



# Basic IoT Board

## User Guide

Ver 2.0

Copyright 2016  
Nettrotter

[www.nettrotter.io](http://www.nettrotter.io)  
[www.axeletttronica.it](http://www.axeletttronica.it)

## Copyright Notice and Disclaimer

All rights reserved. No parts of this manual may be reproduced in any form without the express written permission of Nettrotter.

Nettrotter makes no representations or warranties with respect to the contents hereof. In addition, information contained herein are subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, Nettrotter assumes no responsibility for errors or omissions or any damages resulting from the use of the information contained in this publication.

Nettrotter does not assume any liability arising out of the application or use of any of its products or designs. Products designed or distributed by Nettrotter are not intended for, or authorized to be used in, applications such as life support systems or for any other use in which the failure of the product could potentially result in personal injury, death or property damage.

## Revision Sheet

Release No.	Date	Revision Description
Ver 1.0	07/06/2016	First Release
Ver 2.0	07/03/2017	

Table 1: Document History

## Table of contents

Chapter 1	Introduction	5
Chapter 2	Hardware	5
2.1.	External View	5
2.2.	SmartTutto Block Diagram	6
2.3.	ATMEL SAMD21 Ultra low-power ARM® Cortex®-M0+ MCU	6
2.3.1	ATMEL SAMD21 Internal Memory	7
2.4.	Sigfox Module (Telit LE51-868 S)	7
2.4.1	Main Features	8
2.5.	GPS Module with Embedded Antenna (Telit Jupiter SE868-A)	8
2.6.	Axis sensors (ST LSM9DS1)	8
2.7.	Board Interfaces and Connector	9
Chapter 3	Mechanical Information	10
3.1.	Mechanical Characteristics	10
Chapter 4	Software Development	10
Chapter 5	Getting Started with Arduino IDE and Sketch Projects	11
5.1.	Tools	11
5.2.	Setup the Environment	11
5.3.	Importing the Smart Everything Board Library	13
5.4.	Importing the Smart Everything Board Library	14
5.5.	Example Sigfox mode EU	15
5.6.	Example Sigfox manage GPS	19
Chapter 6	References and Useful Links	19
6.1.	Data sheets	19
6.2.	Tools	19
6.3.	Web Sites	19
Chapter 7	Troubleshooting	20
7.1.	Driver installation problems	20
7.2.	Block in bootloader mode	20

## Tables Index

Table 1: Document History _____	2
Table 2: Board Specifications _____	5
Table 3: Arduino Compatible Headers _____	9

## Illustrations Index

Figure 1: Front View _____	5
Figure 2: SmartTutto Block Diagram _____	6
Figure 3: Headers pinout _____	9
Figure 4: Dimensions _____	10
Figure 5: Launch Board Manager _____	11
Figure 6: Arduino Zero Core Installation _____	12
Figure 7: SmartEverything Core Installation _____	12
Figure 8: Driver Installation _____	13
Figure 9: Driver correctly recognized _____	13
Figure 10: ArduinoIDE: Launch Manage Libraries interface _____	13
Figure 11: ArduinoIDE: Library Manager _____	14
Figure 12: ArduinoIDE: Launch Manage Libraries interface _____	15

## Chapter 1: Introduction

This document describes Nettrotter Basic IoT Board (BIB) based on Atmel D21 Ultra low-power microcontroller using the 32-bit ARM® Cortex®-M0+ processor.

BIB is supported by the Arduino IDE for a fast and easy software development cycle. The software can also be developed using the Atmel Studio IDE commonly preferred by professional software engineers.

### 1.1. Board Specifications

Characteristics	Value
Clock speed	48 MHz
Connector	microUSB
Board supply voltage	5V
Dimensions	68,58mm x 53,34 mm
RoHS status	Compliant

Table 2: Board Specifications

## Chapter 2: Hardware

### 2.1. External View

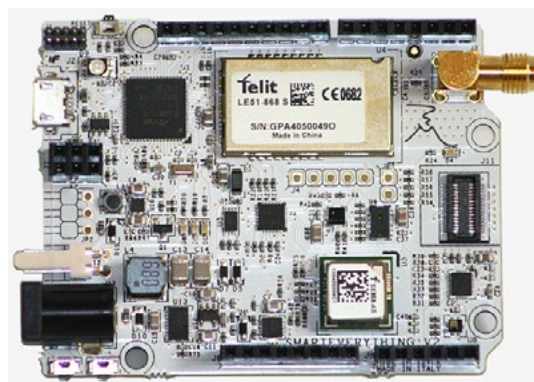


Figure 1: Front View

## 2.2. SmartTutto Block Diagram

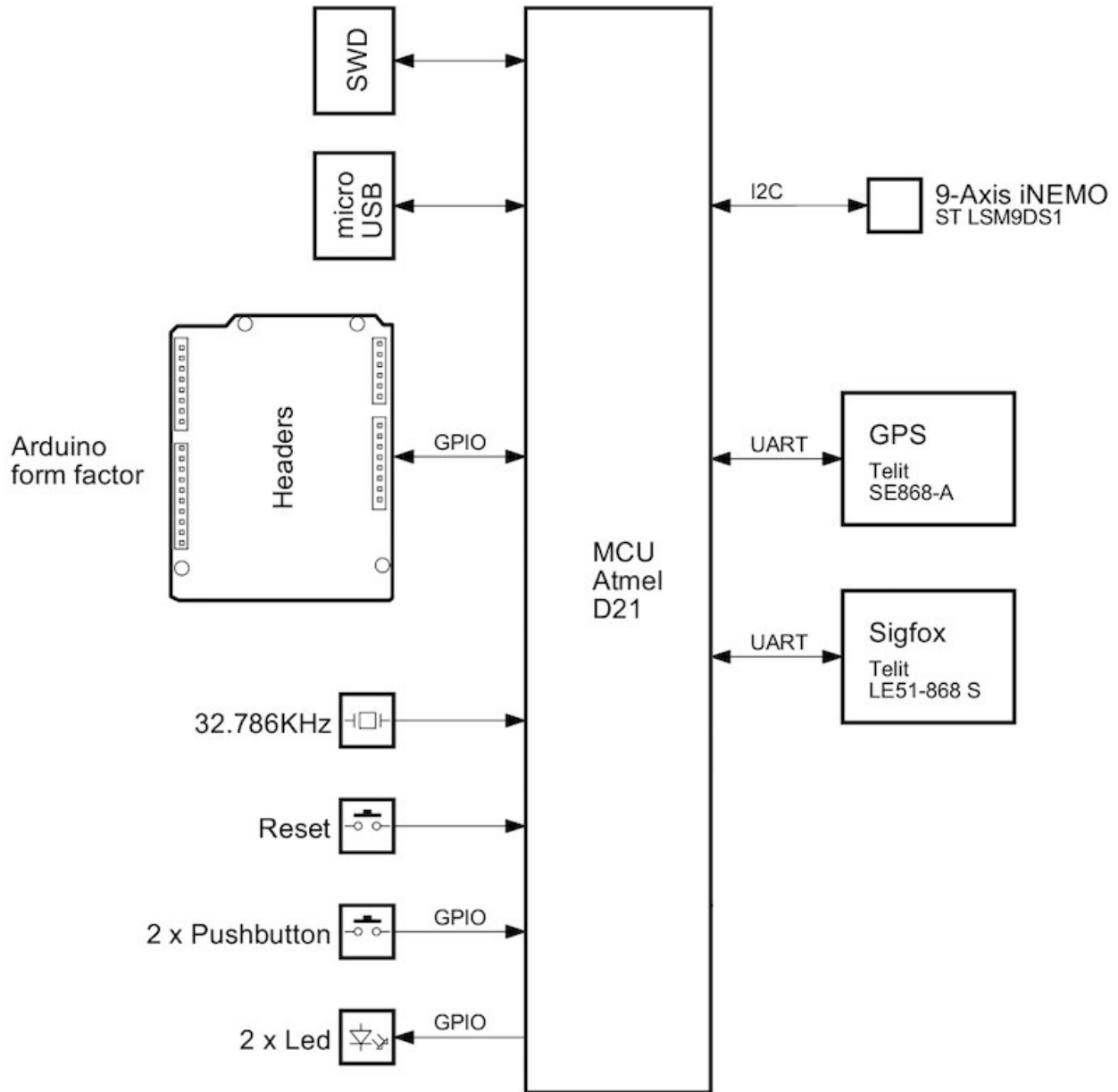


Figure 2: SmartTutto Block Diagram

## 2.3. ATMEL SAMD21Ultra low-power ARM® Cortex®-M0+ MCU

Atmel D21 Ultra low-power microcontroller using the 32-bit ARM® Cortex®-M0+ processor is the core of the entire board.

## 2.3.1. ATMEL SAMD21 Internal Memory

- 256KB in-system self-programmable Flash
- 32KB SRAM Main Memory
- 12KB SRAM Low power Memory.
- 16-channel Direct Memory Access Controller (DMAC)
- 12-channel Event System
- Up to five 16-bit Timer/Counters (TC), configurable as either:
  - One 16-bit TC with compare/capture channels
  - One 8-bit TC with compare/capture channels
  - One 32-bit TC with compare/capture channels, by using two TCs
- Three 16-bit Timer/Counters for Control (TCC), with extended functions:
  - Up to four compare channels with optional complementary output
  - Generation of synchronized pulse width modulation (PWM) pattern across port pins
  - Deterministic fault protection, fast decay and configurable dead-time between complementary output
- 32-bit Real Time Counter (RTC) with clock/calendar function
- Watchdog Timer (WDT)
- CRC-32 generator
- One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 interface- Embedded host and device functions
  - Eight endpoints
- One True Random Generator (TRNG)
- One Configurable Custom Logic (CCL)
- One 12-bit, 1Msps Analog-to-Digital Converter (ADC) with up to 20 channels
  - Differential and single-ended input
  - Automatic offset and gain error compensation
  - Oversampling and decimation in hardware to support 13-, 14-, 15- or 16-bit resolution
- 12-bit, 1Msps Dual Output Digital-to-Analog Converter (DAC)
- Two Analog Comparators (AC) with window compare function
- Three Operational Amplifiers (OPAMP)

## 2.4. Sigfox Module (Telit LE51-868 S)

Telit LE51-868 S module is a high performance module designed to cover the 863-870MHz unlicensed band. It provides the Telit proprietary Star Network protocol and it is able to act as a certified Sigfox™ wireless node.

The following protocol stack is preloaded:

- LE51-868 S SIGFOX™ Network Software.
- “Star Network” Protocol stack

### 2.4.1. Main Features

The LE51-868 S module is a complete solution from serial interface to RF interface. The LE51-868 module has a digital part and a RF part. The radio link on Sigfox network is a Half-Duplex bidirectional link.

The digital part has the following functionalities:

- Communication interface
- I/O management
- Micro controller with embedded Telit Software Stack supporting Sigfox protocol

The RF part has the following functionalities:

- Frequency synthesis
- Front-end
- Power amplification
- Packet handling

More information can be found on relevant Data Sheet (See the link in the References and Useful Links chapter)

## 2.5. GPS Module with Embedded Antenna (Telit Jupiter SE868-A)

The Telit Jupiter SE868-A is a GPS Module designed to fully support GPS, QZSS, GLONASS and it is Galileo ready. It has an embedded SMT antenna and it is able to track GPS + GLONASS (and eventually Galileo) constellations simultaneously and to provide the position through the standard serial interface (UART)

The module software can increase the position accuracy supporting:

- Ephemeris file injection (A-GPS)
- Satellite Based Augmentation System (SBAS)

More information can be found on relevant Data Sheet (See the link in the References and Useful Links chapter).

## 2.6. Axis sensors (ST LSM9DS1)

The LSM9DS1 is a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor.

The LSM9DS1 has a linear acceleration full scale of  $\pm 2g/\pm 4g/\pm 8g$ , a magnetic field full scale of  $\pm 4/\pm 8/\pm 12/\pm 16$  gauss and an angular rate of  $\pm 245/\pm 500/\pm 2000$  dps.



It communicates with the Atmel MCU through the I2C interface.

More information can be found on relevant Data Sheet (See the link in the References and Useful Links chapter)

## 2.7. Board Interfaces and Connector

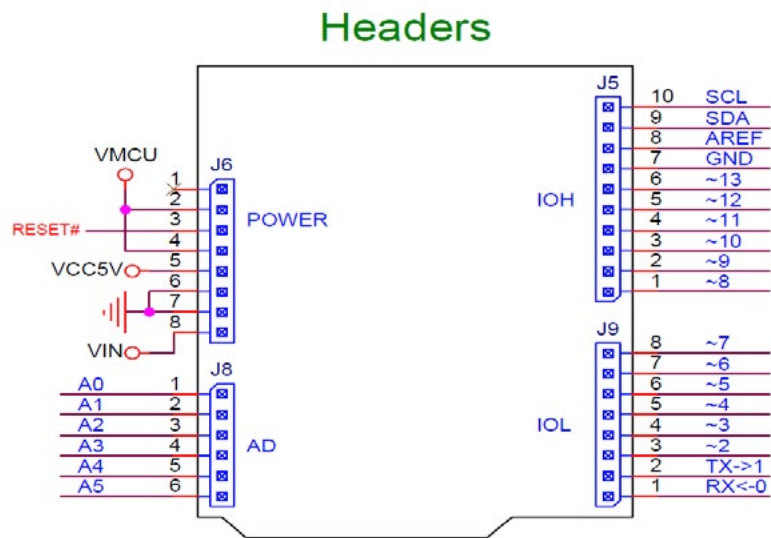


Figure 3: Headers pinout

Name	Conn.	Pin	Description	Name used in the sketch
D8	J5	1	Digital I/O D8 / PWM	8
D9	J5	2	Digital I/O D9 / PWM	9
D10	J5	3	Digital I/O D10 / PWM	10
D11	J5	4	Digital I/O D11 / PWM	11
D12	J5	5	Digital I/O D12 / PWM	12
D13	J5	6	Digital I/O D13 / PWM	13
GND	J5	7	Ground pin	
AREF	J5	8	Analogue Reference (used by ADC)	
SDA	J5	9		
SCL	J5	10		
NC	J6	1	Not Connected	
VMCU	J6	2		
RESET	J6	3	Reset	
VMCU	J6	4		
Vcc 5V	J6	5		
GND	J6	6	Ground pin	
GND	J6	7	Ground pin	
Vin	J6	8	External Power Supply Input	
A0	J8	1	Analog Input A0	A0
A1	J8	2	Analog Input A1	A1
A2	J8	3	Analog Input A2	A2
A3	J8	4	Analog Input A3	A3
A4	J8	5	Analog Input A4	A4
A5	J8	6	Analog Input A5	A5
D0 / RX	J9	1	Digital I/O D0 / Serial1 (RX)	1
D1 / TX	J9	2	Digital I/O D1 / Serial1 (TX)	2
D2	J9	3	Digital I/O D2 / PWM	3
D3	J9	4	Digital I/O D3 / PWM	4
D4	J9	5	Digital I/O D4 / PWM	5
D5	J9	6	Digital I/O D5 / PWM	6
D6	J9	7	Digital I/O D6 / PWM	7
D7	J9	8	Digital I/O D7 / PWM	8

Table 3: Arduino Compatible Headers

## Chapter 3: Mechanical Information

### 3.1. Mechanical Characteristics

The maximum length and width of the BIB PCB are 6,85 cm and 5,33 cm respectively, with the USB connector and antenna extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16”), not an even multiple of the 100 mil spacing of the other pins. This makes the BIB fully compatible with most shields designed for the Arduino Uno, Diecimila or Duemilanove.

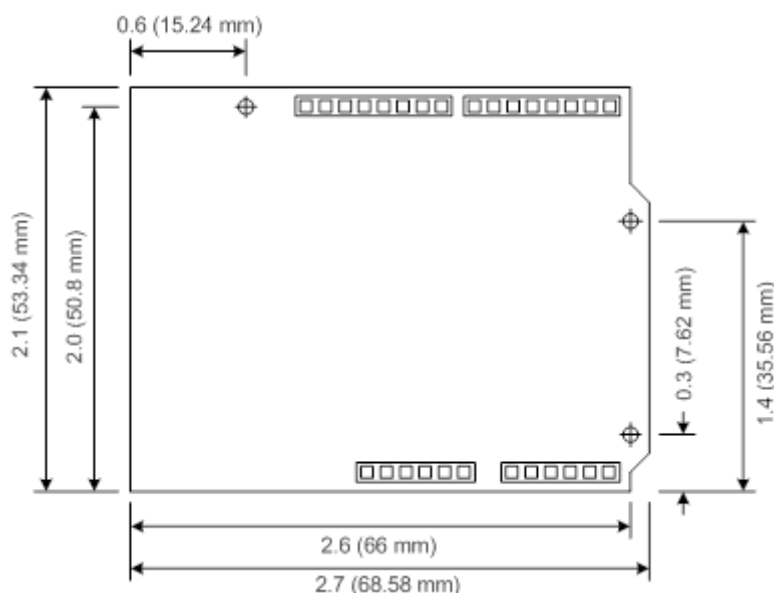


Figure 4: Dimensions

## Chapter 4: Software Development

The following chapters provides an overview about how a user can develop its software and run it on the BIB board.

There are two main ways to develop a software load and debug it on the card:

- Using the Arduino IDE and Sketch Projects
- Using the Atmel Studio and Standard C/C++ language

When developing a software running on a microcontroller it is important to have some tools to easily debug the code and fix what does not work as expected.

The Atmel SAMD21 provides the Atmel’s Embedded Debugger (EDBG): a full debug interface without the need for additional debugger. This feature makes easier the software debugging. EDBG supports a virtual COM port that can be used for device programming and allows a traditional Arduino boot loader functionality. The possibilities to use an external debugger like the JTAGICE3 is still available.

## Chapter 5: Getting Started with Arduino IDE and Sketch Projects

### 5.1. Tools

The following tools are needed:

- Arduino IDE (Release 1.6.4 or newer)
- USB cable

### 5.2. Setup the Environment

Download and install the Arduino IDE from the Arduino web site (See the links in the References and Useful Links chapter)

The first time you run the Arduino IDE it is necessary to load the Arduino Zero & SmartEverything Core (equivalent to the BIB board).

Click on the Tools à Boards à Boards Manager... menu entry

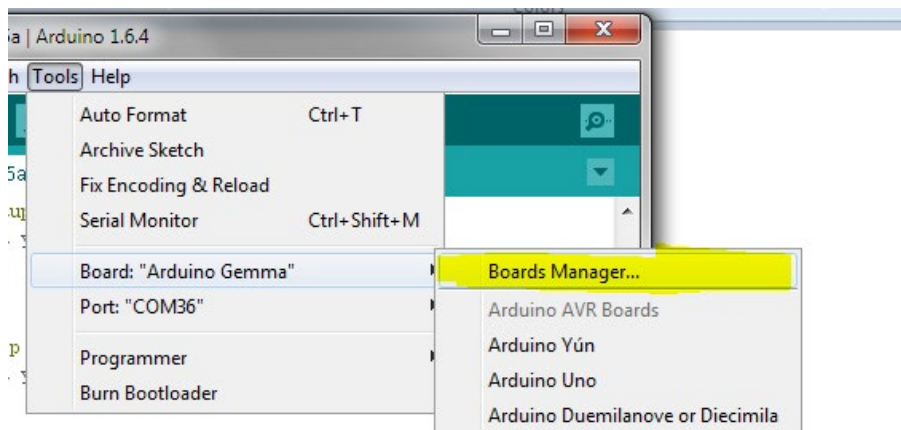


Figure 5: Launch Board Manager

The two steps of installation shall be done as following

- 1) Select from “Type” combo Arduino and choose Arduino SAMD Boards in order to install the core of the Arduino Zero.

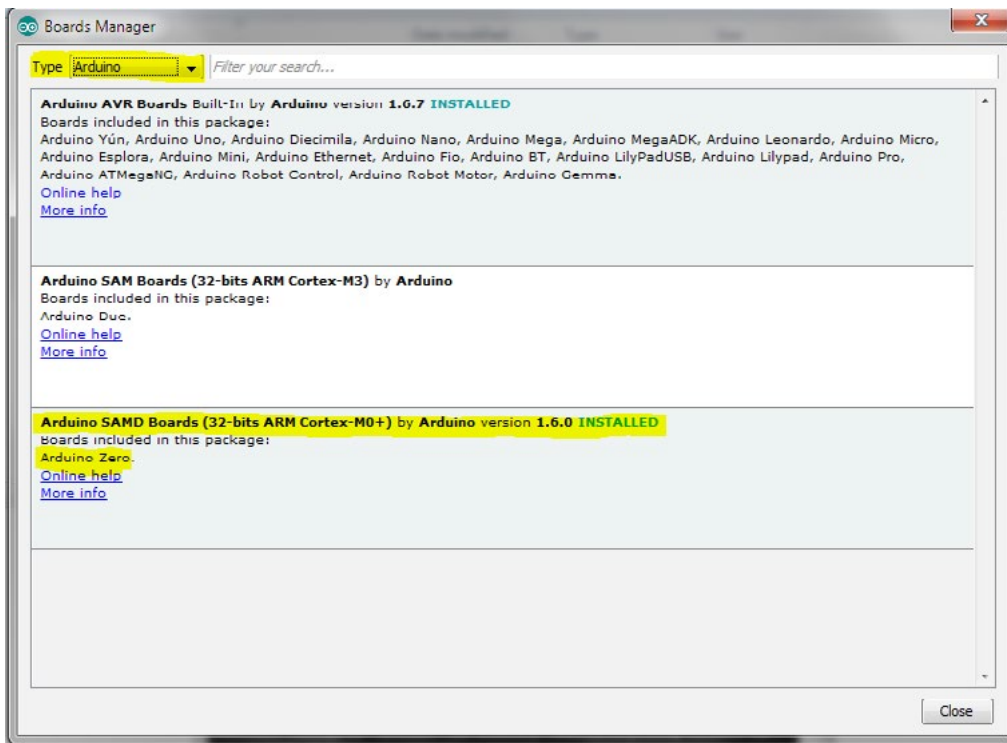


Figure 6: Arduino Zero Core Installation

- 2) Select from “Type” combo Partner and choose the ARROW version 2.1.0 or Higher in order to install the core of the SmartEverything.



Figure 7: SmartEverything Core Installation

Once the installation is completed you are able to connect the Board and continue with the following instructions.

Connect the SmartTutto board to the PC with the USB cable

The Driver installation will start automatically and should complete with no problem.

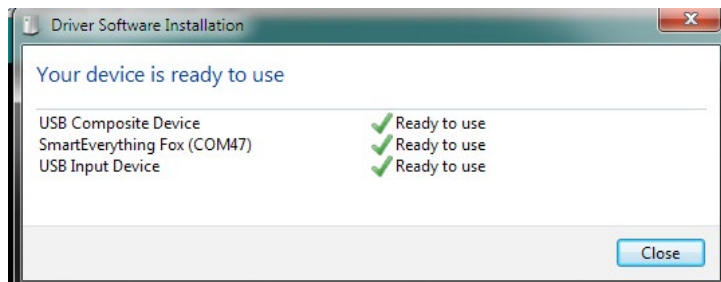


Figure 8: Driver Installation

At the end of the driver installation in the Control Panel à System and Security à System à Device Manager you will have a new COM Port

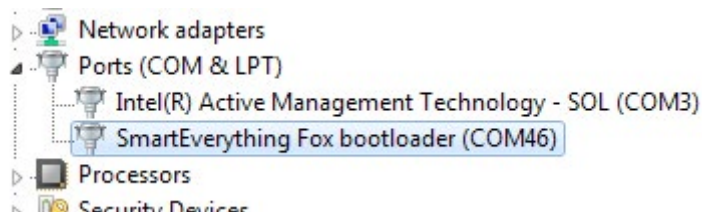


Figure 9: Driver correctly recognized

The name of the new COM port depend of the PC and it may differ among installations on different PC. If the Driver installation process does not complete successfully see the Chapter 7.

### 5.3. Importing the Smart Everything Board Library

The BIB board provides some libraries useful to interface with some of its main components. In order to include the Smart Everything library use the menu Sketch à Include Library à Manage Libraries.

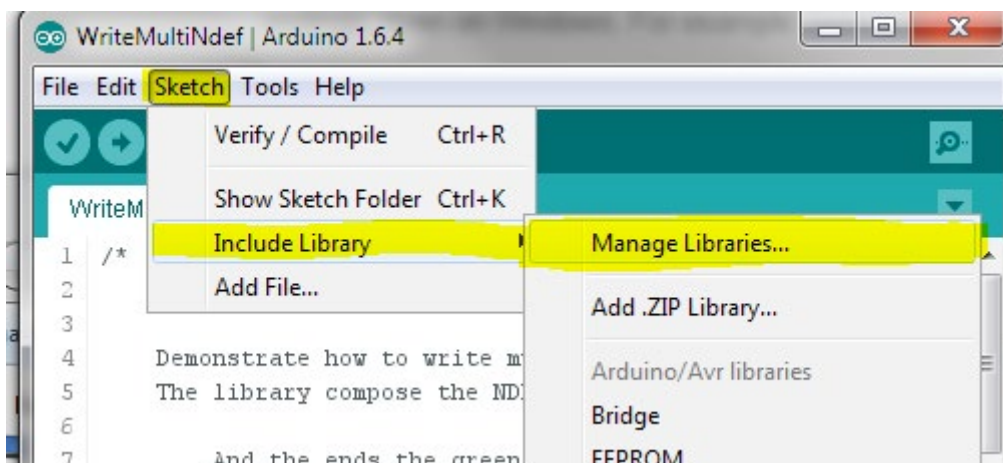


Figure 10: ArduinoIDE: Launch Manage Libraries interface

In order to include the Smart Everything library use the menu Sketch à Include Library à Manage Libraries...

Once the Library Manager is started, you can filter the available library writing SmartEverything in the right top text box end then you can select the necessary library.

An example on how they will be available to the user are showed by the library “SmartEverything SIGFOX LE51-868” (Library for SIGFOX communication)

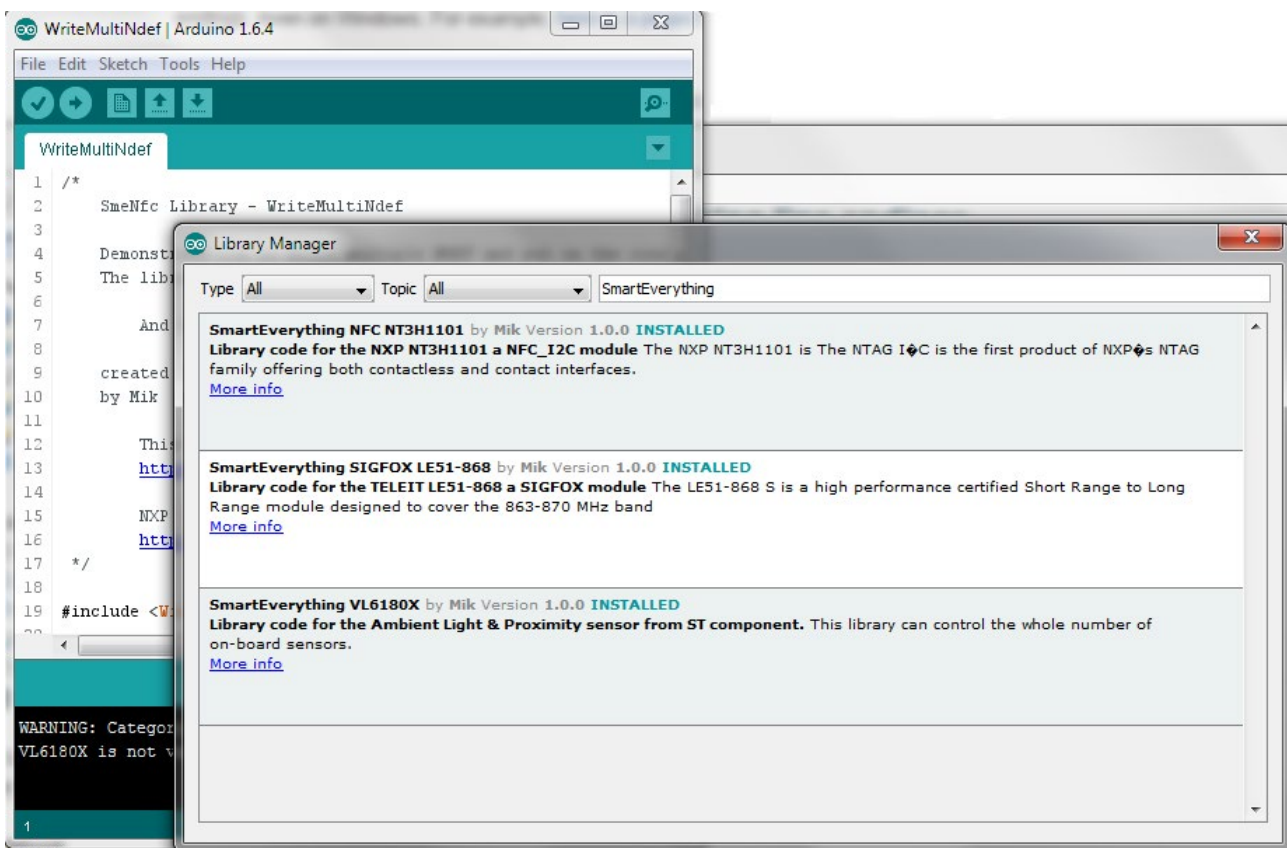


Figure 11: ArduinoIDE: Library Manager

#### 5.4. Importing the Smart Everything Board Library

The BIB board provides some libraries useful to interface with some of its main components.

In order to include the Smart Everything library use the menu Sketch à Include Library à Manage Libraries.

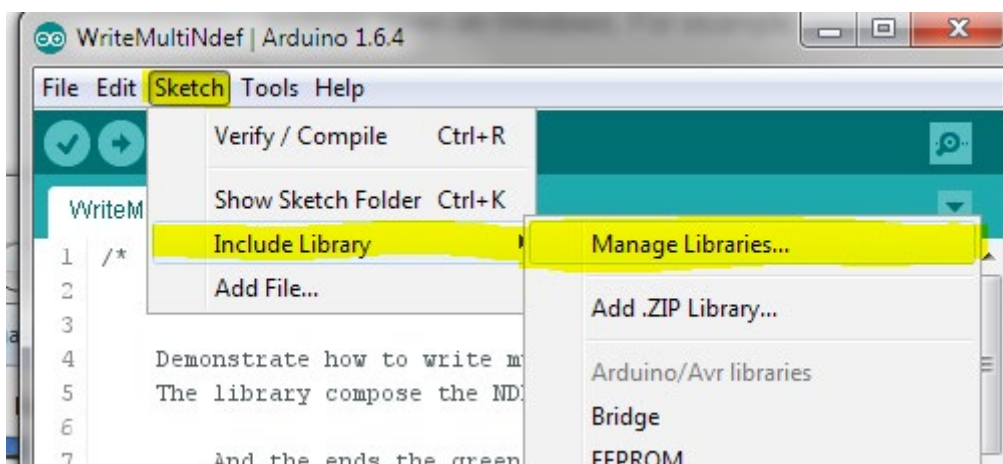


Figure 12: ArduinoIDE: Launch Manage Libraries interface

In order to include the Smart Everything library use the menu Sketch à Include Library à Manage Libraries...

Once the Library Manager is started, you can filter the available library writing SmartEverything in the right top text box end then you can select the necessary library.

An example on how they will be available to the user are showed by the library “SmartEverything SIGFOX LE51-868” (Library for SIGFOX communication)<sup>1</sup>

### 5.5. Example Sigfox mode EU

/\*

SmeloT Library - DataModeEu

Demonstrate how move in data mode for the Telit Sigfox component.

The Library sends an “Hello” String over the SigFox network.

The RGB Green Led lighth on when the Telit sends the message on the air,

lighth off at the positive answer from the component.

<sup>1</sup> The other libraries present in the library manager do not apply to the BIB board, as they require external modules

created 05 May 2015

by Mik (smkk@axelelectronica.it)

This example is in the public domain

<https://github.com/ameltech>

Telit le51-868-s more information available here:

<http://www.telit.com/products/product-service-selector/product-service-selector/show/product/le51-868-s/>

\*/

```
#include <Wire.h>
```

```
#include <SmeSFX.h>
```

```
#include <Arduino.h>
```

```
char helloMsg[5]= {'H','e', 'l', 'l', 'o'};
```

```
bool messageSent;
```

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {
```

```
  SerialUSB.begin(115200);
```

```
  sfxAntenna.begin();
```



```
int initFinish=1;

while (!SerialUSB) {

;

}

SerialUSB.println("sending Hello over the network");

// send Hello on the air

sfxAntenna.sfxSendData(helloMsg, strlen((char*)helloMsg));

}

// the loop function runs over and over again forever

void loop() {

bool answerReady = sfxAntenna.hasSfxAnswer();

if (answerReady) {

if (sfxAntenna.getSfxMode() == sfxDataMode) {

switch (sfxAntenna.sfxDataAcknowledge()) {

case SFX_DATA_ACK_START:

SerialUSB.println("Waiting Answer");

break;

case SFX_DATA_ACK_PROCESSING:

SerialUSB.print('.');
```

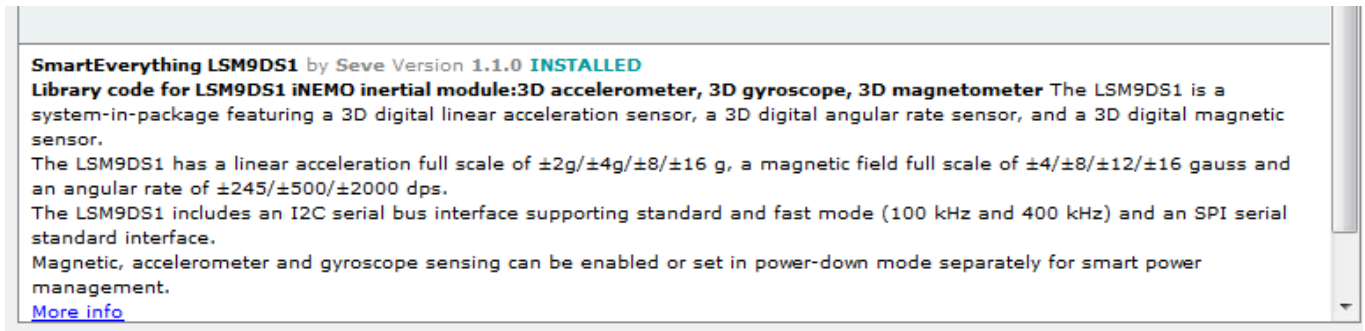
```
break;

case SFX_DATA_ACK_OK:
#ifdef ASME3_REVISION
    ledGreenLight(HIGH);
#endif
    SerialUSB.println(' ');
    SerialUSB.println("Answer OK :) :) :)");
    break;

case SFX_DATA_ACK_KO:
#ifdef ASME3_REVISION
    ledRedLight(HIGH);
#endif
    SerialUSB.println(' ');
    SerialUSB.println("Answer KO :( :( :( (");
    break;
}
}
}
}
```

## 5.6. Example Sigfox manage GPS

Select from “Type” combo Partner and choose the LSM9DS1 in order to install the core of the SmartEverything.



Upload the new firmware from My Account of the BIB site .

## Chapter 6: References and Useful Links

### 6.1. Data sheets

- [ATMEL SAMD21 Ultra low-power ARM® Cortex®-M0+ MCU](#)
- [Dynaflex 868Mhz Antenna](#)
- [Sigfox Module \(Telit LE51-868 S\)](#)
- [GPS Module with Embedded Antenna \(Telit Jupiter SE868-A\)](#)
- [Axis sensors \(ST LSM9DS1\)](#)

### 6.2. Tools

- [Arduino IDE](#)
- [Atmel Studio](#)

### 6.3. Web Sites

- [Axel Elettronica – www.axelelettronica.it](#)
- [Arduino – www.arduino.cc](#)
- [Atmel – www.atmel.com](#)

## Chapter 7: Troubleshooting

### 7.1. Driver installation problems

It can happen the installation of the driver does not automatically complete in a successful way. The driver to manage the COM Port are not properly installed and the device is reported as an Unknown Device.

If this happen, it is necessary to install manually the driver. The driver can be found in the driver folder of the Arduino IDE folder tree: Example C:\Program Files (x86)\Arduino\drivers

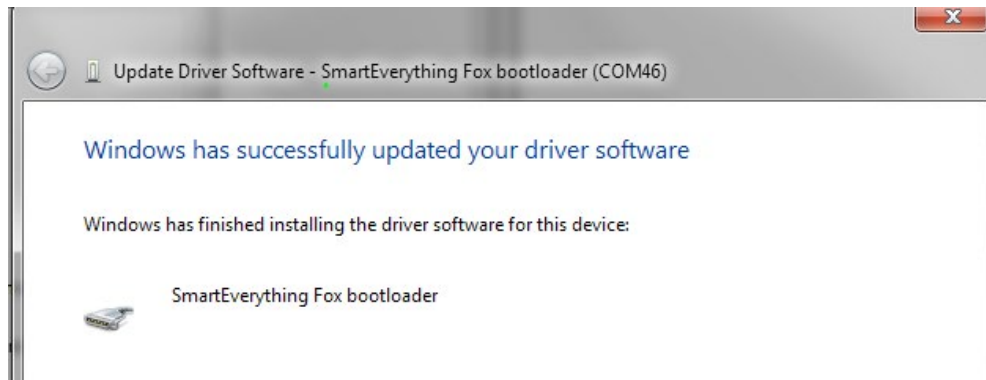


Figure 13: Correct USB Driver installation

### 7.2 Block in bootloader mode

If your BIB not respond to stimulation (perhaps due to a last downloaded code incorrect) or you can not more to download a new application because even USB is impacted, no problem exist this recovery process that leaves the Asme in bootloader mode, so you found the COM port (a USB different number) and look re-download a new load.

“Double-press the reset button”, the procedure executes this:

- 1) First reset, reset the card that starts with bootloader
- 2) With the second reset, the bootloader doesn't load the program into memory but remains waiting on the USB.

**Basic IoT Board**  
User Guide

Ver 2.0

Copyright 2016  
Nettrotter

[www.nettrotter.io](http://www.nettrotter.io)  
[www.axeletttronica.it](http://www.axeletttronica.it)