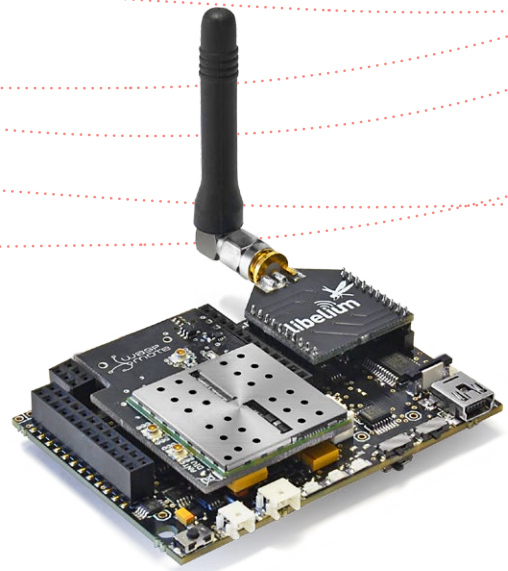
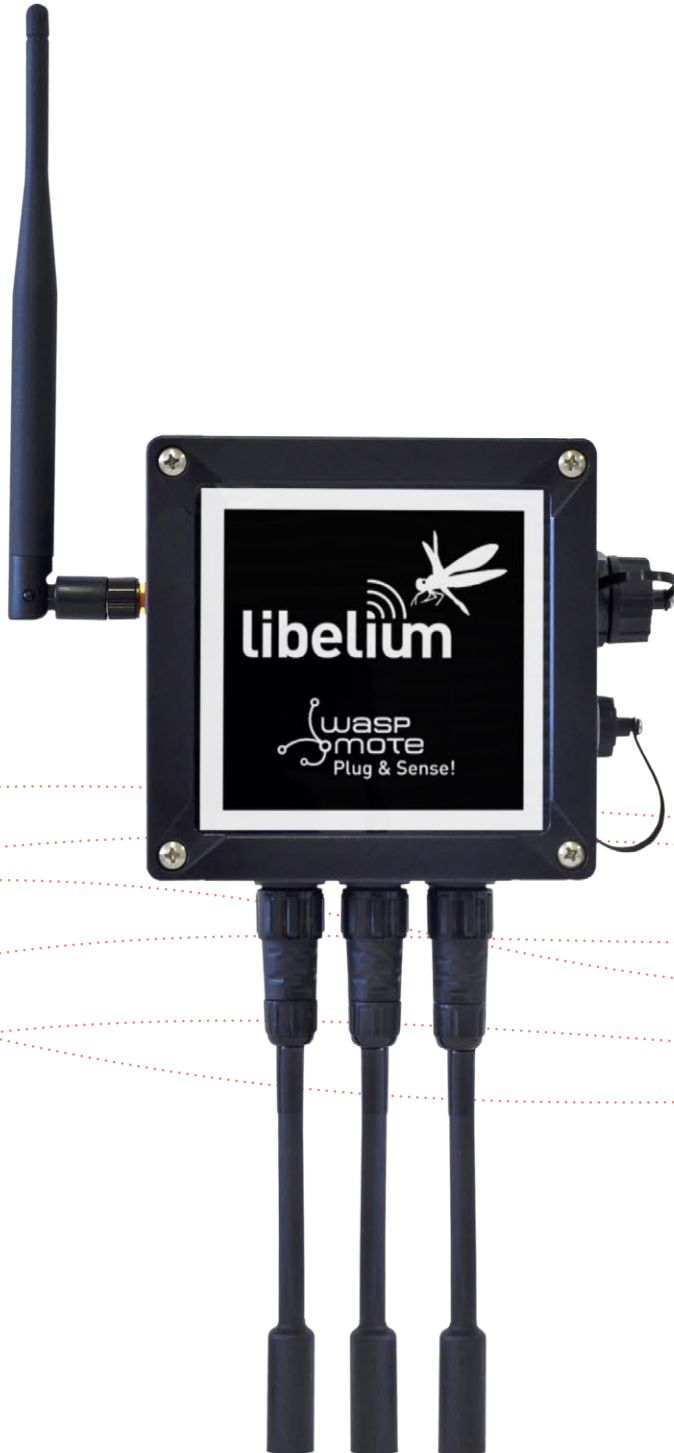
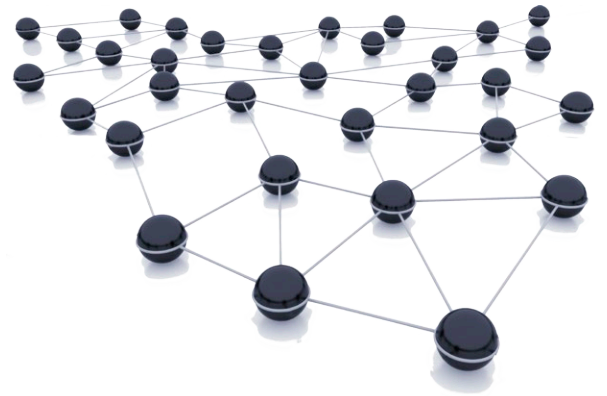


Over the Air Programming

Laying the groundwork



Document version: v7.0 - 02/2017

© Libelium Comunicaciones Distribuidas S.L.

INDEX

1. Introduction	3
1.1. Benefits	3
2. Concepts	4
2.1. OTA via FTP	6
3. OTA with 4G/3G/GPRS/WiFi via FTP	7
3.1. Procedure.....	7
3.2. Setting the FTP server configuration	8
3.2.1. Extern FTP server setup	8
3.2.2. Meshlium FTP server setup.....	9
3.3. Wasp mote basic sketch	11
3.4. OTA via GPRS and GPRS+GPS module	12
3.5. OTA via 3G module	15
3.6. OTA via WiFi module	16
4. Certifications.....	18

1. Introduction

This guide explains the Over The Air Programming features and functions. There are some variations in this feature for our new product lines Waspote v15 and Plug & Sense! v15, released on October 2016.

Some functions may not be compatible with Waspote v12 or Plug & Sense! v12. Besides, old functions may no longer exist. If you are using previous versions of our products, please use the corresponding guides, available on our [Development website](#).

You can get more information about the generation change on the document "[New generation of Libelium product lines](#)".

Differences of this library compared to the previous version:

- Over the air programming is no longer available for any kind of XBee modules

The concept of Wireless Programming or commonly known as Programming Over the Air (OTA) has been used in the past years overall for the reprogramming of mobile devices such as cell phones. However, with the new concept of Internet of Things, where networks consist of hundreds or thousands of nodes, OTA is taken to a new direction. Libelium provides an OTA method based on FTP transmissions to be used with GPRS, 3G, 4G and WiFi modules.

Note that the concept of OTA may have some other names such as:

- Over the air -> OTA
- Over the air Programming -> OTAP
- Firmware over the air -> FOTA
- Programming Over the air-> POTA
- Over the air service provisioning -> OTASP
- Over the air provisioning -> OTAP
- Over the air parameter administration -> OTAPA
- Over the air upgrade -> OTAU
- Over the air update -> OTAUR
- Over the air Download -> OAD
- Over the air flashing -> OTAF
- Multihop Over the air programming (MOTAP)

Important: The OTA process has been defined to work with the Waspote platform. For more info go to:

<http://www.libelium.com/development/waspote>

1.1. Benefits

Libelium OTA Benefits:

- Enables the upgrade or change of firmware versions without physical access.
- Upgrades the new firmware by querying an FTP server which helps to keep battery life.
- Upload new firmware in few minutes.

2. Concepts

Libelium OTA has been designed to work exclusively over the Wasp mote platform.

Wasp mote is a sensor device oriented to Internet of Things. It works with different protocols such as: 802.15.4, ZigBee, WiFi, 4G/GPRS, etc.

It counts with sleep modes which allow to save battery when it is not transmitting. Besides, there are more than 90 sensors available and a complete open source IDE (API libraries + compiler) which make it really easy to start working with the platform.

Note: to run programs which make use of the hibernate mode, it is needed to set the hibernate switch off, so this feature cannot be used with OTA. However, sleep or deepsleep modes can be performed.



Figure : Wasp mote Sensorial Device

The Wasp mote IDE v05 generates the binary file to be used in OTA programmings. There is a specific option in the Wasp mote IDE that permits to generate the necessary binary file for OTA in: "Sketch" → "Export binary to OTA file". When this option is chosen, the program is compiled and two files are generated inside /sketchbook/OTA-FILES folder: a hexadecimal and a binary file. The second one is valid for OTA via FTP with 4G/3G/GPRS/WiFi modules.

STEP 1: Click “Sketch” → “Export binary to OTA file” to generate the binary file:

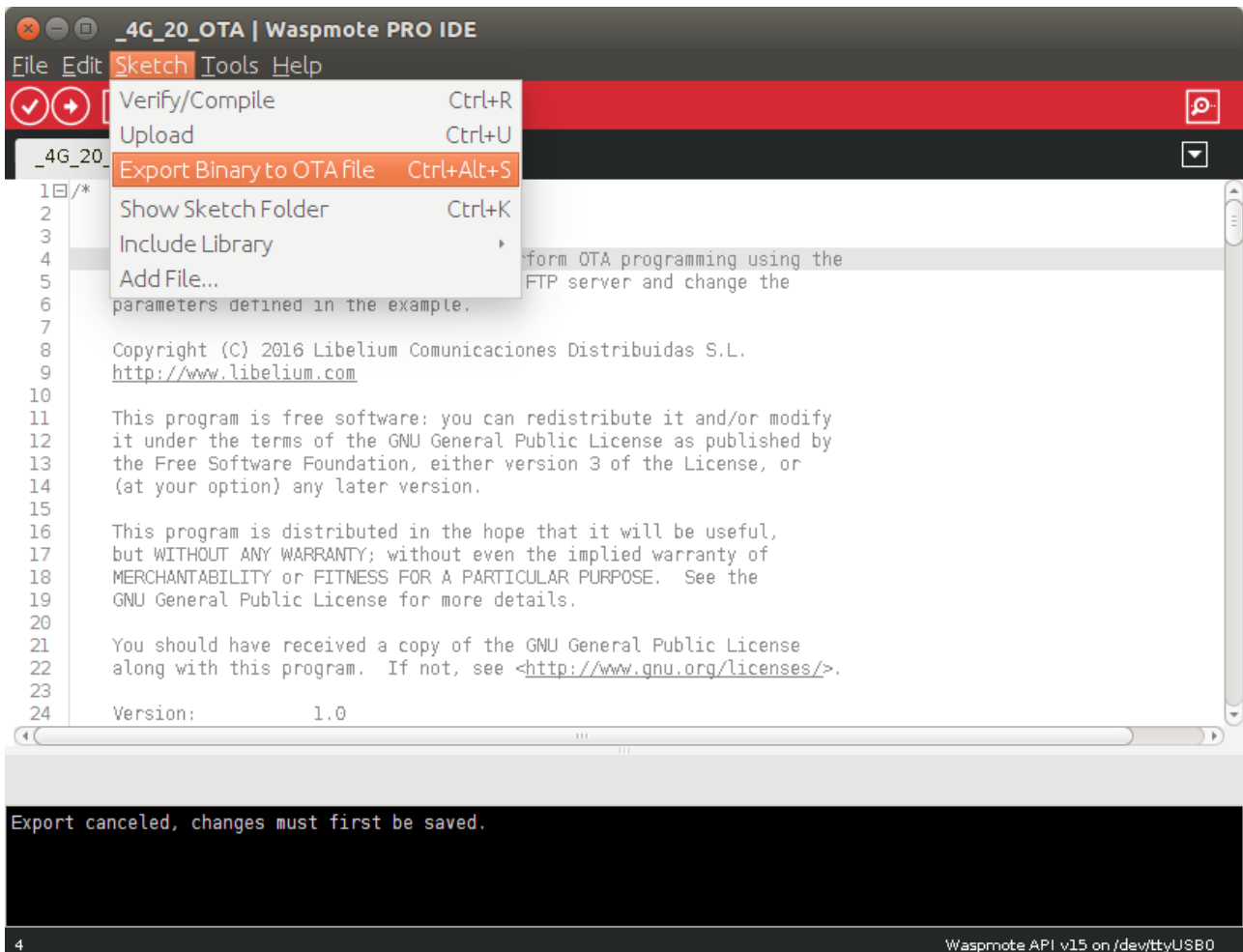


Figure: WaspMote IDE generates binary files

STEP 2: Search the binary file in the correct folder:

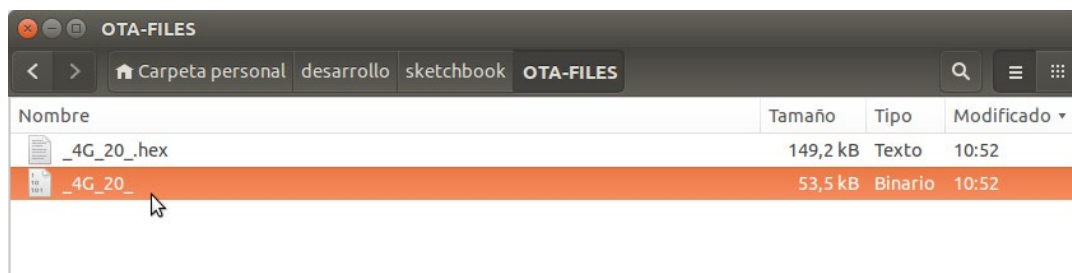


Figure: OTA-FILES folder

For instance, we are managing a program called `_4G_20_OTA.pde`. When the binary files are exported, two files are created inside the sketchbook sub-folder called **OTA-FILES**:

- `_4G_20_.hex` is the program compiled
- `_4G_20_` is the binary file needed for OTA via 4G (always defined by seven characters).

The WaspMote IDE gives these filenames truncating to seven characters the name of the WaspMote program.

Note: It is important to remark that the filename given by the WaspMote IDE to the binary files **CAN NOT be changed**. Otherwise, OTA will not work.

2.1. OTA via FTP

The reprogramming process is initiated by Waspote and it is supported by an FTP server.

Steps:

- Waspote queries the FTP server for a new program version
- Check if program name, path and version are correct
- Download the new program
- Reboot and start with the new program

Topologies:

- Protocols which support FTP transmissions are directly connected to the Network Access Point

Protocols supported:

- 4G modules:
 - LE910 EUG: 4G bands (800/1800/2600), 3G bands (850/900/2100), 2G bands (900/1800)
 - LE910 NAG: 4G bands (700/850/1700/1900), 3G bands (850/1900), 2G bands (850/1900)
 - LE910 AU: 4G bands (700/1800/2600)
- 3G - Tri-Band (2100/1900/900 MHz), Quad-Band (850/900/1800/1900 MHz)
- GPRS - 850/900/1800/1900 MHz
- WiFi - 2.4GHz (Worldwide)

Storage System:

Once the program is downloaded to Waspote it is stored in the 2GB SD card.

Meshlium OTA-FTP plug-in

Meshlium provides both FTP server and Manager System plug-in which permit to configure the server automatically by attaching the program binary file to be used.

3. OTA with 4G/3G/GPRS/WiFi via FTP

It is possible to update the Wasp mote's program using Over The Air Programming and the following modules: 4G, 3G, GPRS or WiFi module.

3.1. Procedure

The Wasp mote reprogramming is done using an FTP server and an FTP client which is Wasp mote itself. The FTP server can be configured by Meshlium. Otherwise, the user will have to setup an FTP server with the settings described in this guide.

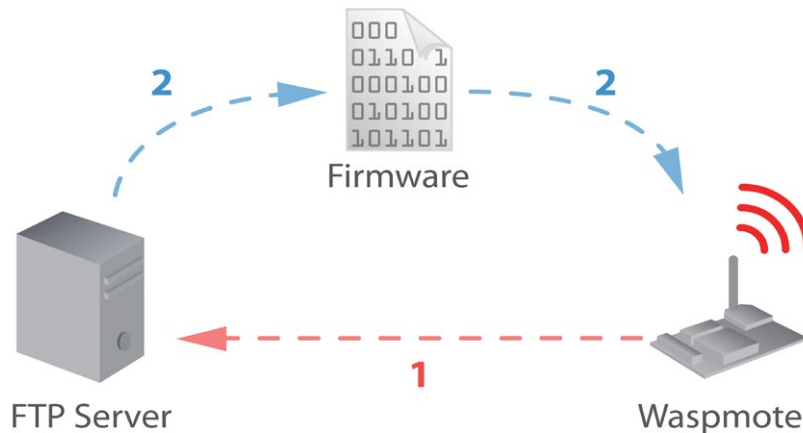


Figure : OTA via FTP protocol

There are two basic steps involved in OTA procedure:

- **Step 1:** Wasp mote requests a special text file which gives information about the program to update: program name, version, size, etc.
- **Step 2:** If the information given is correct, Wasp mote queries the FTP server for a new program binary file and it updates its flash memory in order to run the new program.

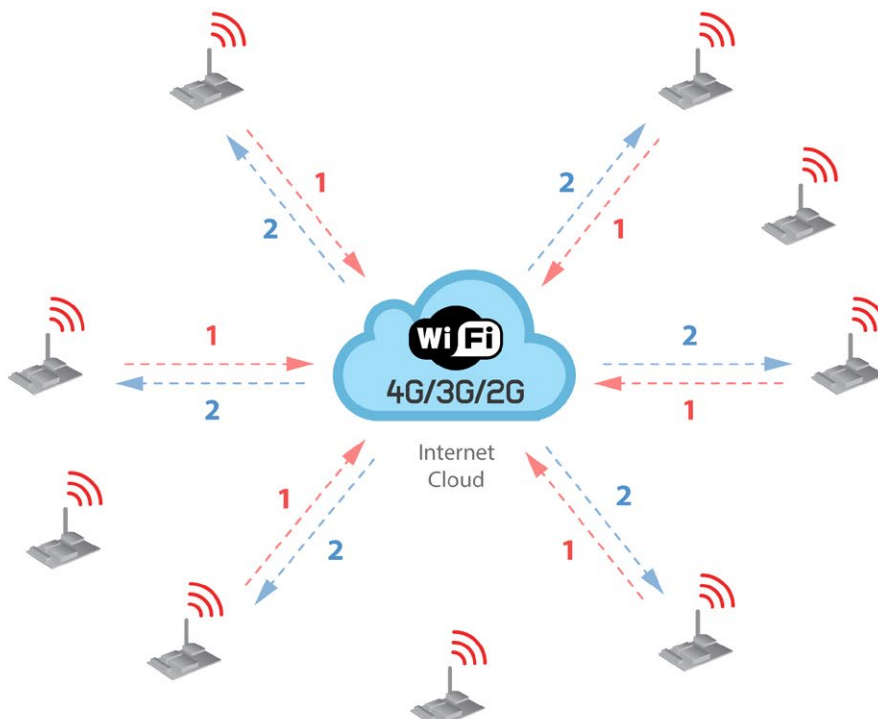


Figure : OTA steps via FTP protocol

3.2. Setting the FTP server configuration

The FTP server that Waspote connects to needs a specific configuration so as to OTA work properly. There are two ways to set up the FTP server:

- Extern user's FTP server
- Meshlium FTP server

3.2.1. Extern FTP server setup

There is a special text file called UPGRADE.TXT which defines several fields. This file must be located in the FTP Server root directory. Each field is defined by a specific label as shown below. Each line of the text file must end with the corresponding line end character: `'\n'`.

There are labels defined for each parameter:

- **FILE:** identifies the name of the binary to be downloaded (it is a 7-character name)
- **PATH:** identifies the path inside the FTP server where the file is found.
- **SIZE:** identifies the size of the binary file. It must be written as the number of bytes.
- **VERSION:** identifies the program version. It must be defined as a 1-unsigned-byte number (range: from 0 to 255). This label must be used by the user so as to validate new program versions when the file name specified in FILE does not change. Thus, Waspote will know if the same program has changed its version and if it is necessary to download it or not.

There is a restriction regarding the filename: it must be defined by 7 characters given by the Waspote IDE when it generates the binary file. It is important not to change the file name. If no OTA file needs to be downloaded, then the FILE field must be filled with `NO_FILE` pattern. Thus, Waspote will know that no downloading process is required. If no UPGRADE.TXT file is found in the FTP server, the OTA process will be called off as well.

Example 1: No binary file has to be downloaded

```
FILE:NO_FILE
PATH:
SIZE:
VERSION:
```

Example 2: Filename and Path specified in the text file. Notice that no `'/'` character appears at the end of the path.

```
FILE:PROG001
PATH:/demo/test
SIZE:46999
VERSION:10
```

Example 3: Filename and FTP Server's Root Directory are specified in the text file. Notice that root directory is specified as a unique `'/'` character.

```
FILE:PROG001
PATH:/
SIZE:56999
VERSION:11
```

During Waspote program execution, several cases can occur when the mote requests OTA via 4G/3G/GPRS/WiFi module:

- If there is no UPGRADE.TXT file or it is not found, Waspote will not perform any OTA process and will carry on with the current program.
- If there is an UPGRADE.TXT file and `NO_FILE` is specified as filename, then Waspote will carry on with the current program.
- If every label is correctly defined, Waspote will attempt to download and update the new program automatically.

3.2.2. Meshlium FTP server setup

The other possibility is to setup the FTP server using Meshlium.

Meshlium works as the Gateway of the Waspote Sensor Networks. It reads the sensor frames coming from the nodes and store them in its **internal data base** and in **external cloud systems** located on the Internet.

The Manager System is a web interface which comes with Meshlium. It allows you to control all the interfaces and system options in a secure, easy and quick way. For more information:

www.libelium.com/development/meshlium/documentation/meshlium-technical-guide/

There is a plug-in for Meshlium's Manager System which permits to set up the FTP server automatically. Thus, it is easier to prepare the binary files to be downloaded by Waspote. This plug-in is already installed in Manager System version 3.0.7 or more. Please check the Technical Guide in order to update the Manager System to its last version.

OTA via GPRS, 3G or WiFi is supported by Meshlium Xtreme. If you have a previous version, you need to upgrade it; please contact the Sales Department for more information:

<http://www.libelium.com/company/contact>

Besides, a default user is configured in **Meshlium FTP Server** with the following settings:

- user: ota
- password: libelium

This user directly connects to the following path in Meshlium's system directory where the application creates all the binary and UPGRADE.TXT files:

```
/mnt/user/ota
```

The OTA-FTP application can be found inside the Manager System -> Sensor Networks -> OTA-FTP

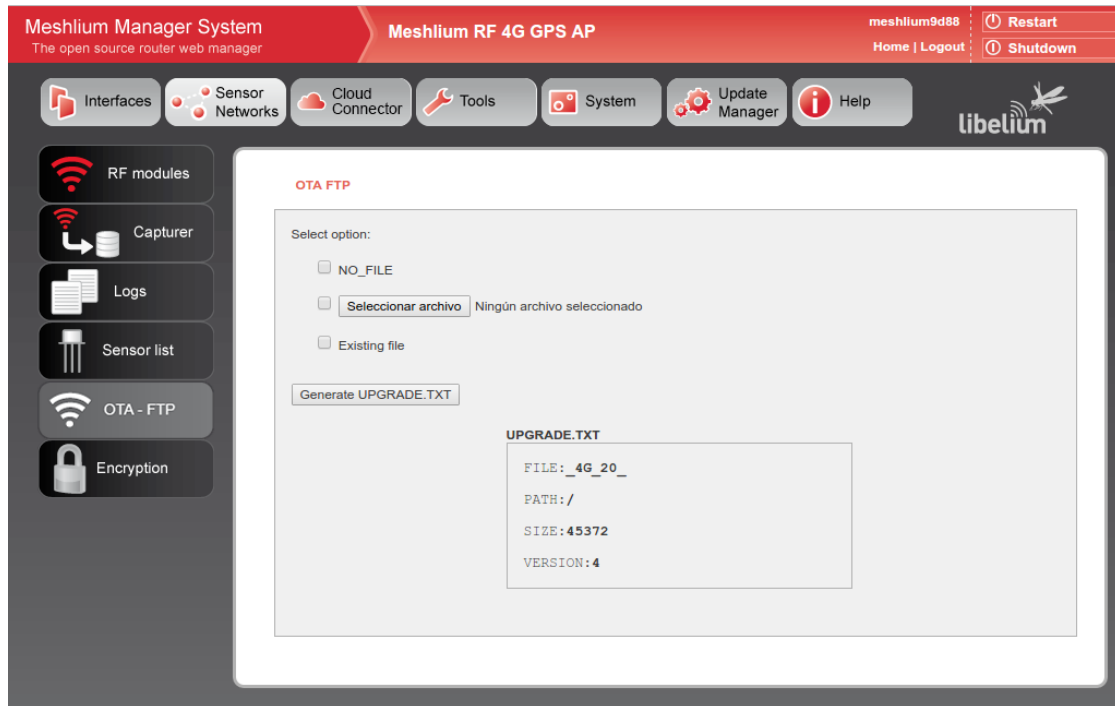


Figure : OTA-FTP Meshlium plugin

Firstly, there are three possibilities to be chosen:

- Select NO_FILE to inform Waspote that no OTA is necessary
- Select a new file generated by Waspote-IDE so as to update the Waspote's program.
- Select a existing binary if the user needs to update to an older program. The files are stored in the following path: /mnt/user/ota

Secondly, the program version is always set by the user before generating the new UPGRADE.TXT file. There is a specific input to indicate the program version. It must be defined as a 1-unsigned-byte number (range: from 0 to 255).

Finally, there is a button to generate the new UPGRADE.TXT file.

Once these steps have been followed, the binary file and the proper UPGRADE.TXT file are ready for the Waspote devices deployed which try to perform OTA via FTP transmission. This file is shown in the last window of the application representing the actual binary prepared for OTA.

3.3. Wasp mote basic sketch

Every Wasp mote sketch which provides OTA via FTP must have the following appearance:

```
// include and variable definitions

// setup function
void setup()
{
    // 1. Check if the program has been programmed successfully

    // 2. User code
    /*
     * Put your setup code here, to run once
     */
}

void loop()
{
    // 3. User code
    /*
     * Put your main code here, to run repeatedly
     */

    // 4. OTA request function
}
```

Step by step:

Step 1: The first checking is done by calling a specific function defined in Utilities library. This function must be called at the beginning of the code. This function returns three possible values. This allows the user to perform an action when a reprogramming has taken place.

```
{
    Utils.checkNewProgram();
}
```

Returns

- 0: Reprogramming error
- 1: Reprogramming ok
- 2: Normal starting (no OTA process)

Step 2: The user must fill this point with the code which is executed just once

Step 3: The user must fill this point with the code which is executed every loop repeatedly.

Step 4: The OTA request function depends on the module used. This function should be called when the user considers it so as to keep battery, i.e. once a day or once a week.

3.4. OTA via GPRS and GPRS+GPS module

When using a GPRS module, Wasp mote connects to the nearest Mobile Cell and follows the OTA procedure.

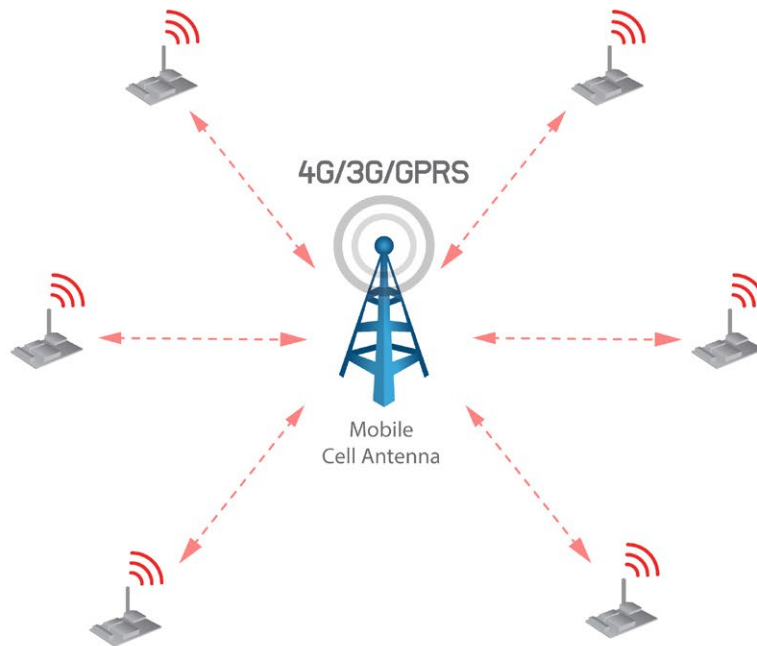


Figure : OTA with GPRS module

The GPRS_Pro and GPRS+GPS libraries implement the request function indicating four input parameters: FTP address, port, user and password. So a function example for GPRS_Pro is:

```
{
  GPRS_Pro.requestOTA("ftp_address","ftp_port","ftp_user","ftp_password");
}
```

This function returns a int8_t variable which indicates the error found:

- 1: if success
- 2' if error
- '-4' if error setting the type of Internet connection,
- '-5' if error setting the apn
- 6' if error setting the user name
- '-7' if error setting the password
- '-8' if error saving the configuration
- '-9' if error opening connection with the GPRS provider
- '-10' error downloading OTA version file
- '-11' if error getting the IP address
- 12: if error setting the FTP/HTTP ID (-50 if CME error available)
- 13: if error setting the FTP mode (-51 if CME error available)

- 14: if error setting the FTP type (-52 if CME error available)
- 15: if error setting the FTP server (-53 if CME error available)
- 16: if error setting the FTP port (-54 if CME error available)
- 17: if error setting the user name (-55 if CME error available)
- 18: if error setting the password (-56 if CME error available)
- 21: if error setting the file name in the FTP server (-57 if CME error available)
- 22: if error setting the path of the file in the FTP server (-58 if CME error available)
- 23: if error opening the FTP session (-59 if CME error available)
- 24: if error starting the SD
- 25: if error creating the file
- 26: error requesting data to the FTP (-60 if CME error available)
- 27: if error saving data into the SD
- 28: if error requesting more data to the FTP (-61 if CME error available)
- 30: setting the file name in the FTP to get the file size (-62 if CME error available)
- 31: setting the path in the FTP to get the file size (-63 if CME error available)
- 32: if error getting the file size (-64 if CME error available)
- 65: if FTP is busy
- 66: if there isn't FILE tag
- 67: if there isn't PATH tag
- 68: if there isn't VERSION tag
- 69: if OTA is not necessary
- 70: if OTA files are the same program version
- 71: if error opening connection with the GPRS provider
- 72: error downloading OTA file
- 73: if error getting the IP address
- 74: if error setting the FTP/HTTP ID (-112 if CME error available)
- 75: if error setting the FTP mode (-113 if CME error available)
- 76: if error setting the FTP type (-114 if CME error available)
- 77: if error setting the FTP server (-115 if CME error available)
- 78: if error setting the FTP port (-116 if CME error available)
- 79: if error setting the user name (-117 if CME error available)
- 80: if error setting the password (-118 if CME error available)
- 83: if error setting the file name in the FTP server (-119 if CME error available)

- 84: if error setting the path of the file in the FTP server (-120 if CME error available)
- 85: if error opening the FTP session (-121 if CME error available)
- 86: if error starting the SD
- 87: if error creating the file
- 88: error requesting data to the FTP (-122 if CME error available)
- 89: if error saving data into the SD
- 90: if error requesting more data to the FTP (-123 if CME error available)
- 92: setting the file name in the FTP to get the file size (-124 if CME error available)
- 93: setting the path in the FTP to get the file size (-125 if CME error available)
- 94: if error getting the file size (-126 if CME error available)
- 127: if FTP is busy

Before calling the request function, it is necessary to switch on the GPRS module and connect to a GPRS network.

The following example shows how to perform OTA via GPRS_Pro module. Keep in mind that the user has to change the APN settings to their own (constants `AT_GPRS_APN`, `AT_GPRS_LOGIN` and `AT_GPRS_PASSW` into the file `""WaspGPRS_Pro_core.h""`) and must keep the `OTA_FUSE` with '1':

<http://www.libelium.com/development/waspmote/examples/gprs-24-ota>

Note: It is absolutely necessary to use the proper firmware in the GPRS module so as to perform the OTA process. The correct firmware version is 1137B01SIM900M64_ST_ENHANCE. You can check the firmware version executing the following example:

<http://www.libelium.com/development/waspmote/examples/gprs-22-getting-imsi-from-sim-and-imei/>

Note: It is absolutely necessary to use the proper firmware in the GPRS+GPS module so as to perform the OTA process. The correct firmware version is 1137B03SIM908M64_ST_ENHANCE . You can check the firmware version executing the following example:

<http://www.libelium.com/development/waspmote/examples/gprs-sim908-22-getting-imsi-from-sim-and-imei/>

Note: In the case you want to upgrade the firmware to a valid version, please ask Libelium [Sales Department](#) for more information.

3.5. OTA via 3G module

When using the 3G module, Wasp mote connects to the nearest Mobile Cell and follows the OTA procedure.

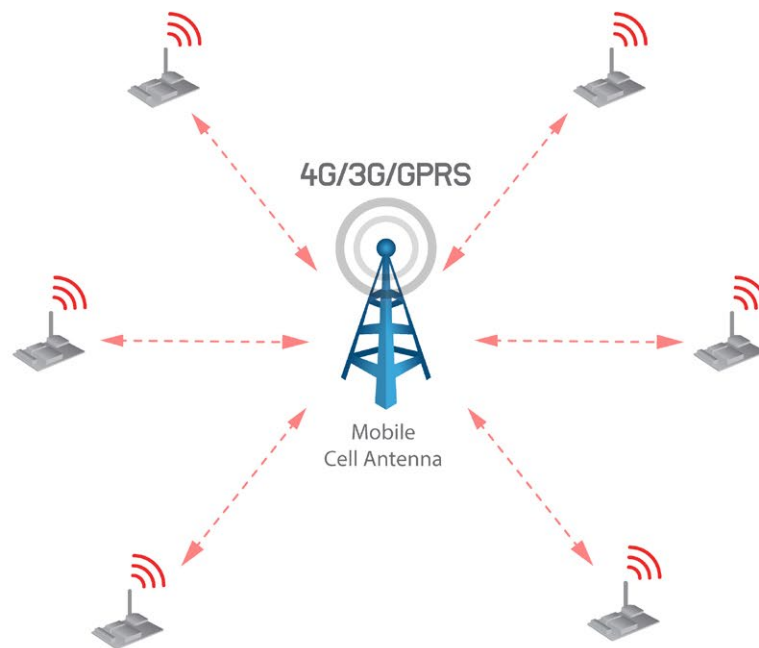


Figure : OTA with 3G module

The 3G library implements the request function indicating four input parameters: FTP address, port, user and password. So a function example is:

```
{
  _3G.requestOTA("ftp_address","ftp_port","ftp_user","ftp_password");
}
```

This function returns a `int8_t` variable which indicates the error found:

- 1: on success
- 2: if error setting the connection parameters (APN)
- 3: if error setting the FTP server ('-13' if CME error available)
- 4: if error setting the FTP port ('-14' if CME error available)
- 5: if error setting the FTP mode ('-15' if CME error available)
- 6: if error setting the FTP type ('-16' if CME error available)
- 7: if error setting the user name ('-17' if CME error available)
- 8: if error setting the FTP password ('-18' if CME error available)
- 19: if error downloading the OTA version file
- 21: if error with CME code (FTP error) downloading the OTA version file
- 22: if error sending the OTA version file from 3G module to Wasp mote's SD
- 23: if there isn't FILE tag
- 24: if there isn't PATH tag

- 25: if there isn't VERSION tag
- 26: if OTA is not necessary
- 27: if OTA files are the same program version
- 28: if error downloading the OTA file
- 31: if error with CME code (FTP error) downloading the OTA file
- 32: if error sending the OTA file from 3G module to Wasp mote's SD

Before calling the request function, it is necessary to switch on the 3G module and connect to a 3G or GPRS network.

The following example shows how to perform OTA via 3G module. Keep in mind that the user has to change the APN settings to their own (constants `_3G_APN`, `_3G_LOGIN` and `_3G_PASSW` into the file "Wasp3G.cpp") and must keep the `OTA_FUSE` with '1':

www.libelium.com/development/waspmote/examples/3g-32-ota

3.6. OTA via WiFi module

When using the WiFi module, Wasp mote connects to the Access Point set up by the user and follows the OTA procedure.

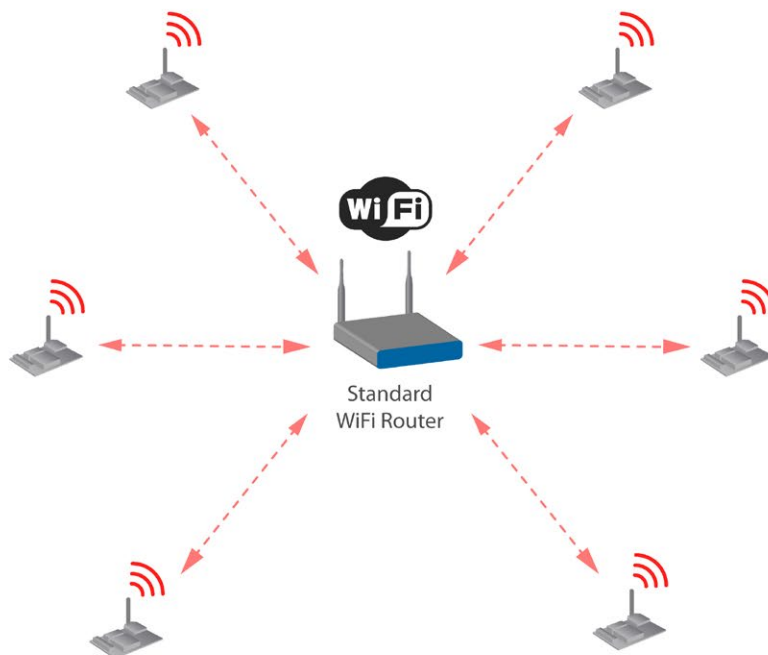


Figure : OTA with WiFi module

Note: Be aware of the decrease in success percentage depending on the devices the user sets up in a WiFi network working together.

The `requestOTA()` function permits to open the FTP session to the desired FTP server and request the new binary. This function needs four input parameters:

- **server:** IP address or URL of the FTP server
- **port:** port number (21 is default port number for FTP)
- **user:** user for opening the FTP session
- **password:** password needed for opening the FTP session

Example:

```
{  
  WIFI_PRO.requestOTA(server, port, user, password);  
}
```

This function returns '1' in case of error. There is an error code variable called `_errorCode`, which permits to know what kind of error happened. The possibilities are:

`ERROR_CODE_0010`: SD not present

`ERROR_CODE_0021`: error downloading UPGRADE.TXT

`ERROR_CODE_0022`: filename in UPGRADE.TXT is not a 7-byte name

`ERROR_CODE_0023`: no FILE label is found in UPGRADE.TXT

`ERROR_CODE_0024`: NO_FILE is defined as FILE in UPGRADE.TXT

`ERROR_CODE_0025`: no PATH label is found in UPGRADE.TXT

`ERROR_CODE_0026`: no SIZE label is found in UPGRADE.TXT

`ERROR_CODE_0027`: no VERSION label is found in UPGRADE.TXT

`ERROR_CODE_0028`: version indicated in UPGRADE.TXT is lower/equal to Waspote's version

`ERROR_CODE_0029`: file size does not match the indicated in UPGRADE.TXT

`ERROR_CODE_0030`: error downloading binary file

Before calling the request function, it is necessary to switch on the WiFi module and join a network.

The following example shows how to perform OTA via WiFi module. Keep in mind that the user has to change the AP settings to their own. Besides, the FTP server settings must be changed in the case an external FTP server is used. If not, the default Meshlium's parameters are set, but the Meshlium's network address must be specified as the real one:

www.libelium.com/development/waspmote/examples/wifi-pro-21-ota

4. Certifications

Libelium offers 2 types of IoT sensor platforms, Wasp mote OEM and Plug & Sense!:

- **Wasp mote OEM** is intended to be used for research purposes or as part of a major product so it needs final certification on the client side. More info at: www.libelium.com/products/wasp mote
- **Plug & Sense!** is the line ready to be used out-of-the-box. It includes market certifications. See below the specific list of regulations passed. More info at: www.libelium.com/products/plug-sense

Besides, Meshlium, our multiprotocol router for the IoT, is also certified with the certifications below. Get more info at:

www.libelium.com/products/meshlium

List of certifications for Plug & Sense! and Meshlium:

- CE (Europe)
- FCC (US)
- IC (Canada)
- ANATEL (Brazil)
- RCM (Australia)
- PTCRB (cellular certification for the US)
- AT&T (cellular certification for the US)



Figure : Certifications of the Plug & Sense! product line

You can find all the certification documents at:

www.libelium.com/certifications