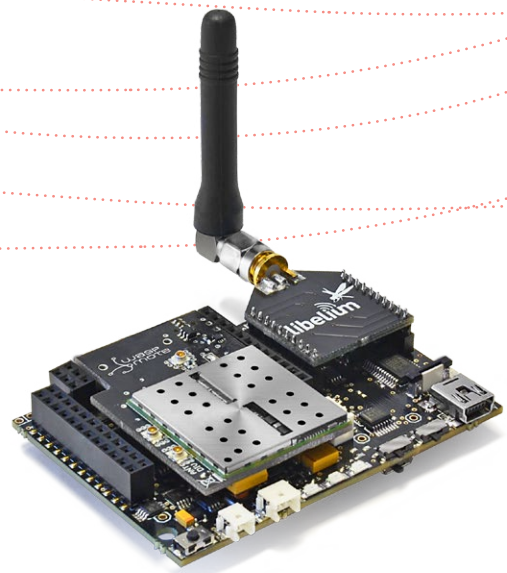
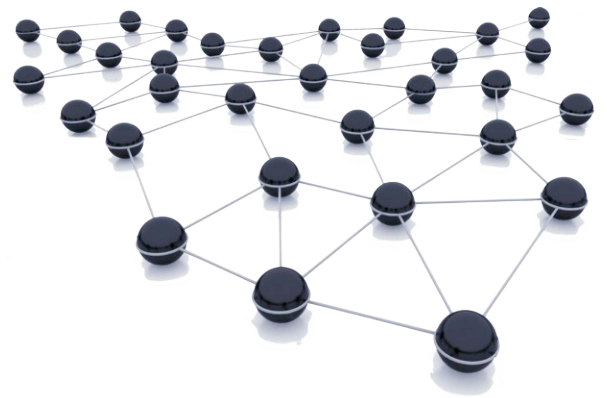
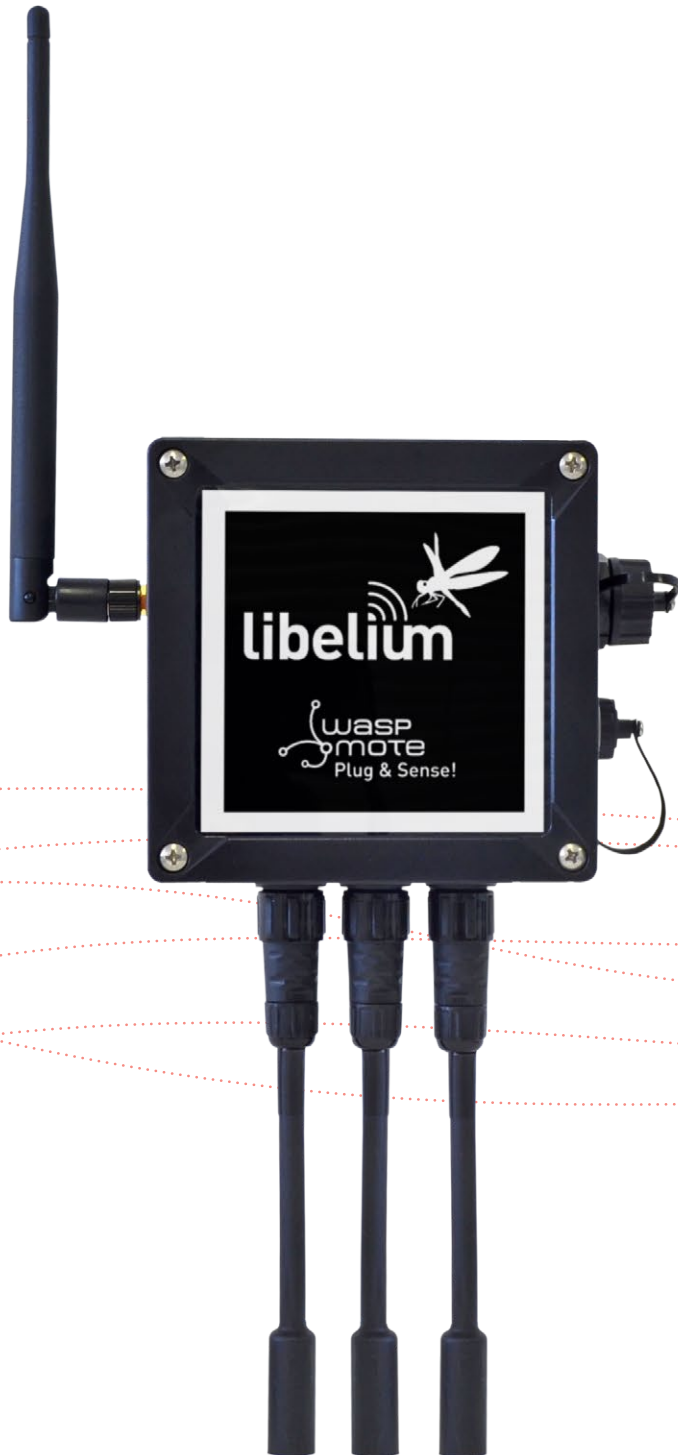


Wasp mote IDE

User Guide



Document Version: v7.0 -02/2017
© Libelium Comunicaciones Distribuidas S.L.

INDEX

| | |
|---|-----------|
| 1. Introduction | 3 |
| 1.1. New features | 3 |
| 1.2. Other notes | 3 |
| 2. Installation | 4 |
| 2.1. Windows | 4 |
| 2.2. Mac OS X | 5 |
| 2.3. Linux | 6 |
| 3. Environment | 8 |
| 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 |
| 4. OTA compatibility | 12 |

1. Introduction

This guide explains the IDE features. The IDE v05 is compatible with both old and new product lines Waspmote (v12 and v15) and Plug & Sense! (v12 and v15), released on October 2016.

Anyway, 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](#)".

The aim of this Guide is to introduce the user to the 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 is 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 v05, and Waspmote API versions higher or equal to v25. Waspmote IDE versions higher or equal to v05 are valid for Waspmote v12 or v15 and Waspmote Plug & Sense!.

1.1. New features

There are several benefits to use the new Waspmote IDE:

- Line number display
- Error line remarking
- Serial plotter
- New AVR compiler version

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 libraries organization for Waspmote, 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, they need to have 'n' IDEs installed in the computer. The 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 Wasmote IDE in your Operating System.

First, download the Wasmote IDE from the Wasmote software webpage:

http://www.libelium.com/development/wasmote/sdk_applications/

And then select your OS:



» Wasmote Pro IDE - v.XX

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

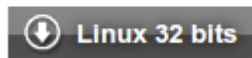


Figure : Wasmote IDE download links

If you are using Linux and do not know if you should download the 32-bit or 64-bit version, you can open a terminal and type `uname -p` to find out (it will probably be the 2nd option).

2.1. Windows

Unzip

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

Prepare the computer

Then connect your Wasmote board to the computer with the mini-USB cable. When you connect the board, Windows should start the driver installation process (if you haven't used the computer with Wasmote 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 have been installed correctly, you should see in *Start* → *Devices and printers* a device called *FT232R USB UART*:

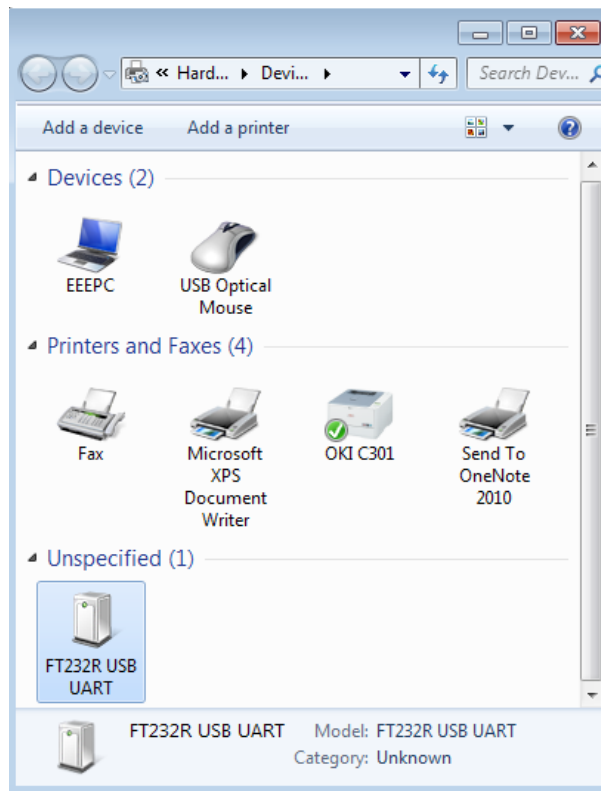


Figure : Waspote drivers installed in Windows

Launch Waspote IDE

Double-click on the IDE application.

Note: If the Waspote software loads in the wrong language, you can change it in the preferences menu. See the “Environment” section for more details.

2.2. Mac OS X

Unzip

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

Prepare the 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 on the package and follow the instructions in the installer. You will need to restart your computer after installing the drivers.

Then connect your Waspote board to the computer with the mini-USB cable, and the Mac OS will detect it automatically.

Launch Wasmote IDE

Double-click on the IDE application.

Note: If the Wasmote software loads in the wrong language, you can change it in the preferences menu. See the “Environment” section for more 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.

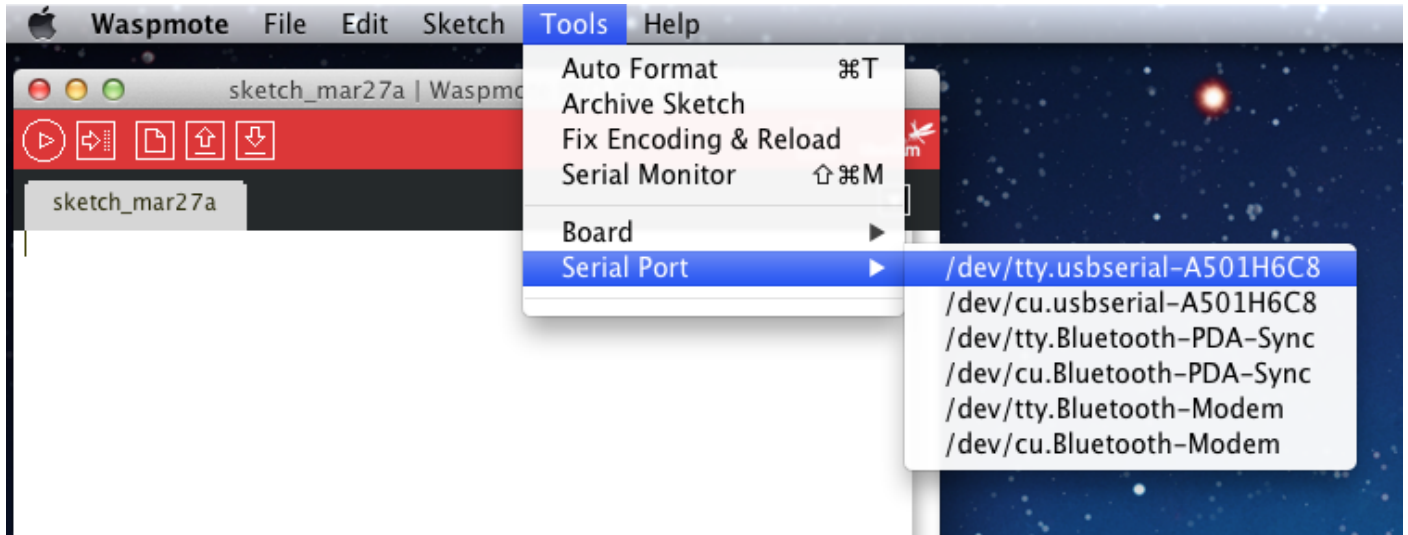


Figure : Wasmote correctly detected by Mac OS

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 you use 4.7.2 version or newer.
- `librxtx-java` package.

Then connect your Wasmote board to the computer with the mini-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 14.04



Figure : Wasmote IDE on Linux Unity desktop

Other notes:

• Locking assertion failure from Java when using Xinerama

To fix, run this:

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

• USB-Serial devices not detected

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

• Kernel configuration

If you configured your own kernel, the driver is in one of these 2 paths:

- Device Drivers → USB Support → USB Serial Converter Support → USB FTDI Single Port Serial Driver
- 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 access 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 Wasp mote IDE.

3.1. Writing sketches

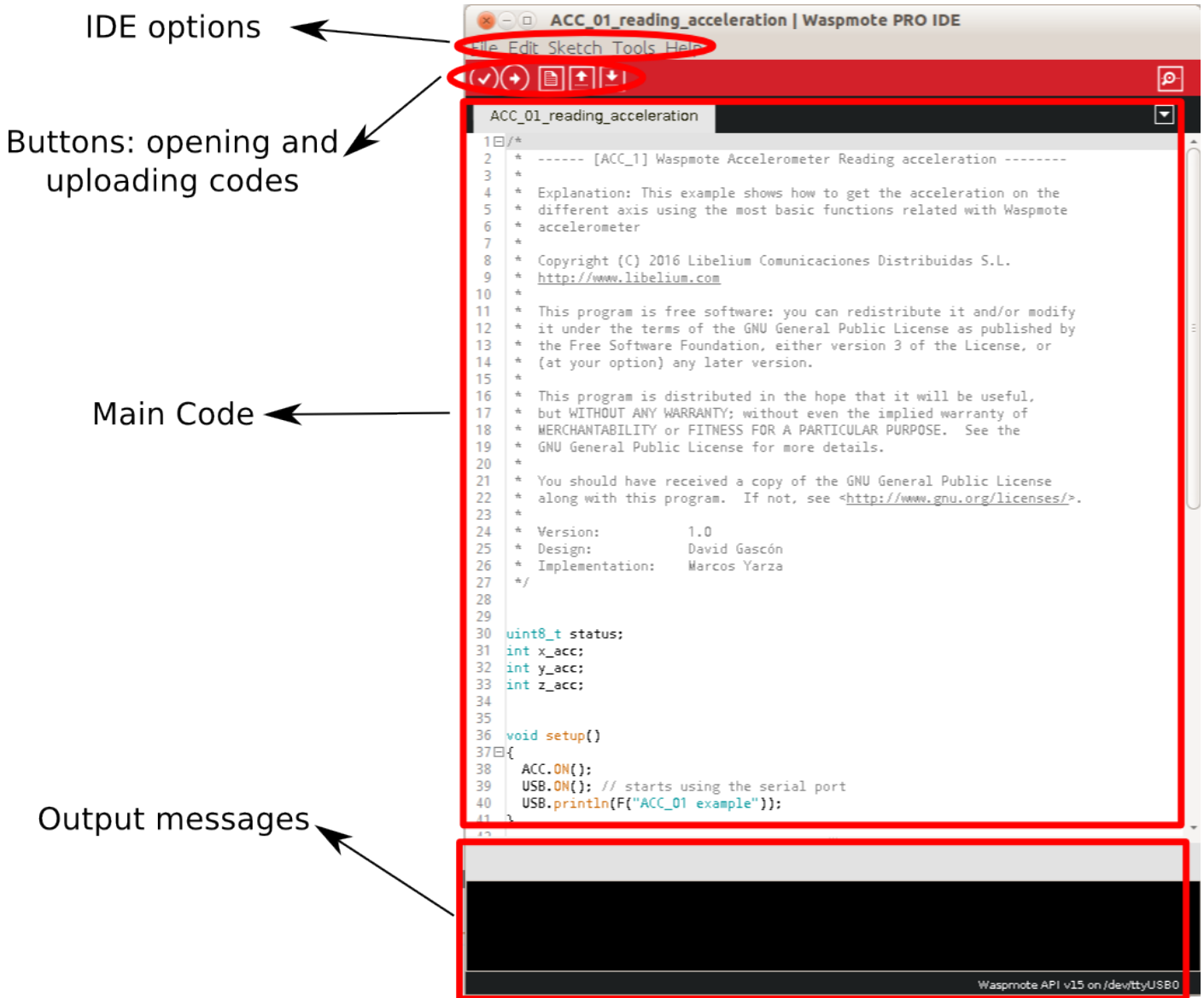


Figure: Wasp mote IDE sections

Software units 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-right corner of the window displays the current board and serial port.

This IDE allows you to manage more than one file at the same time (in the tab zone). These can be normal Wasp mote 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 after that uploads it to the Wasp mote board.



- *New*: Creates a new sketch.



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



- *Save*: Saves changes on 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 menu* and the *Tools* → *Serial Port* menu.

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. Wasp mote will reset automatically and begin the compile and upload process.

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 Wasp mote API. Others can be downloaded from a variety of sources.

Adding 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 add the library, first quit the Wasp mote 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 are not in a folder, you will need to create one. In this case, you have to make a folder called "MyLibrary". Drag the MyLibrary folder and drop it into the IDE libraries folder. There may be more files than just the .cpp and .h files, just make sure they are 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 port) or from a Gateway unit..

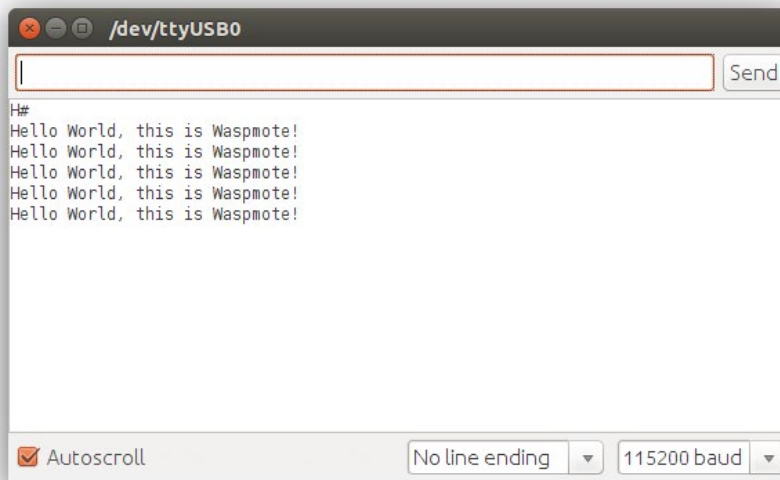


Figure : Serial Monitor

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

For advanced use, if you need to send data to the board, enter the text and click on the 'Send' button or press enter.

3.6. Preferences and language support

Some preferences can be set in the Preferences menu (found under the *Wasp mote menu* on the Mac, or File on Windows and Linux). The rest of the options (for example, the window size) can be found in the preferences file, whose location is shown in the preference dialog. If you want to edit this preferences file, make sure you closed the IDE before editing.

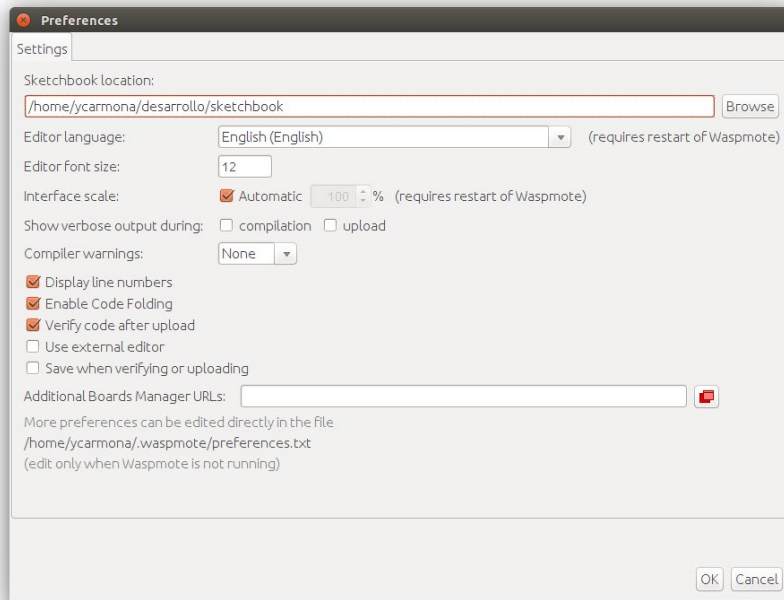


Figure : Preferences and Language Support

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

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

4. OTA compatibility

The Waspnote IDE v05 generates the binary file to be used in OTA programmings. There is a specific option in the Waspnote 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:

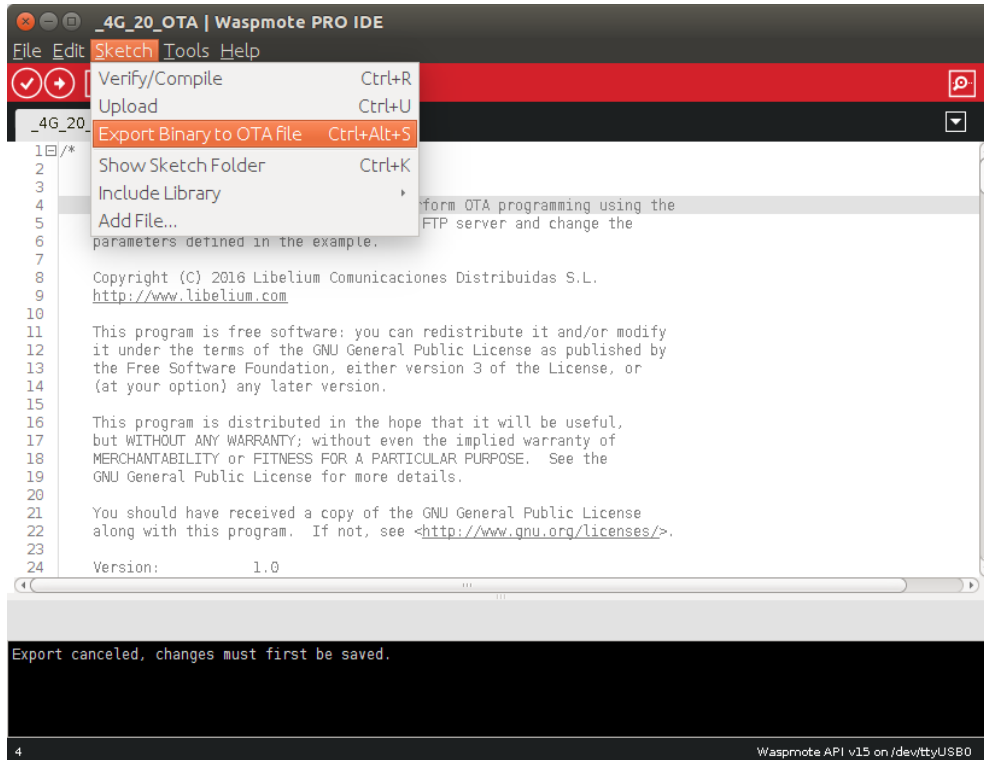


Figure : Waspnote IDE generates binary files

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

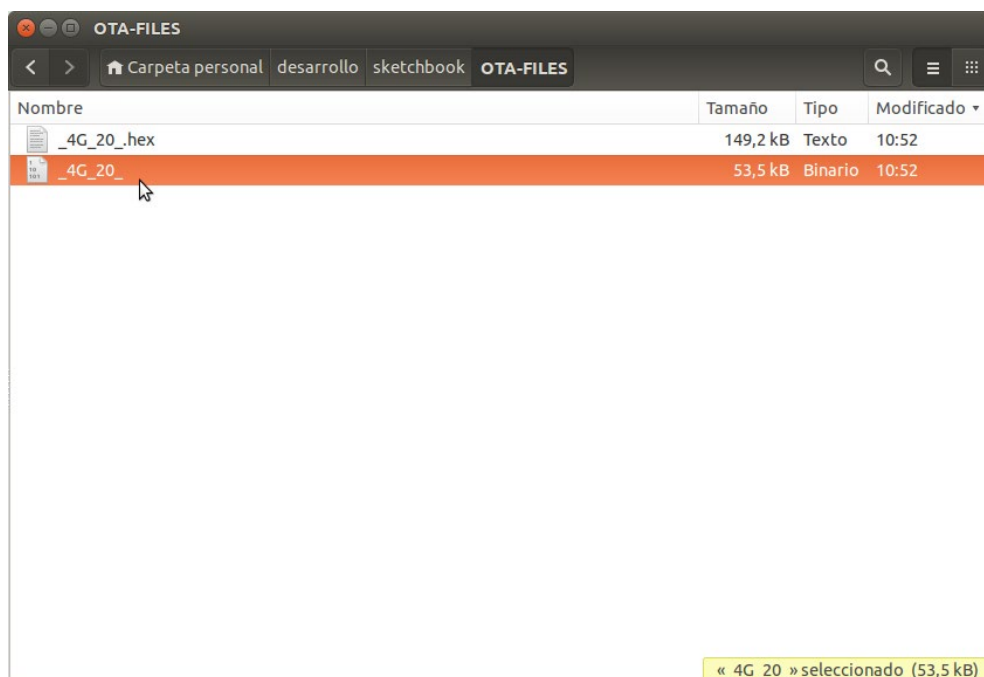


Figure : OTA-FILES folder

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.

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 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. More information about OTA programming in <http://www.libelium.com/ota/>