ARROW Electronics EMEA - ESC

ASME Lion – LoRa demo

User Guide

Kristóf Féja – Application Engineer
1/27/2017

Table of Contents

I.   Introduction

This document describes the necessary steps to set up the Lion board as a LoRa end point. You can run the demo without a LoRa gateway and an account at Digimondo's FireFly IoT (LoRa) platform. In this case You will only be able to see the LoRa messages sent by the Lion board on the serial port.

(For acquiring a gateway and an account please contact the Arrow office of your region.)

II.   Hardware prerequisites

- Arrow SmartEverything Lion board
- X-NUCLEO-IKS01A1
  - LSM6DS0 – MEMS 3D accelerometer
  - LIS3MDL – MEMS 3D magnetometer
  - LPS25HB – MEMS pressure sensor
  - HTS221 – humidity sensor

1) Attach the antenna and place the sensor shield on top of the Lion board!
2) Connect the board to your computer's USB port using the mini-USB port on the board.
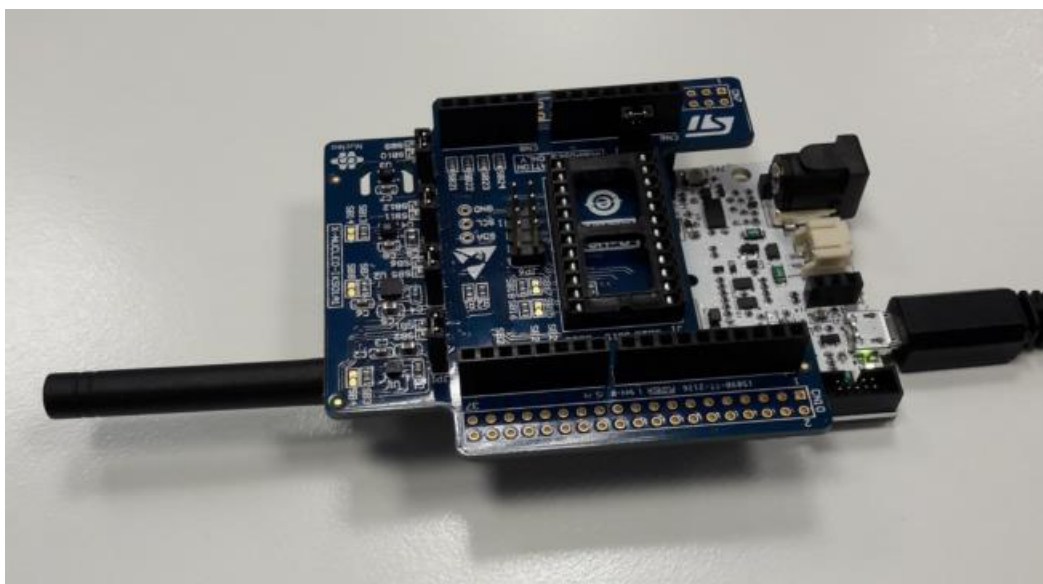
III.   Software prerequisites

- Arduino IDE (check Lion board's User Manual for link)

IV. Installing sensor libraries in Arduino IDE

This section will guide You through how to install the necessary extra libraries.

1. The correct setup of Arduino IDE is explained in the Lion board's User Manual. This is a necessary first step.
2. Open Arduino IDE
3. Go to Sketch -> Include Library -> Manage Libraries…
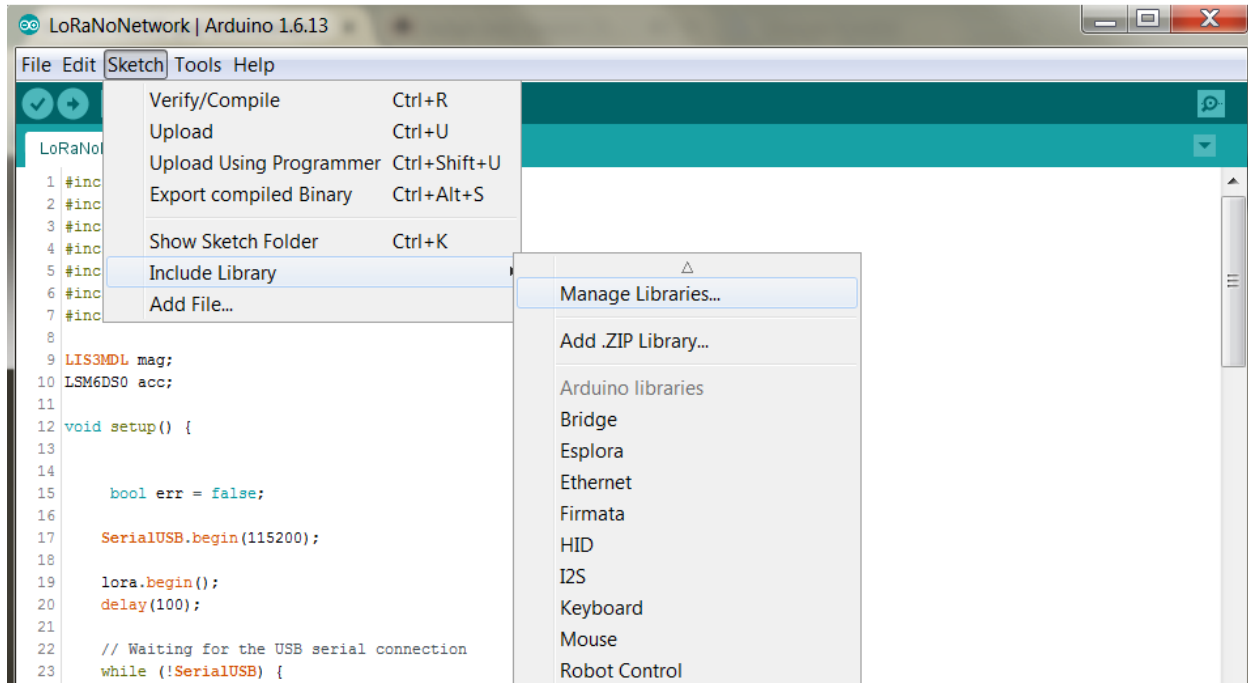
4. Type the following part numbers in the upper-right search box.

LSM6DS0, LIS3MDL, LPS25HB, HTS221

For each search result click Install. (For HTS221 install: SmartEverything HTS221)
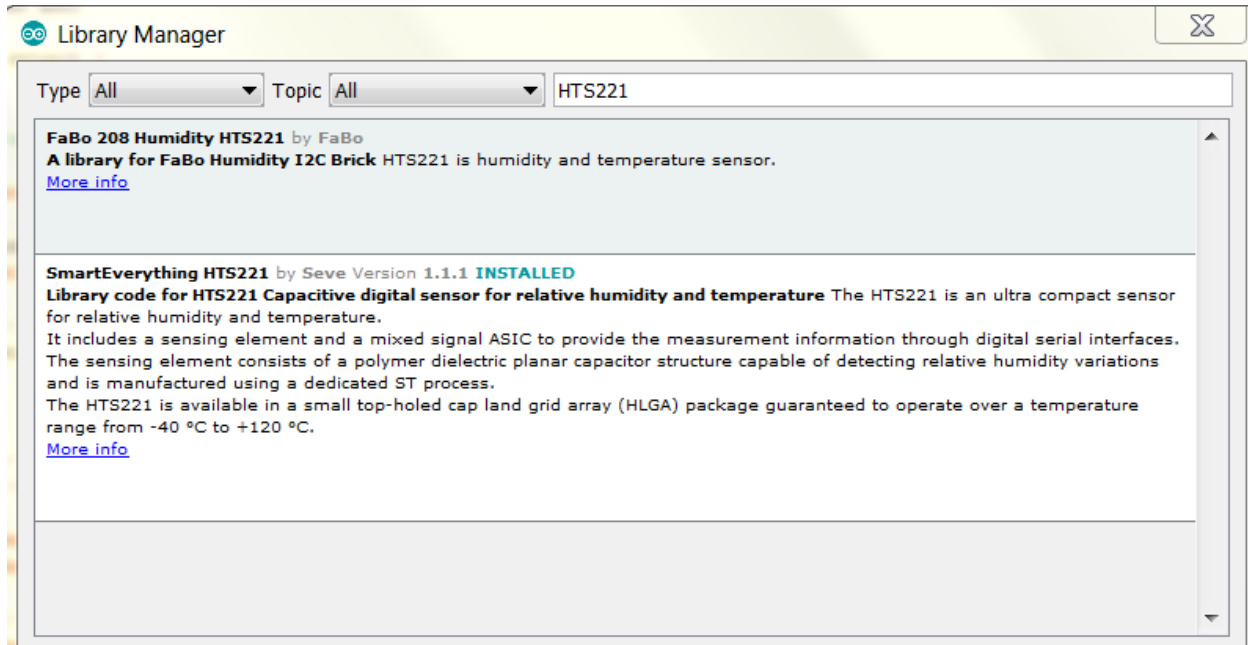
V.   Necessary modifications in the code

LPS25H sensor library

Locate "LPS25HReg.h" file on your computer. In line 13 You can see the following:

```
#define LPS25H_ADDRESS      0x5C
```

Change this value to 0x5D.

This is necessary as the library downloaded is written for a slightly different hardware setup.

Getting sensor feedback

Open LoRaDemo.ino file with Arduino IDE!

It is possible to see the sent sensor data. In order to do so, delete the comment symbol (//) before #define SENSOR_FEEDBACK (line 10).

With or without a gateway

As it was explained earlier it is possible to run the demo without a gateway. If this is the case then make sure You comment out: **#define LIVE** in the 9th line in **LoRaDemo.ino**.

```
7  #include <LIS3MDL.h>
8
9  //#define LIVE  //If defined the Lion board will try to connect to a gateway.
10
```

This way the following part of the code will not run:

```
139 // Activation procedure
140     #ifdef LIVE
141     String response = "denied";
142     while(response[0] == 'd') // If server response is "denied" try again
143     {
144         while (lora.sendRawCmd(join)) // Sending join command
145         {
146           SerialUSB.println("\nOTA JOIN FAILED "); // Join command sending not successful
147           delay(5000);
148         }
149         SerialUSB.println("\nOTA Network JOINED! "); // Join command sent
150         while(!lora.available()) // Waiting for server response
151         {
152          ;
153         }
154
155         response = lora.read(); // Reading server response
156         SerialUSB.print("\nRx> ");
157         SerialUSB.println(response);
158     }
159     #endif
```

Meaning that the board will not attempt connecting to a network server. It will still send LoRa messages but there will be no channel set up through which the message could be forwarded to the cloud.

Still, You can see the created message on the serial port.

ABP or OTAA join

You can test either join method that is ABP – Activation By Personalization or OTAA – Over The Air Activation.
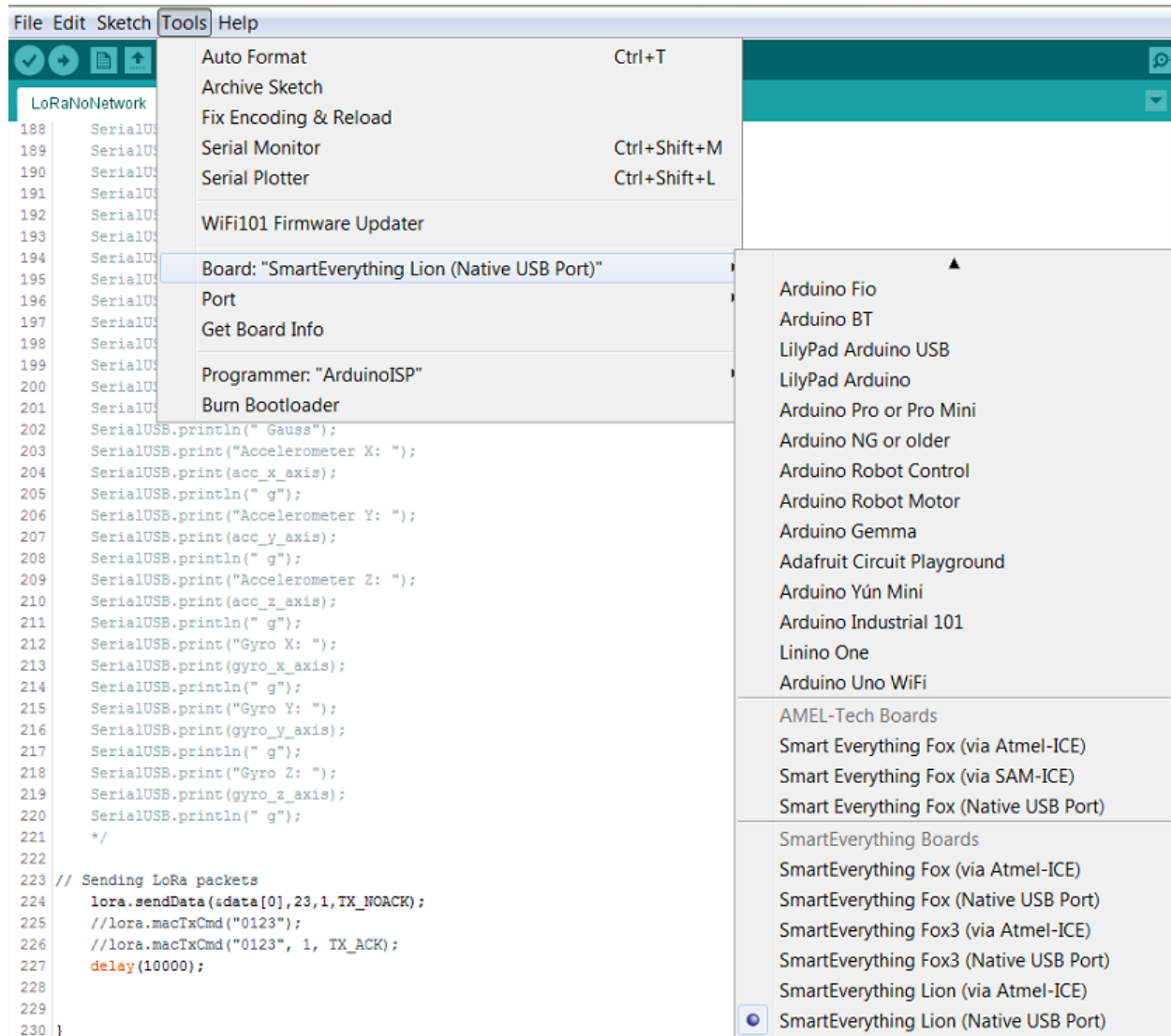
```
9  #define ABP // Either ABP or OTAA is allowed to be defined, the other has to be commented out!
10 //#define OTAA
```
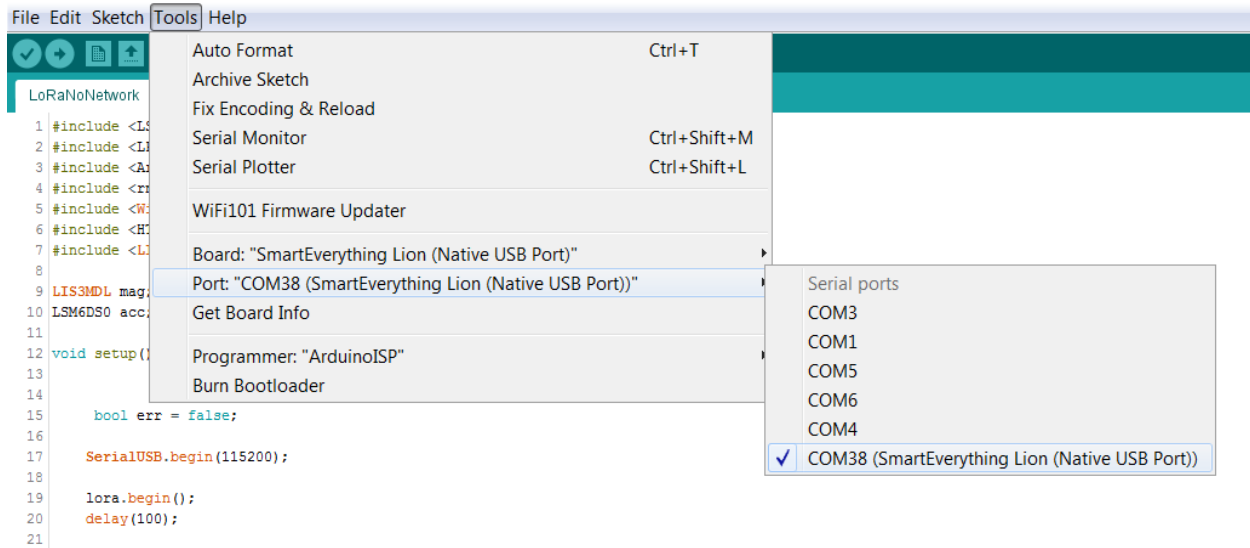
VI.   Flashing the Lion board

When You are done modifying the code it's time to load it up into the board.
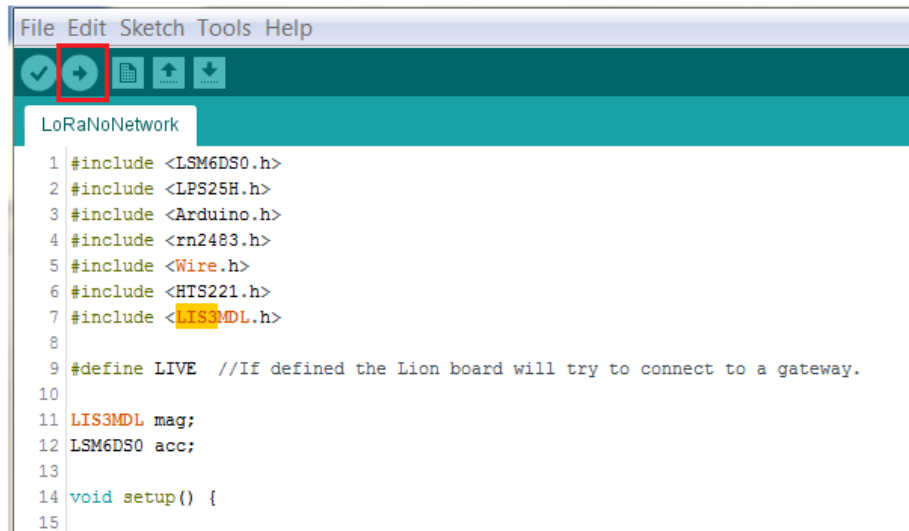
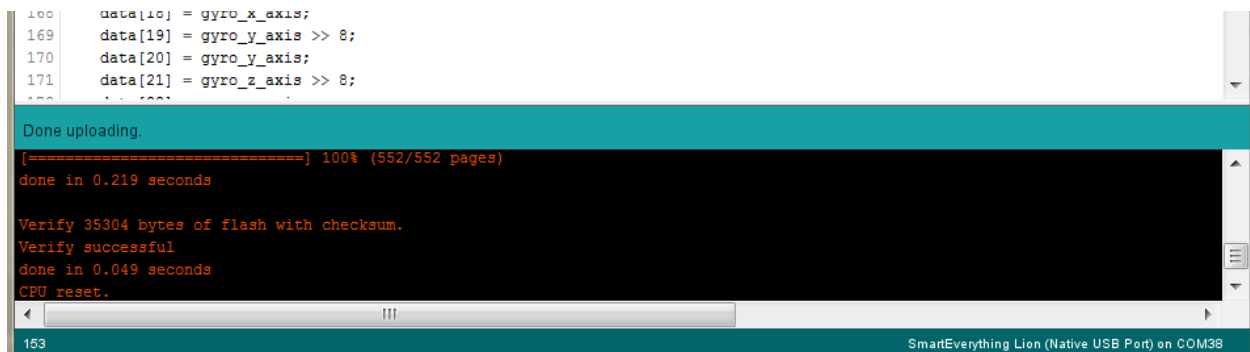1. In Arduino IDE select: Tools -> Board: -> "SmartEverything Lion (Native USB port)"

2. Go to: "Tools -> Port: " and select the serial port your Lion board is connected to.

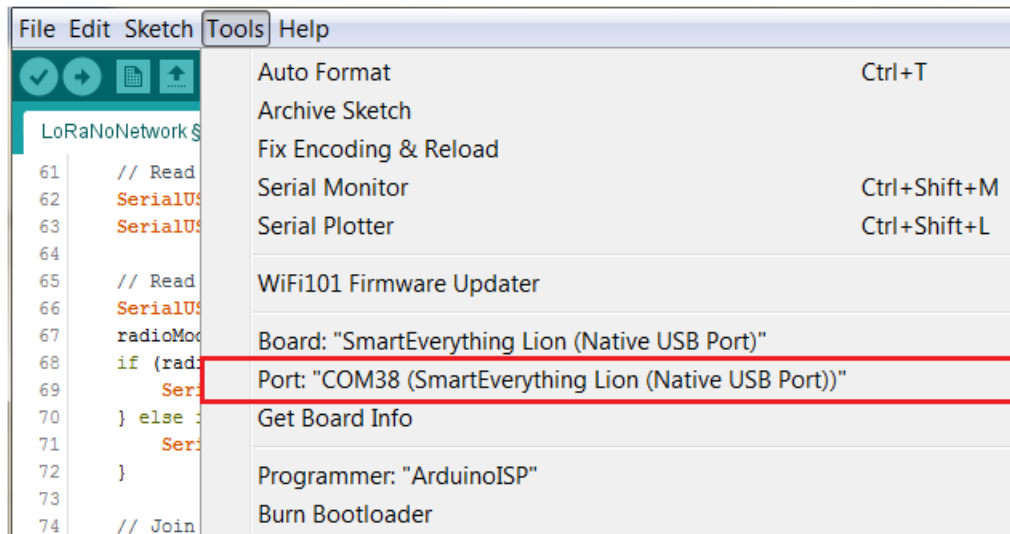3) Now You can verify and upload the code by pressing the arrow button:

If it's successful You should receive this feedback:

VII.   Serial monitor

To open a serial monitor in Arduino IDE follow these steps:

1. Make sure that You have the right port selected. Check "Tools -> Port:"! It should look something like this:

2. Then You can open the corresponding COM port's serial monitor.
   (Tools -> Serial Monitor)

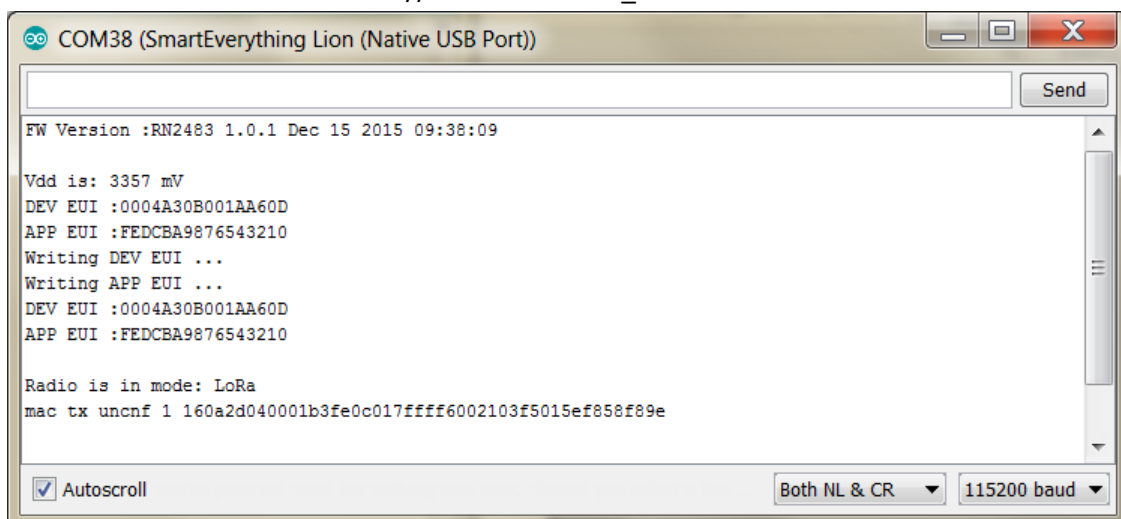3. After 10 seconds You should see something similar to this:

a) Without gateway, no sensor feedback

Make sure to use the same serial port settings as You can see below in the lower-right corner!

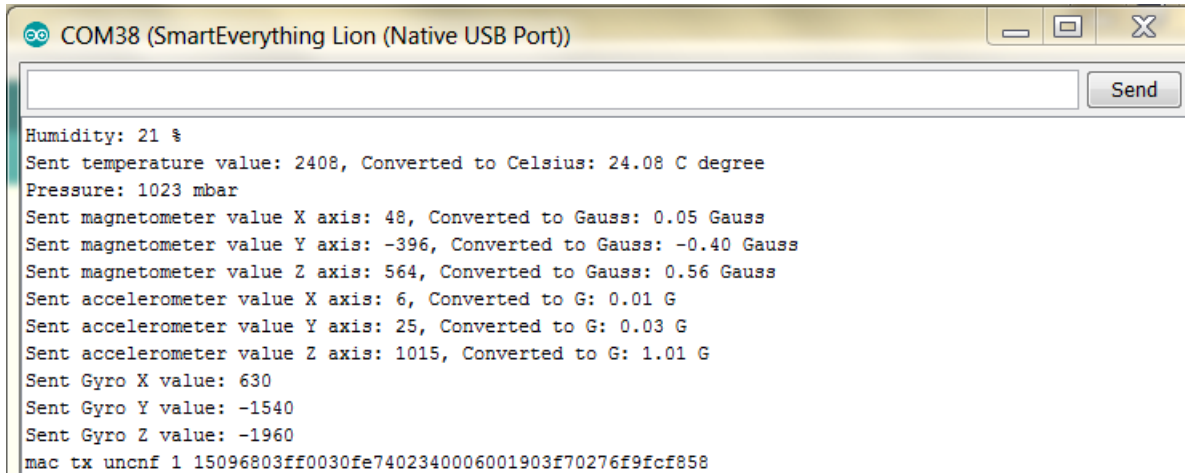//#define LIVE

//#define SENSOR_FEEDBACK

b) Sensor feedback enabled

If You have enabled sensor feedback, You should see something like this:

#define SENSOR_FEEDBACK

```
COM38 (SmartEverything Lion (Native USB Port))                    [_] [□] [X]

[                                                           ]  [ Send ]

Humidity: 21 %
Sent temperature value: 2408, Converted to Celsius: 24.08 C degree
Pressure: 1023 mbar
Sent magnetometer value X axis: 48, Converted to Gauss: 0.05 Gauss
Sent magnetometer value Y axis: -396, Converted to Gauss: -0.40 Gauss
Sent magnetometer value Z axis: 564, Converted to Gauss: 0.56 Gauss
Sent accelerometer value X axis: 6, Converted to G: 0.01 G
Sent accelerometer value Y axis: 25, Converted to G: 0.03 G
Sent accelerometer value Z axis: 1015, Converted to G: 1.01 G
Sent Gyro X value: 630
Sent Gyro Y value: -1540
Sent Gyro Z value: -1960
mac tx uncnf 1 15096803ff0030fe7402340006001903f70276f9fcf858
```
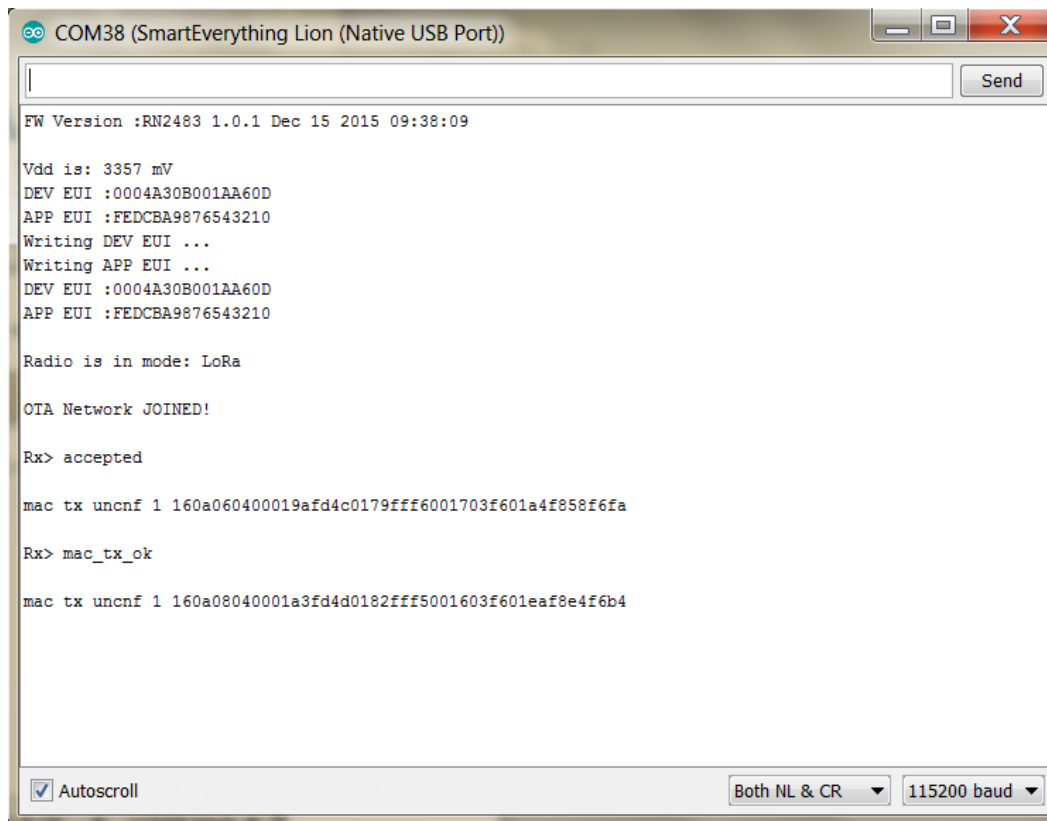
	Humidity	Temperature	Pressure	Magneto X	Magneto Y	Magneto Z	Accelero X	Accelero Y	Accelero Z	Gyro X	Gyro Y	Gyro Z
hex value	15	09 68	03 FF	00 30	FE 74	02 34	00 06	00 19	03 F7	02 76	F9 FC	F8 58
decimal value	21	2408	1023	48	-396	564	6	25	1015	630	-1540	-1960

Every sent sensor data is 2 bytes long, except for humidity which is only 1 byte long.

c) With Live system

If You have a complete environment set up (meaning a gateway & an account in Digimondo's system):