUTH=1,\""); //configures the SIM5218 with user
name and password
    Serial.print(user_name);
    Serial.print "\",\"");
    Serial.print(password);
    Serial.println("\"");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.print("AT+SMTPFROM=\""); //sender adress and name
    Serial.print(sender_add);
    Serial.print "\",\"");
    Serial.print(sender_name);
    Serial.println("\"");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);

```

```

        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.print("AT+SMTPRCPT=0,0,\"");    //normal recipient address and
name
    Serial.print(to_add);
    Serial.print("\",\"");
    Serial.print(to_name);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    //If you don't need a carbon copy, delete this part
    Serial.print("AT+SMTPRCPT=1,0,\"");    //carbon copy recipient address
and name
    Serial.print(cc_add);
    Serial.print("\",\"");
    Serial.print(cc_name);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    //If you don't need a blind carbon copy, delete this part
    Serial.print("AT+SMTPRCPT=2,0,\"");    //blind carbon copy recipient
address and name
    Serial.print(bcc_add);
    Serial.print("\",\"");
    Serial.print(bcc_name);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.print("AT+SMTPSUB=\"");    //subjec of email
    Serial.print(subject);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);

```

```

        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.print("AT+SMTPBODY=\"");    //body of email
    Serial.print(body);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.println("AT+SMTPSEND");    //sends the email
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='K'&&response[x-2]=='O'));

    Serial.print("Sending email...");

    x=0;
    do{
        do{
            digitalWrite(led,HIGH);
        }
        while(Serial.available()==0);
        digitalWrite(led,LOW);
        response[x]=Serial.read();
        x++;
    }
    while(!(response[x-1]=='S'&&response[x-2]=='S'));

    Serial.println(response);

    while(1);
}

```

14.2. Receiving an e-mail

For read an e-mail from a POP3 server we need an e-mail account. The code example is here:

```
/*
 * 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 x;

//server and account data
char server[ ]="";
char user name[ ]="";
char password[ ]="";
char port[ ]="";

char data[1024];
char path[50];

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+CGSOCKCONT=1,\"IP\", \"myapn\");
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop(){

Serial.print("AT+POP3SRV=\");
//configures the SIM5218 with server name, user, password and port
Serial.print(server);
Serial.print("\",\");
Serial.print(user_name);
Serial.print("\",\");
Serial.print(password);
Serial.print("\",");
Serial.println(port);
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}
while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+POP3IN"); //logs into the server
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+POP3GET=1"); //gets the first email
Serial.flush();

do{
    while(Serial.available()==0);
    path[0]=Serial.read();
}while(path[0]!=' /');

do{
    while(Serial.available()==0);
    path[0]=Serial.read();
}
while(path[0]!=' /');

x=0;
do{
    while(Serial.available()==0);
    path[x]=Serial.read();
    x++;
}while(path[x-1]!=' ');

```



```

x--; //overwrite " " charracter
x--; //overwrite "," charracter

do{
    while(Serial.available()==0);
    path[x]=Serial.read();
    x++;
}while(path[x-1]!=0x0D);
path[x-1]='\0';

Serial.println("AT+POP3OUT"); //logs out the server
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+FSCD=C:/Email"); //selects email directory as
current directory
Serial.flush();

x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.print("AT+POP3READ=0,\""); //shows the email
Serial.print(path);
Serial.println("\");
Serial.flush();
do{
    while(Serial.available()==0);
    data[0]=Serial.read();
}while(data[0]!='D');

do{
    while(Serial.available()==0);
    data[0]=Serial.read();
}while(data[0]!='D');

x=1;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
    if(x==1020){
        Serial.print(data);
        x=0;
    }
}while(!(data[x-1]=='K'&&data[x-2]=='O'&&data[x-3]==0x0A&&data[x-4]==0x0D));

data[x-2]='\0';
Serial.println(data);

while(1);}

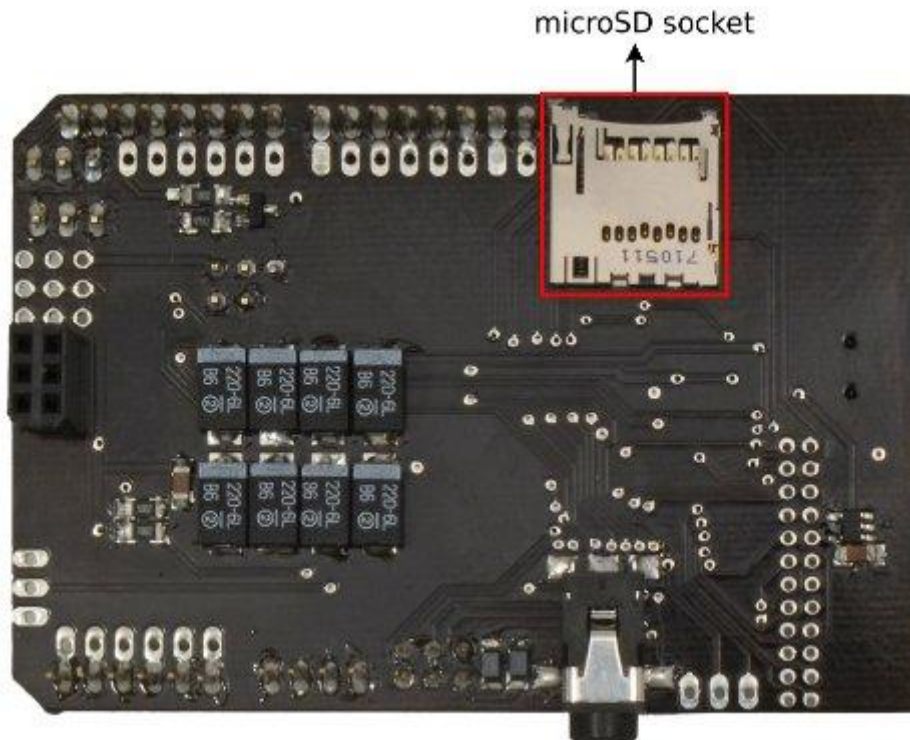
```

14.3. Command summary

Command	Response	Description
AT+CGSOCKCONT=**,&&&	OK	** is the protocol and &&& Access Point Name.
AT+SMTPSRV="*****",&&&	OK	Sets SMTP server address and server's port . ***** is the sever address and &&& is the server's port.
AT+SMTPAUTH=1,"****",&&&"	OK	Controls SMTP authentication. *** is the user name and &&& is the password.
AT+SMTPFROM="****",&&&"	OK	Sets sender's address and nam e. *** is the sender's address and &&& is the sender's name.
AT+SMTPRCPT=x,"****",&&&"	OK	Sets recipient address/name and kind . x is the kind (TO/CC/BCC), *** is the recipient address and &&& is the recipient name.
AT+SMTPSUB="*****"	OK	Sets the subject of e-mail . **** is the subject.
AT+SMTPBODY		Sets e-mail body.
AT+SMTPSEND		Initiates TCP session with SMTP server and sends an e-mail
AT+POP3SRV="**",&&","xx",yy	OK	Sets all parameters to get an e-mail from POP3 . ** is the server address, && is the user name, xx is the password and yy the server port
AT+POP3IN	OK	Logs in POP3 server
AT+POP3GET=***		Gets an e-mail from POP3 server , *** is the number of the e-mail.
AT+POP3OUT	OK	Logs out POP3 server
AT+POP3READ="****",&&&"		Reads an e-mail from file system . *** is the location and &&& is the name.

15. SD and system memory

The file system is used to store files in a hierarchical (tree) structure. Local storage space is mapped to "C:", and SD is mapped to "D:". In both "C:" and "D:" directories, module creates four directories named "Picture", "Audio", "Video" and "VideoCall" automatically; "Picture" is used to store static image when taking picture by camera, "Audio" is used to store audio file, "Video" is used to store video file when recor ding by camera, and "VideoCall" is used to store media file which is recorded during a video call.



The maximum size of the microSD card is 32GB. Here are some AT commands to manage the file system:

AT command	Response	Description
AT+FSKD=****	OK	Select a directory as current directory . **** is the directory.
AT+FSMKDIR=****	OK	Create a new directory in current directory . **** is the name of the new directory.
AT+FSRMDIR=****	OK	Delete existing directory in current directory . **** is the name of the directory to delete.
AT+FSLS		Make a list with informations of directories and/or files in current directory.
AT+DEL=****	OK	Delete a file in current directory. **** is the name of the file to delete.
AT+FSRENAME=****,&&&	OK	Rename a file in current directory. **** is the actual name and &&& is the new name.
AT+FSATTRI=****		Request the attributes of file which is existing in current directory. **** is the name of the file.
AT+FSMEM		Check the size of available memory.
AT+FSFMT	OK	Format storage card which is plugged in.
AT+FSLOCA=	OK	Set the storage place , 0 for local and 1 for SD

16. FTP and FTPS

16.1. FTP

Using the following code, you'll be able to create a file on a FTP server, write and read the file.

```
/*
 * 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 file_size;

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

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+CGSOCKCONT=1,\"IP\", \"myapn\");
```

```

Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop()
{
    Serial.print("AT+CFTPSERV=\""); //Sets the FTP server
Serial.print(server);
Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPPORT="); //Sets FTP port
Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPUN=\""); //Sets the user name
Serial.print(user_name);
Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPPW=\""); //Sets password
Serial.print(password);
Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.println("AT+CFTPMODE=1"); //Selects pasive mode
Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();

```

```

    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPTYPE=A");    //Selects ASCII mode
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPPUT=\"/test.txt\"");    //Creates a file and
sends data (ASCII)
Serial.flush();
while(Serial.read()!='N');    //Waits 'N' from BEGIN

Serial.println("01234567890123456789");    //Data for the file
Serial.write(0x1A);    //EOL character
Serial.write(0x0D);
Serial.write(0x0A);

x=0;

do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

delay(1000);

Serial.println("AT+CFTPGET=\"/test.txt\"");    //Reads the data from
test file
Serial.flush();

x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while((data[x-1]!=','));

x=0;

while(Serial.peek()!=0x0A){
    file_size=(file_size*10)+(Serial.read()-0x30);
};

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

Serial.print(data);

while(1);
}

```

Uploading a file from local storage or SD to FTP server.

```
/*
 * 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[256];
int x=0;

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

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+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
    }
}
```

```

        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop()
{
    Serial.print("AT+CFTPSERV=\""); //Sets the FTP server
    Serial.print(server);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPPORT="); //Sets FTP port
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPUN=\""); //Sets the user name
    Serial.print(user_name);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPPW=\""); //Sets password
    Serial.print(password);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.println("AT+CFTPMODE=1"); //Selects pasive mode
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    //Selecciona el tipo ASCII

```



```

Serial.println("AT+CFTPTYPE=A"); //Selects ASCII mode
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPPUTFILE=\"/testfile.txt\",0"); //Uploads a
test file into FTP server.
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

while(1);
}

```

Downloading a file from FTP server or SD to local storage.

```

/*
 * 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;

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

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+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop()
{

    Serial.print("AT+CFTPSERV=\""); //Sets the FTP server
    Serial.print(server);
    Serial.println("\"");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPPORT="); //Sets FTP port
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPUN=\""); //Sets the user name
    Serial.print(user_name);
    Serial.println("\"");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

```

```

    Serial.print("AT+CFTPPW=\""); //Sets password
Serial.print(password);
Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.println("AT+CFTPMODE=1"); //Selects pasive mode
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    //Selecciona el tipo ASCII
    Serial.println("AT+CFTPTYPE=A"); //Selects ASCII mode
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    //Envia data
    Serial.println("AT+CFTPGETFILE=\"/testfile.txt\",0"); //Downloads a
test file from FTP server.
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    while(1);
}

```

16.2. FTTPS

Creating a file into the FTP server, writing it and reading it (likes FTP codes).

```

/*
 * 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;
unsigned int file_size=0;

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

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+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }

```

```

    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop()
{
    Serial.println("AT+CFTPSSTART");          //Acquires FTPS protocol stack
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPSLOGIN=\"");        //Logs into FTPS server
    Serial.print(server);
    Serial.print("\",");
    Serial.print(port);
    Serial.print(",\");
    Serial.print(user_name);
    Serial.print("\",\");
    Serial.print(password);
    Serial.println("\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.println("AT+CFTPMODE=1");         //Selects pasive mode
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.println("AT+CFTPSTYPE=A");        //Selects ASCII mode
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    //Envia data
    Serial.println("AT+CFTPSPUT=\"/test_file.txt\",13");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(data[x-1]!='>');
}

```

```

Serial.write("Test message!");

x=0;
do{
  while(Serial.available()==0);
  digitalWrite(13, LOW);
  data[x]=Serial.read();
  x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

delay(1000);

Serial.println("AT+CFTPSGET=\"/test.txt\""); //Reads the data from
test file
Serial.flush();

x=0;
do{
  while(Serial.available()==0);
  data[x]=Serial.read();
  x++;
}while((data[x-1]!=','));

x=0;

while(Serial.peek()!=0x0A){
  file_size=(file_size*10)+(Serial.read()-0x30);
};

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

Serial.print(data);

while(1);
}

```

Uploading a file from local storage or SD to FTP server

```

/*
 * 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[256];
int x=0;

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

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();               // swithes the module ON

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

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop()
{

    Serial.println("AT+CFTPSSTART");           //Acquires FTPS protocol stack
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));

    Serial.print("AT+CFTPSLOGIN=\"");           //Logs into FTPS server
    Serial.print(server);
    Serial.print("\",");
    Serial.print(port);
    Serial.print("\",\");
}

```

```

Serial.print(user_name);
Serial.print("\",\");
Serial.print(password);
Serial.println("\");
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPMODE=1");    //Selects pasive mode
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPSTYPE=A");    //Selects ASCII mode
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

//Envia data
Serial.println("AT+CFTPSPUTFILE=\"/testfile.txt\",0"); //Uploads a test
file from FTPS server.
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    digitalWrite(13, LOW);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

while(1);
}

```

Downloading a file from FTP server or SD to local storage

```

/*
 * 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[256];
int x=0;

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

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+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

void loop()
{

    Serial.println("AT+CFTPSSTART"); //Acquires FTPS protocol stack
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='K'&&data[x-2]=='O'));
}

```

```

Serial.print("AT+CFTPSLOGIN=\""); //Logs into FTPS server
Serial.print(server);
Serial.print("\",");
Serial.print(port);
Serial.print(",\");
Serial.print(user_name);
Serial.print("\",\");
Serial.print(password);
Serial.println("\");
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPMODE=1"); //Selects pasive mode
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CFTPSTYPE=A"); //Selects ASCII mode
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

//Envia data
Serial.println("AT+CFTPSGETFILE=\"/testfile.txt\",0"); //Downloads a
test file from FTPS server.
Serial.flush();
x=0;
do{
    while(Serial.available()==0);
    digitalWrite(13, LOW);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

while(1);
}

```

16.3. Command summary

AT command	Response	Description
AT+CGSOCKCONT=**,&&&	OK	** is the protocol and &&& Access Point Name.
AT+CFTPSERV="*****"		Sets FTP server domain name or IP address. ***** is the domain name or the IP.
AT+CFTPPORT=***	OK	Sets FTP server port. *** is the port.
AT+CFTPUN="*****"	OK	Sets user name for FTP server access. *** is the user name.
AT+CFTPPW="*****"	OK	Sets password for FTP server access. *** is the password.
AT+CFTPMODE=	OK	Sets FTP mode. 1 for passive or 0 for proactive.
AT+CFTPTYPE=	OK	Sets the transfer type on FTP server . A for ASCII or I for binary.
AT+CFTPPUT="*****",&&		Puts a file from FTP server to serial port . *** is the path with the name of the file and && is the length.
AT+CFTPGET="*****"		Gets a file from FTP server to serial port . *** is the path with the name of the file.
AT+CFTPPUTFILE="*****",&		Uploads a file to FTP server from module . *** is the path and & is the storage directory.
AT+CFTPGETFILE="*****",&		Downloads a file from FTP server to module . *** is the path with the name and & is the storage directory.
AT+CFTPSSTART	OK	Acquires FTPS protocol stack.
AT+CFTPSLOGIN="***",&&&,"xxx","yyy"	OK	Logs in FTPS server. *** is the host address, &&& is the port, xxx is the user name and yyy is the password.
AT+CFTPSSTYPE=	OK	Sets the transfer type on FTPS server . A for ASCII or I for binary.
AT+CFTPSPUT="*****",&&		Puts a file from FTPS server to serial port . *** is the path with the name of the file and && is the length.
AT+CFTPSGET="*****"		Gets a file from FTPS server to serial port . *** is the path with the name of the file.
AT+CFTPSPUTFILE="*****",&		Uploads a file to FTPS server from module . *** is the path and & is the storage directory.
AT+CFTPSGETFILE="*****",&		Downloads a file from FTPS server to module . *** is the path with the name and & is the storage directory.

17. TCP and UDP

17.1. UDP client

Sending data to a UDP server:

```
/*
 * 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< 5;i++){
        delay(5000);
    }

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
```

```

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

void loop()
{
    Serial.print("AT+NETOPEN=\"UDP\","); //Opens the socket with the type
of protocol and the port
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='d'&&data[x-2]=='e')); //waits for response
    "Network opened"

    Serial.print("AT+UDPSEND=12,\""); //Sends UDP data
    Serial.print(server);
    Serial.print "\",");
    Serial.println(port);
    Serial.flush();
    while(Serial.read()!='>');

    Serial.println("Test message");

    Serial.println("AT+NETCLOSE"); //Opens the socket with the type of
protocol and the port
    Serial.flush();
    while(Serial.read()!='K');

    while(1);
}

```

17.2. TCP client

Sending data to a TCP server:

```

/*
 * 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< 5;i++){
        delay(5000);
    }

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    while(Serial.read()!='K');

}

void loop()
{

    Serial.print("AT+NETOPEN=\"TCP\",");
    //Opens the socket with the type of protocol and the port
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='d'&&data[x-2]=='e'));           //waits for response
    "Network opened"

    Serial.print("AT+TCPCONNECT=\""); //Connects with the server
    Serial.print(server);
    Serial.print("\",");
    Serial.println(port);
    Serial.flush();
    while(Serial.read()!='K');
}

```

```

Serial.println("AT+TCPWRITE=12");          //Sends TCP data
Serial.flush();
while(Serial.read()!='>');

Serial.println("Test message");

x=0;
do{
  while(Serial.available()==0);
  data[x]=Serial.read();
  x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+NETCLOSE"); //Opens the socket with the type of
protocol and the port
Serial.flush();
while(Serial.read()!='K');

while(1);
}

```

17.3. Multiclient

SIM5218 allows to use ten connections simultaneously. Here is the example code with a UDP and TCP connections.

```

/*
 * 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 UDP_server[ ]="";
char UDP_port[ ]="";
char TCP_server[ ]="";
char TCP_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< 5;i++){
        delay(5000);
    }

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    while(Serial.read()!='K');

}

void loop()
{

    Serial.println("AT+NETOPEN=,,1");           //Opens the socket in
multiclient mode
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='d'&&data[x-2]=='e'));           //waits for response
"Network opened"

    Serial.println("AT+CIPOPEN=0,\"UDP\", \"\");           //Connects with the UDP
server
    Serial.print(UDP_server);
    Serial.print("\",");
    Serial.println(UDP_port);
    Serial.flush();
    while(Serial.read()!='K');

    Serial.println("AT+CIPOPEN=1,\"TCP\", \"\");           //Connects with the TCP
server
    Serial.print(TCP_server);
    Serial.print("\",");
    Serial.println(TCP_port);
    Serial.flush();
    while(Serial.read()!='K');

    Serial.println("AT+CIPSEND=0,16");           //Sends data to UDP server
    Serial.flush();
    while(Serial.read()!='>');

    Serial.println("UDP test message");
}

```



```

x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CIPSEND=1,16");    //Sends data to TCP server
Serial.flush();
while(Serial.read()!='>');

Serial.println("TCP test message");

x=0;
do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
}while(!(data[x-1]=='K'&&data[x-2]=='O'));

Serial.println("AT+CIPCLOSE=0");    //Closes UDP connection
Serial.flush();
while(Serial.read()!='K');

Serial.println("AT+CIPCLOSE=1");    //Closes TCP connection
Serial.flush();
while(Serial.read()!='K');

Serial.println("AT+NETCLOSE");        //Opens the socket with the type
of protocol and the port
Serial.flush();
while(Serial.read()!='K');

while(1);
}

```

17.4. TCP server

SIM5218 allows to create a TCP server with a maximum of ten TCP clients. In this example the server send a message every 5 seconds to all clients.

```

/*
 * 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];

int clients=-2;
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< 5;i++){
        delay(5000);
    }

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    while(Serial.read()!='K');

    Serial.print("AT+NETOPEN=\"TCP\",");
    //Opens the socket with the type of protocol and the port
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='d'&&data[x-2]=='e'));          //waits for response
    "Network opened"

    Serial.println("AT+SERVERSTART"); //Starts TCP server
    Serial.flush();
    while(Serial.read()!='K');

}

void loop()
{

    Serial.println("AT+LISTCLIENT"); // List all of clients

```

```

Serial.flush();
clients=-2;
do{
  if(Serial.read()==0x0D){
    clients++;
  }
}while(Serial.peek()!='K'));

for(int y=0;y< clients;y++){
  Serial.print("AT+ACTCLIENT="); //activates the connection with the
client
  Serial.println(y);
  while(Serial.read()!='K'));

  Serial.println("AT+TCPWRITE=12"); //Sends TCP data
  Serial.flush();
  while(Serial.read()!='>');

  Serial.println("Test message");

  x=0;
  do{
    while(Serial.available()==0);
    data[x]=Serial.read();
    x++;
  }while(!(data[x-1]=='k'&&data[x-2]=='o')); //Waits for "send ok"

  Serial.print("AT+CLOSECLIENT="); //desactivates the connection with the
client
  Serial.println(y);
  while(Serial.read()!='K'));
}

delay(5000);
}

```

17.5. Command summary

AT command	Response	Description
AT+NETOPEN="***",&&		Opens a socket. *** is the socket type and && is the port.
AT+NETCLOSE	OK	Closes socket.
AT+UPDSEND		Sends UDP data.
AT+TCPCONNECT="***",&&		Establishes TCP connection with TCP server. *** is the IP address and && is the port.
AT+TCPWRITE		Sends TCP data when the TCP connection is established.
AT+SERVERSTART	OK	Starts up TCP server.
AT+LISTCLIENT		Lists all of clients' information.
AT+ACTCLIENT=*	OK	Activates the specified client. * is the number of the client.
AT+CLOSECLIENT=*	OK	Disconnects the specified client. * is the number of the client.
AT+CIPOPEN="*",&&,"*xx",yy		Establish a connection with TCP server or UDP server. * is the number of the connection, && is the type of transmission protocol, xx is the server IP and yy is the server port.
AT+CIPSEND		Sends some data to remote host in multi-client mode.
AT+CIPCLOSE=*	OK	Closes a specified connection in multi-client mode. * is the number of the connection.

18. HTTP AND HTTPS

18.1. http

SIM5218 can launch a HTTP operation like GET or POST. Here is an example with GET operation:

```
/*
 * 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)

int x = 0;

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< 5;i++){
        delay(5000);
    }

    Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\");
    Serial.flush();
    while(Serial.read()!='K');
}

void loop()
{

    Serial.print("AT+CHTTPACT=\"); //Connects with the HTTP server
    Serial.print(server);
    Serial.print("\",");
    Serial.println(port);
    Serial.flush();
    x=0;
    do{
        while(Serial.available()==0);
        data[x]=Serial.read();
        x++;
    }while(!(data[x-1]=='T'&&data[x-2]=='S'));           //waits for response
"REQUEST"

    Serial.println("GET /index.html HTTP/1.1");
    Serial.println("Host: www.mywebsite.com");
    Serial.println("Content-Length: 0");
    Serial.write(0x1A);           //sends ++
    Serial.write(0x0D);
    Serial.write(0x0A);

    while(Serial.read()!='K');
    while(Serial.read()!=' ');

    data_size=0;
}

```

```

do{
  data_size*=10;
  data_size=data_size+(Serial.read()-0x30);
}while(Serial.peek()!=0x0D);

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

Serial.print(data);

  while(1);
}

```

18.2. HTTPS

```

/*
 * 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)

int x = 0;

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< 5;i++){
    delay(5000);
}

Serial.println("AT+CGSOCKCONT=1,\"IP\", \"myapn\"");
Serial.flush();
while(Serial.read()!='K');
}

void loop()
{
    Serial.println("AT+CHTTPSSTART"); //Acquires HTTPS protocol stack
    Serial.flush();
    while(Serial.read()!='K');

    Serial.println("AT+CHTTPSOPSE"); //Opens a new HTTPS session
    Serial.flush();
    while(Serial.read()!='K');

    Serial.println("AT+CHTTPSEND=63"); //Sends HTTPS request
    Serial.flush();
    while(Serial.read()!='>');

    Serial.println("GET /index.html HTTP/1.1");
    Serial.println("Host: www.mywebsite.com");
    Serial.println("Content-Length: 0");
    while(Serial.read()!='K');

    Serial.println("AT+CHTTPSRECV=1"); //Receives HTTPS response
    Serial.flush();
    while(Serial.read()!=',');

    data_size=0;
    do{
        data_size*=10;
        data_size=data_size+(Serial.read()-0x30);
    }while(Serial.peek() !=0x0D);

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

    Serial.println("AT+CHTTPSCLSE"); //Closes a new HTTPS session
    Serial.flush();
    while(Serial.read()!='K');

    Serial.println("AT+CHTTPSSTOP"); //Releases HTTPS protocol stack
    Serial.flush();
    while(Serial.read()!='K');

    Serial.print(data);

    while(1);
}

```

18.3. Command summary

AT command	Response	Description
AT+CHTTPACT="***",&&		Launches a HTTP operation . *** is the server address and && is the port.
AT+CHTTPSSTART	OK	Acquires HTTPS protocol stack.
AT+CHTTPSSTOP	OK	Releases HTTPS protocol stack.
AT+CHTTPSOPSE	OK	Opens a new HTTPS session.
AT+CHTTPSCLSE	OK	Closes the opened HTTPS session.
AT+CHTTPSEND=*		Sends HTTPS request. * is the length of the data in the sending buffer.
AT+CHTTPSRECV=*		Receives HTTPS response after sending HTTPS request.