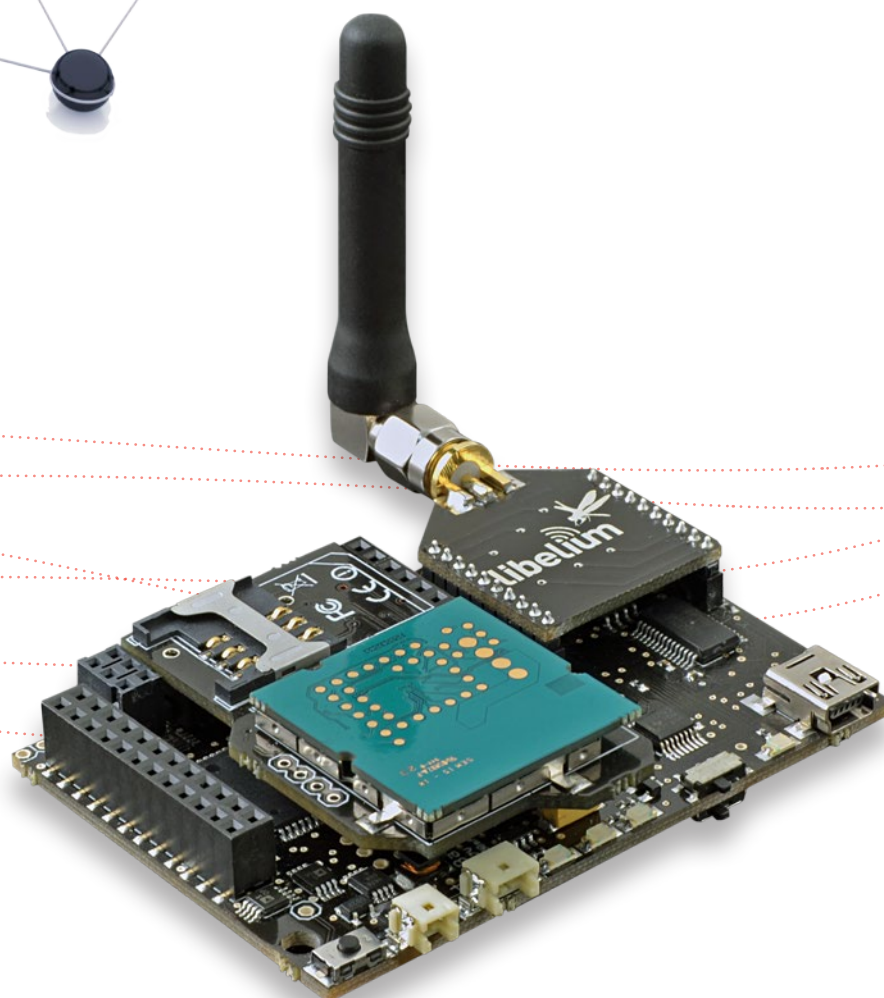


# Wasp mote IDE

## User Guide



Document Version: v4.1 - 01/2014  
© Libelium Comunicaciones Distribuidas S.L.

## INDEX

<b>1. Introduction .....</b>	<b>3</b>
1.1. New features .....	3
1.2. Other notes .....	3
<b>2. Installation .....</b>	<b>4</b>
2.1. Windows .....	4
2.2. Mac OS X .....	5
2.3. Linux .....	6
<b>3. Environment .....</b>	<b>8</b>
3.1. Writing Sketches .....	8
3.2. Menus .....	9
3.3. Uploading .....	9
3.4. Libraries .....	9
3.5. Serial Monitor .....	10
3.6. Preferences and Language Support .....	11
<b>4. Automatic updates .....</b>	<b>12</b>
4.1. IDE update .....	12
4.2. API update .....	13
<b>5. OTA compatibility .....</b>	<b>14</b>
<b>6. Documentation changelog .....</b>	<b>16</b>

# 1. Introduction

The aim of this Guide is to introduce the user to the new Integrated Development Environment for Waspmote (Waspmote IDE). This Integrated Development Environment (IDE) is used for writing the code and uploading it to Waspmote and Waspmote Plug & Sense!. It also used to monitor serial output and for debugging. This IDE contains the Waspmote API (the API is the set of all libraries Waspmote needs for compiling programs). New API versions are released instantly by Libelium whenever improvements are made or bugs fixed.

This manual is **only intended** for Waspmote IDE versions higher or equal to v03, and Waspmote API versions higher or equal to v003. Waspmote IDE versions higher or equal to v03 are valid **only for** Waspmote PRO (v1.2) and Waspmote Plug & Sense!.

## 1.1. New features

There are several benefits to use the new Waspmote IDE:

- Faster compilation
- New API structure
- More debug messages
- RAM memory used information
- Easier installation
- More preferences
- 30+ different languages
- New automatic updates
- OTA compatibility
- Scrollable editor tabs

We want to specially thank all developers that have given us their feedback to improve the Development Environment. We hope developers can take advantage of all these new features.

## 1.2. Other notes

The user must have one IDE for each API. According to the new libraries organization for Waspmote v12, there are core libraries which are unique for each IDE (they are always invoked). Besides, there is an extra folder with the optional libraries (not always invoked). So due to the unique core libraries, if the user wants to experiment with 'n' APIs, he needs to have 'n' IDEs installed in the computer. The new libraries organization reduces the RAM usage, and keeps everything tidy (core+optional).

As a portable program, you can have so many IDEs as you want. The location of the API for each OS is:

### Windows:

- Examples: <Waspmote IDE folder>\examples\
- Libraries: <Waspmote IDE folder>\libraries\
- API core: <Waspmote IDE folder>\hardware\waspmote\cores\waspmote-api\

### MacOS:

- Examples: Waspmote.app/Contents/Resources/Java/examples/
- Libraries: Waspmote.app/Contents/Resouces/Java/libraries/
- API core: Waspmote.app/Contents/Resources/Java/hardware/waspmote/cores/waspmote-api/

### Linux:

- Examples: <Waspmote IDE folder>/examples/
- Libraries: <Waspmote IDE folder>/libraries/
- API core: <Waspmote IDE folder>/hardware/waspmote/cores/waspmote-api/

## 2. Installation

This section explains how to install the Waspote v12/Pro Integrated Development Environment (Waspote IDE in short) in your Operating System.

First, download the Waspote v12 IDE from the Waspote software webpage:

[http://www.libelium.com/development/waspote/sdk\\_applications/](http://www.libelium.com/development/waspote/sdk_applications/)

and select your OS:



### » Waspote Pro IDE - v.XX

The Waspote Pro IDE is Waspote's software development kit. It is used for writing and uploading code to Waspote. It also can monitor the serial output and be used in debugging.

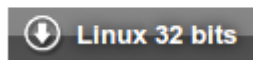


Figure 1: Waspote IDE download links

If you are using Linux and don't know if download the 32-bit or 64-bit version, you can open a terminal and type `uname -p` to find out.

### 2.1. Windows

#### Unzip

When the download finishes, unzip the downloaded file (*waspote-pro-ide-vXX-windows.zip*). Make sure to preserve the folder structure. Double-click the folder to open it. There should be a few files and sub-folders inside.

#### Prepare the computer

Then plug your Waspote v12 board into the computer with the USB cable. When you connect the board, Windows should start the driver installation process (if you haven't used the computer with Waspote before).

On **Windows Vista** and **Windows 7**, the driver will be automatically downloaded and installed.

On **Windows XP**, the Add New Hardware wizard will open:

- When asked Can Windows connect to Windows Update to search for software? Select No, not this time. Click next.
- Select Install from a list or specified location (Advanced) and click next.
- Make sure that Search for the best driver in these locations is checked; uncheck Search removable media; check Include this location in the search and browse to the *drivers/FTDI USB Drivers* directory of the Windows distribution. (The latest version of the drivers can be found on the FTDI website <http://www.ftdichip.com/Drivers/VCP.htm>) Click next.
- The wizard will search for the driver and then tell you that a "USB Serial Converter" was found. Click finish.
- The new hardware wizard will appear again. Go through the same steps and select the same options and location to search. This time, a "USB Serial Port" will be found.

To check if the drivers has been installed correctly, you should see in *Start* → *Devices and printers* a device called *FT232R USB UART*:

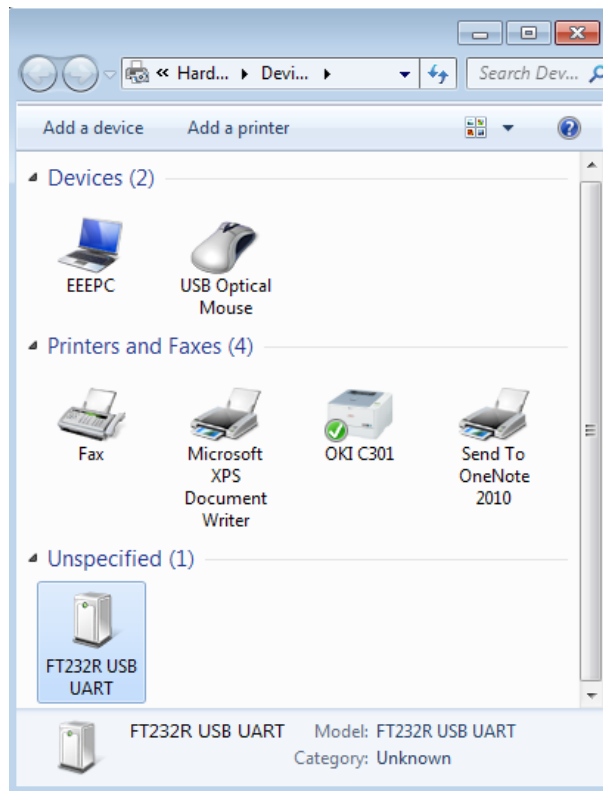


Figure 2: Waspnote drivers installed in Windows

### Launch Waspnote IDE

Double-click the IDE application.

Note: if the Waspnote software loads in the wrong language, you can change it in the preferences dialog. See the Environment section for details.

## 2.2. Mac OS X

### Unzip

When the download finishes, double-click on the downloaded file (*waspnote-pro-ide-vXX-macosx.zip*). This will expand the Waspnote application. Copy the Waspnote application into the Applications folder.

### Prepare computer

You need to download the latest version of the drivers from the FTDI website <http://www.ftdichip.com/Drivers/VCP.htm>. Once downloaded, double click the package, and follow the instructions in the installer. You'll need to restart your computer after installing the drivers.

Then plug in your Waspnote v12 board into the computer with the USB cable, and the Mac OS will detect it automatically.

## Launch Wasmote IDE

Double-click the IDE application.

Note: if the Wasmote software loads in the wrong language, you can change it in the preferences dialog. See the Environment section for details.

To check if the drivers have been correctly installed, you should see a device called `/dev/tty.usbserial-XXXXXX` in the IDE's Serial Port.

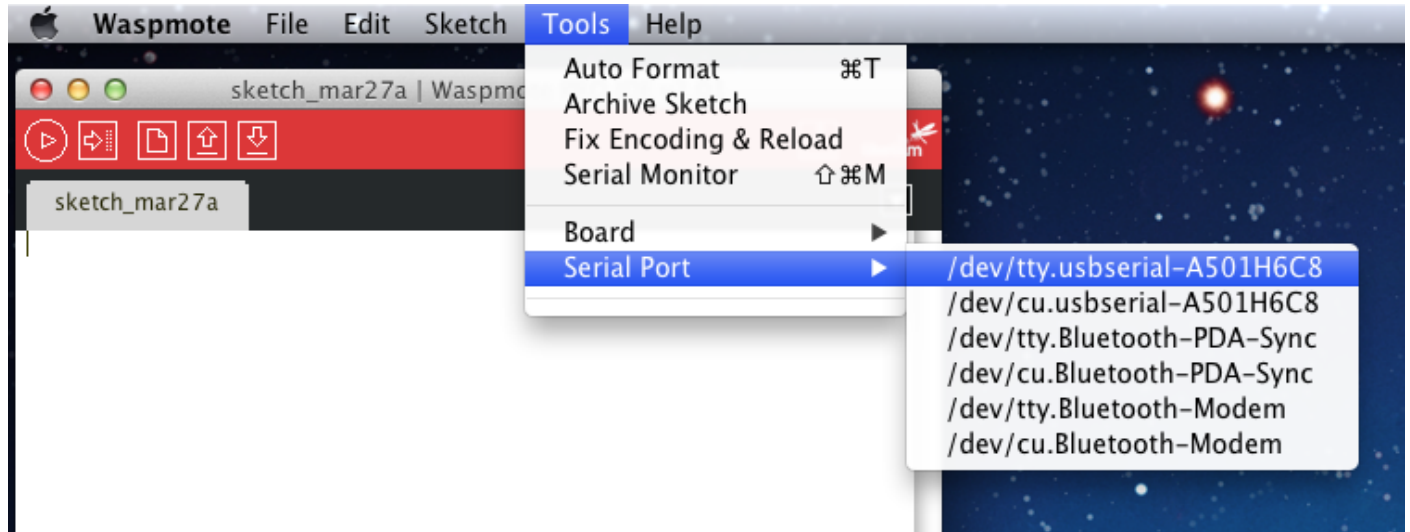


Figure 3: Wasmote correctly detected by Mac OS

## 2.3. Linux

### Unzip

When the download finishes, double-click on the downloaded file (`wasmote-pro-ide-vXX-linuxXX.tar.gz`). Make sure to preserve the folder structure. Double-click the folder to open it. There should be a few files and sub-folders inside.

### Prepare computer

You will need to install some programs to use the Wasmote IDE under Linux (the way you do this depends on your distribution):

- A Java Runtime Environment: `openjdk-7-jre`, `openjdk-6-jre`, sun's java 6 runtime or oracle JRE 7.
- Wasmote IDE has an internal pre-build gcc compiler, but If you have installed your own `avr-gcc` compiler, make sure to use 4.7.2 version.
- `librxtx-java` package.

Then plug your Wasmote v12 board into the computer with the USB cable, and the Linux OS should identify it like `/dev/ttyUSBX`.

## Launch Wasmote IDE

Copy the extracted files to a directory (or your desktop) and run the “wasmote” script.

\* For Ubuntu distributions higher or equal to 12.04 (with Unity UI) we provide an script to create a launcher App:

- Go to the unzipped folder
- Double-click on createAppUnity.sh
- Move the IDE application to your launcher.

**Note:** These tests were done using Ubuntu 10.x, 12.x and 13.04



Figure 4: Wasmote IDE on Unity

## Other notes:

- **Locking assertion failure from java when using Xinerama**

To fix:

```
sudo sed -i 's/XINERAMA/FAKEEXTN/g' /usr/lib/jvm/java-6-sun/jre/lib/amd64/motif21/libmawt.so
```

- **USB-Serial Devices not detected**

Wasmotes (and other USB-serial devices) end up as /dev/ttyUSBX.

- **Kernel Configuration**

If you configured your own kernel, the driver is either under

“Device Drivers → USB Support → USB Serial Converter Support → USB FTDI Single Port Serial Driver” or “Device Drivers → USB Support → USB Modem (CDC ACM) support”.

Select [Y], to have it “built in” to the kernel or [M] to have it built as a module.

Dmesg then should show the device like this:

```
ftdi_sio 2-2:1.0: FTDI USB Serial Device converter detected
```

or like this:

```
cdc_acm 5-2:1.0: ttyUSB0: USB ACM device
```

- **User Rights**

Make sure you are in the tty group. On some distributions, you might need to create an Udev rule for the FTDI USB Serial Device converter.

## 3. Environment

This section explains the main features of the Wasmote IDE.

### 3.1. Writing Sketches

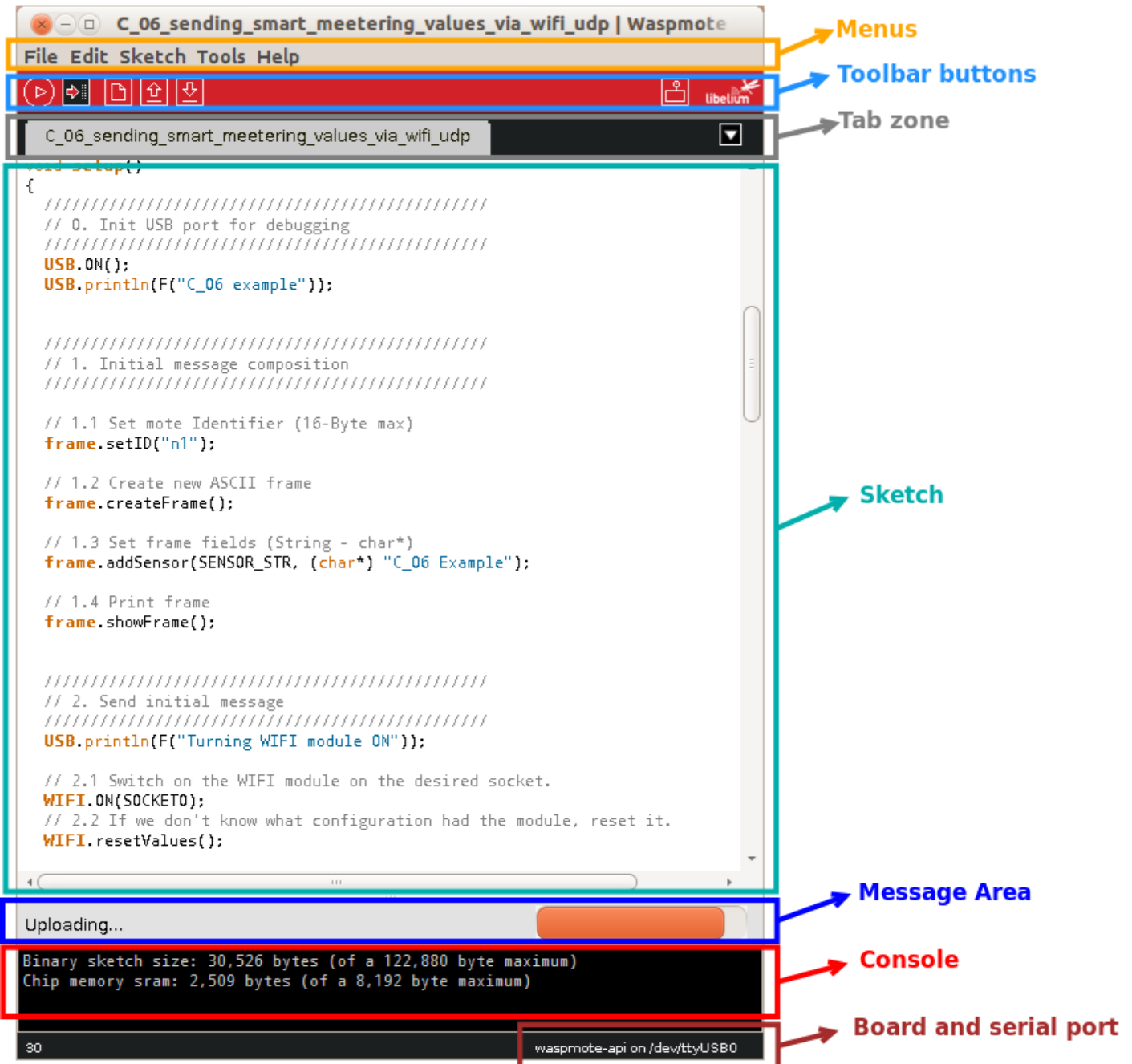


Figure 5: Wasmote IDE sections

Software written using the IDE are called **sketches**. These sketches are written in the text editor. Sketches are saved with the file extension ".pde".

The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the IDE including complete error messages and other information. The bottom righthand corner of the window displays the current board and serial port.

This IDE allows you to manage sketches with more than one file (in the tab zone). These can be normal Wasmote code files (.pde), C files (.c extension), C++ files (.cpp), or header files (.h).



## 3.2. Menus

The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.



• *Compile*: Checks your code for errors.



• *Upload*: Compile your code and uploads it to the Wasmote board.



• *New*: Creates a new sketch.



• *Open*: Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window.



• *Save*: Saves your sketch.



• *Serial Monitor*: Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive which means only those items relevant to the work currently being carried out are available.

## 3.3. Uploading

Before uploading your sketch, you need to select the correct items from the *Tools* → *Target and Tools* → *Serial Port* menus.

On Windows, the serial port is something like COMX; to find out, you look for USB serial device in the ports section of the Windows Device Manager. On a Mac, the serial port is something similar to `/dev/tty.usbserialXXX`. And on Linux, it should be `/dev/ttyUSBX`.

Once you have selected the correct serial port and target, press the upload button in the toolbar or select the Upload item from the File menu. Wasmote will reset automatically and begin the upload.

The IDE will display a loading bar and a message when the upload is complete, or show an error.

## 3.4. Libraries

Libraries provide extra functionality for use in sketches. To use a library in a sketch, select it from the *Sketch* → *Import Library* menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. There is a list of libraries in the reference. Some libraries are included with the Wasmote API. Others can be downloaded from a variety of sources.

**To install a third-party library:** Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a `.cpp` file, a `.h` file and often a `keywords.txt` file, examples folder, and other files required by the library.

To install the library, first quit the Wasmote IDE. Then uncompress the ZIP file containing the library. It should contain a folder called `MyLibrary`, with files like `MyLibrary.cpp` and `MyLibrary.h` inside. If the `.cpp` and `.h` files aren't in a folder, you'll need to create one. In this case, you'd make a folder called "MyLibrary". Drag the `MyLibrary` folder into the IDE libraries folder. There may be more files than just the `.cpp` and `.h` files, just make sure they're all there. Restart the IDE. Make sure the new library appears in the *Sketch* → *Import Library* menu item.

**Note:** If you are not advanced user, we recommend to use only Libelium's official libraries. Other libraries can cause damages to the electronics, and their use is obviously out of the warranty scope.

## 3.5. Serial Monitor

The Serial Monitor displays serial data being sent from the Waspote board (via USB or serial board).

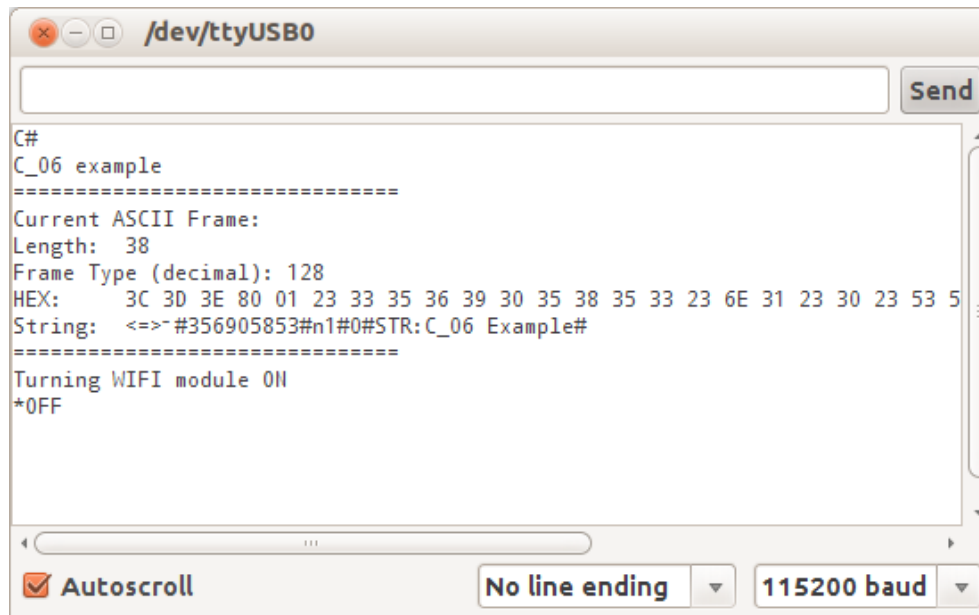


Figure 6: Serial Monitor

Choose the baud rate from the drop-down that matches the rate of Waspote v12 (USB = 115200 bps).

To send data to the board, enter text and click on the “send” button or press enter.

## 3.6. Preferences and Language Support

Some preferences can be set in the preferences dialog (found under the *Wasmote* menu on the Mac, or *File* on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

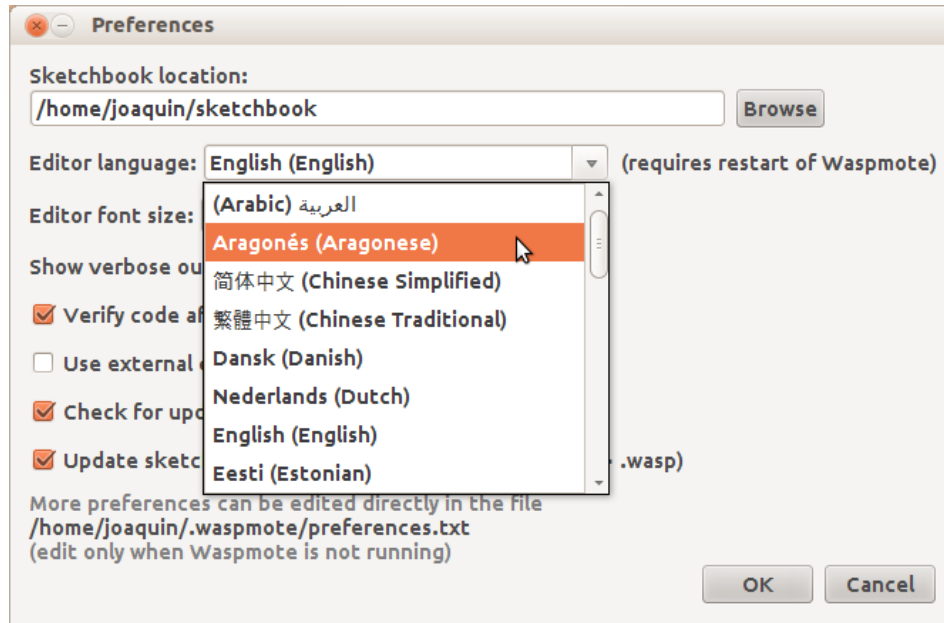


Figure 7: Preferences and Language Support

The IDE has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system.

If you would like to change the language manually, start the IDE and open the Preferences window. Next to *Editor Language* there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the IDE to use the selected language. If your preferred language is not supported, the IDE will default to English.

## 4. Automatic updates

The new Wasp mote IDE has a complete system of updates both for both the Wasp mote API and the IDE.

### 4.1. IDE update

When Libelium team releases a new version of the IDE for Wasp mote v12/Pro, the next window will appear:

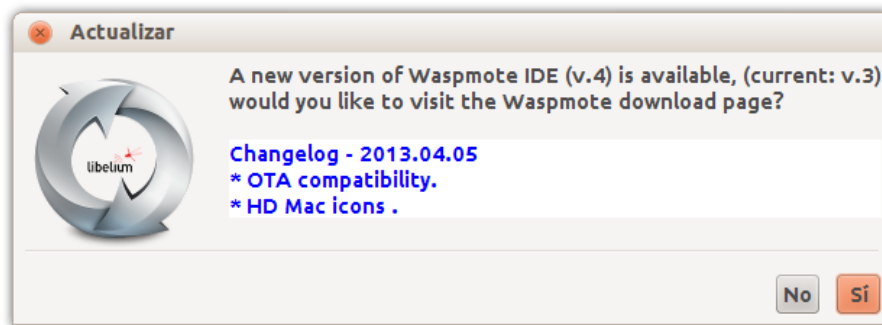


Figure 8: Wasp mote IDE update

On clicking Yes, the Libelium software downloads webpage is opened on your browser.

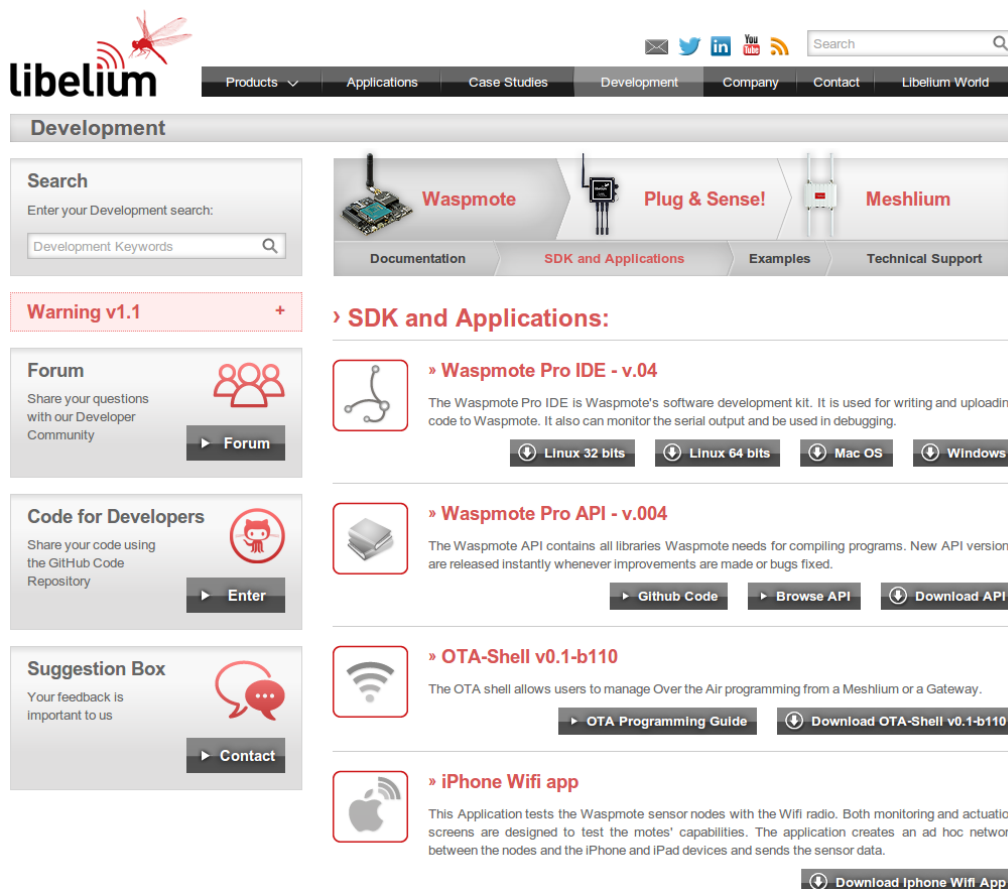


Figure 9: Libelium webpage

## 4.2. API update

When Libelium team releases an update of the API, the following window appears:

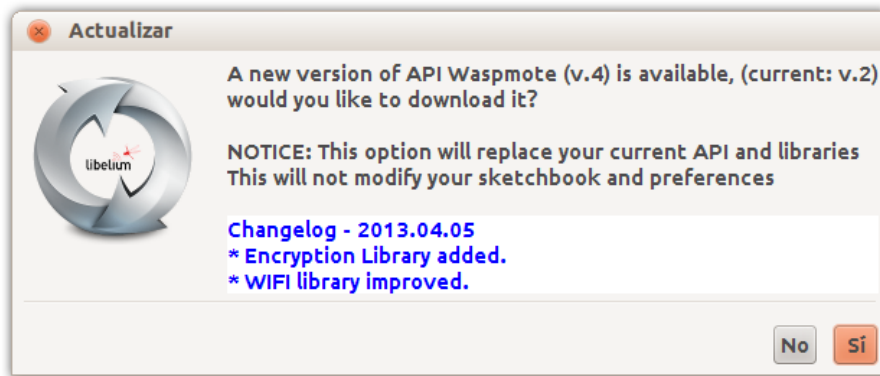


Figure 10: API update

On clicking Yes, next window appears while the API is updating.



Figure 11: API updating

This option will replace your current API, libraries and examples.

For security, the system makes a backup of your API, libraries and examples before updating them to the last version. The name of these back up files are "old\_api\_bckup.zip", "old\_examples\_bckup.zip", "old\_libraries\_bckup.zip".

Anyway, we recommend to make backups of your API and libraries time to time. This update will not make changes in your sketchbook and preferences.

## 5. OTA compatibility

The Wasmote IDE v03 or higher automatically generates the binary file to be used in OTA programmings. When a program is compiled or uploaded to Wasmote, two files are generated: a hexadecimal and a binary file. The first one is valid for OTA with XBee modules and the second one is valid for OTA via FTP with 3G/GPRS/WiFi modules.

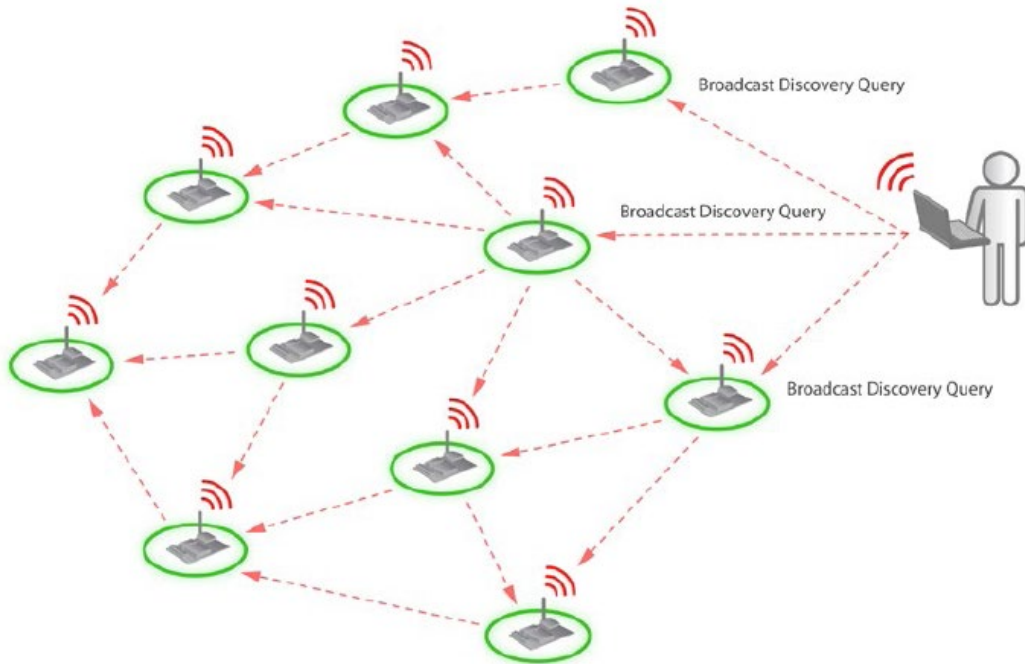


Figure 12: Over The Air programming

STEP 1: Click Verify button so as to compile the code

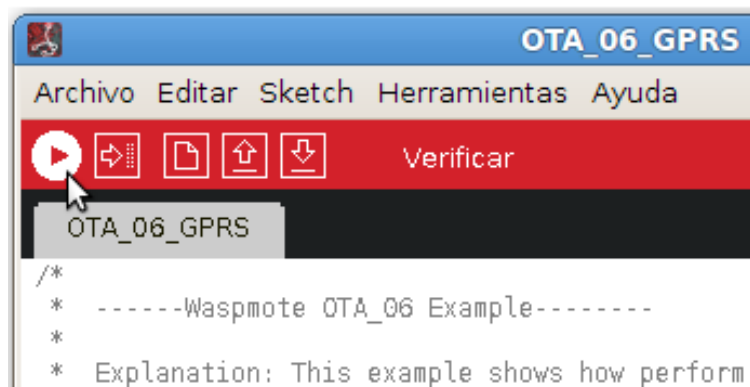


Figure 13: Wasmote IDE generates OTA files

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



Figure 14: OTA-FILES folder

For instance, we are managing a program called OTA\_06\_GPRS.pde. When the program is compiled, two files are created inside the sketchbook sub-folder called OTA-FILES:

- OTA\_06\_.hex is the hexadecimal file used for conventional OTA programming.
- OTA\_06\_ (seven characters) is the binary file needed for OTA via 3G or GPRS.

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

**Note:** It is important to remark that the filename given by the Waspote IDE to the binary files CAN NOT be changed. Otherwise, OTA will not work. More information about OTA programming in <http://www.libelium.com/ota/>

## 6. Documentation changelog

### From v4.0 to v4.1

Added references to the new compiler used from IDE v04. This new avr-gcc version has been proved to run larger binaries in a robust way, compared to the former IDE v03. Using an updated compiler version and libraries, several bugs will be avoided. The new features are:

- avr-gcc 4.7.2
- avr-libc 1.8.0