Copyright Notice and Disclaimer

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

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

Axel Elettronica does not assume any liability arising out of the application or use of any of its products or designs. Products designed or distributed by Axel Elettronica 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

<table>
<thead>
<tr>
<th>Release No.</th>
<th>Date</th>
<th>Revision Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ver 1.0</td>
<td>14/12/2016</td>
<td>First Release</td>
</tr>
<tr>
<td>Ver 1.1</td>
<td>14/01/2017</td>
<td>PDF error conversion fixed &amp; General review</td>
</tr>
<tr>
<td>Ver 1.2</td>
<td>31/01/2017</td>
<td>Fig. 2 and Fig. 5 changed</td>
</tr>
<tr>
<td>Ver1.3</td>
<td>11/02/2017</td>
<td>MCU pin column added to table 4; Tables 5, 6, 7 added</td>
</tr>
</tbody>
</table>

Table 1: Document History
# TABLE OF CONTENTS

Chapter 1  Introduction .................................................................................................................. 5
  1.1 Board Specifications .................................................................................................................. 5

Chapter 2  Hardware ......................................................................................................................... 6
  2.1 External View ............................................................................................................................ 6
  2.2 ASME Lion Block Diagram ......................................................................................................... 6
  2.3 ATMEL SAMD21 Ultra low-power ARM® Cortex®-M0+ MCU .................................................. 7
  2.4 Power Supply Circuitry .............................................................................................................. 7
  2.5 Microchip LoRa Module (RN2483) .......................................................................................... 8
  2.6 Crypto Authentication Chip (ATMEL ATECC508A) .................................................................... 9
  2.7 GPS Module with Embedded Antenna (Telit Jupiter SE868-A) .................................................. 10
  2.8 Bluetooth Low Energy (Microchip RN4871) ............................................................................. 10
  2.9 32Kx8 Bits EEPROM (AT24C256C) ....................................................................................... 11
  2.10 Board Interfaces and Connector .............................................................................................. 12

Chapter 3  Mechanical Information .................................................................................................. 14
  3.1 Main components layout .......................................................................................................... 14
  3.2 Mechanical Characteristics ...................................................................................................... 14

Chapter 4  Software Development .................................................................................................. 15

Chapter 5  Getting Started with Arduino IDE and Sketch Projects .................................................. 16
  5.1 Tools ......................................................................................................................................... 16
  5.2 Setup the Environment ............................................................................................................ 16
  5.3 Compile the Example project .................................................................................................. 18
  5.4 Run the software .................................................................................................................... 18
  5.5 Importing the Lora SmartEverything Board Library ............................................................... 19
  5.6 Official GitHub library repository .......................................................................................... 21

Chapter 6  References and Useful Links .......................................................................................... 22
  6.1 Data sheets ............................................................................................................................... 22
  6.2 Tools ......................................................................................................................................... 22
  6.3 Web Sites ................................................................................................................................ 22

Chapter 7  Troubleshooting .............................................................................................................. 23
  7.1 Driver installation problems .................................................................................................... 23
Tables Index
Table 1: Document History ........................................................................................................................................2
Table 2: Board Specifications .....................................................................................................................................5
Table 3: Power Supply Connectors and ranges .........................................................................................................8
Table 4: Arduino Compatible Headers ......................................................................................................................12
Table 5 BLE port description ......................................................................................................................................13
Table 6 LORA port description ...................................................................................................................................13
Table 7 GPS/GNSS port description ........................................................................................................................13

Illustrations Index
Figure 1: Front View .....................................................................................................................................................6
Figure 2: SmartEverything Block Diagram ..................................................................................................................6
Figure 3: Power Supply block diagram ........................................................................................................................8
Figure 4: Headers pinout .................................................................................................................................................12
Figure 5: Main Component layout (top) .........................................................................................................................14
Figure 6: Dimensions ....................................................................................................................................................14
Figure 7: Launch Board Manager ................................................................................................................................16
Figure 8: ArduinoZero Core Installation ......................................................................................................................17
Figure 9: SmartEverything Core Installation ...............................................................................................................17
Figure 10: ArduinoIDE: Verify the code .........................................................................................................................18
Figure 11: ArduinoIDE: Upload the code ........................................................................................................................19
Figure 12: ArduinoIDE: Launch Manage Libraries interface ..........................................................................................19
Figure 13: ArduinoIDE: Library Manager ......................................................................................................................20
Figure 14 Correct USB Driver installation ..................................................................................................................23
Chapter 1 Introduction

This document describes ASME Lion SoM (System On Module) based on Atmel D21 Ultra low-power microcontroller using the 32-bit ARM® Cortex®-M0+ processor.

The ASME Lion Board provides the following peripherals or modules:

- Crypto Authentication chipset
- LoRa Module
- GPS Module with Embedded Antenna
- Bluetooth Low Energy (BLE) Interface

The ASME Lion Board is supported by the Arduino IDE for a fast and easy software development cycle (https://www.arduino.cc/en/Main/Software).

The software can also be developed using the Atmel Studio IDE commonly preferred by professional software engineers (http://www.atmel.com/Microsite/atmel-studio/).

### 1.1 Board Specifications

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Clock speed</td>
<td>48MHz</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>256K</td>
</tr>
<tr>
<td>SRAM</td>
<td>32K</td>
</tr>
<tr>
<td>Connector</td>
<td>1 Power Jack</td>
</tr>
<tr>
<td></td>
<td>1 USB</td>
</tr>
<tr>
<td></td>
<td>1 Lora antenna</td>
</tr>
<tr>
<td></td>
<td>1 Li-Po Battery</td>
</tr>
<tr>
<td></td>
<td>1 SWD Debugger</td>
</tr>
<tr>
<td></td>
<td>Arduino compatible PinOut</td>
</tr>
<tr>
<td>Board supply voltage</td>
<td>6V – 24V CC</td>
</tr>
<tr>
<td>Operative Voltage</td>
<td>3.3V</td>
</tr>
<tr>
<td>Temperature</td>
<td>-40°C to +85°C</td>
</tr>
<tr>
<td>Dimensions</td>
<td>68x30 mm</td>
</tr>
<tr>
<td>RoHS status</td>
<td>Compliant</td>
</tr>
</tbody>
</table>

Table 2: Board Specifications
Chapter 2 Hardware

2.1 External View

Figure 1: Front View

2.2 ASME Lion Block Diagram

Figure 2: SmartEverything Block Diagram
2.3 ATMEL SAMD21 Ultra low-power ARM® Cortex®-M0+ MCU

Atmel SAMD21 Ultra low-power microcontroller using the 32-bit ARM® Cortex®-M0+ processor is the core of the entire board.

**Memory**
- 256KB in-system self-programmable Flash
- 8KB Flash Read-While-Write section
- 32KB SRAM Main Memory
- 8KB SRAM Low power Memory

**Clock Frequency**
- 32.768kHz crystal oscillator (XOSC32K)

**Arduino compliance**
- Digital i/o Pins : 20
- PWM Pins: all but 2 and 7
- USART: 2 (USB and TTL Digital)
- Analog Input Pins: 6, 12-bit DAC
- Analog Output Pins: 1, 10-bit DAC
- External Interrupts: All pins except pin 4

More information can be found on the ATMEL SAMD21 Data Sheet (See the link in Chapter 6)

2.4 Power Supply Circuitry

The board can be powered by three different power supply sources:
- External Li-Po Battery
- A 6V to 24V input
- A 5V Mini USB connector

**WARNING:**
Li-Po batteries are charged at 4,2V with a current that is usually half of the nominal capacity (C/2). This board has a specialized chip that has a preset charging current of 350mAh. This means that the MINIMUM capacity of the Li-Po battery shall be 700 mAh. Smaller cells will be damaged by this current and may overheat, develop internal gasses and explode, setting the surroundings on fire.

All the components can work with the voltage range provided by the batteries, with the exception of the GPS module that requires at least 2.8V. For this reason it is recommended to use an appropriate battery pack for a prolonged use of the GPS.
Use Li-Ion battery $V_{nom} = 3.7V$, $V_{chrg} = 4.2V$ Capacity $\geq 700$ mAh

### Table 3: Power Supply Connectors and ranges

<table>
<thead>
<tr>
<th>Name</th>
<th>Nominal</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>J10</td>
<td>3 V</td>
<td>Battery Power Supply</td>
<td>1.4V to 3.2V</td>
</tr>
<tr>
<td>J12</td>
<td>6V to 24V</td>
<td>External Power Supply</td>
<td>6V to 24V</td>
</tr>
<tr>
<td>USB</td>
<td>5V</td>
<td>USB Port</td>
<td>5V</td>
</tr>
</tbody>
</table>

#### 2.5 Microchip LoRa Module (RN2483)

The Microchip RN2483 is a **Low-Power Long Range LoRa® Technology Transceiver Module**, providing a low-power solution for long range wireless data transmission.

The RN2483 module complies with the LoRaWAN Class A protocol specifications. It integrates RF, a baseband controller and a command Application Programming Interface (API) processor. More information can be found on relevant Data Sheet (See the link in Chapter 6)
2.5.1 Main Features
On-board LoRaWAN™ protocol stack:

- ASCII command interface over UART
- Environmentally friendly, RoHS compliant
- European R&TTE Directive Assessed Radio Module

RF/Analog Features

- Low-Power Long-range Transceiver operating in the 433 MHz and 868 MHz frequency bands
- High Receiver Sensitivity: down to -146 dBm
- TX Power: adjustable up to +14 dBm high efficiency PA
- Up to 15km coverage at suburban and up to 5km coverage at urban area

2.6 Crypto Authentication Chip (ATMEL ATECC508A)
The ATECC508A is a member of the Atmel Crypto Authentication™ family of high-security hardware authentication devices. It has a flexible command set that allows to use it in many applications, including the following, among others:

- Anti-counterfeiting
- Protecting Firmware or Media
- Exchanging Session Keys
- Storing Data Securely
- Checking User Passwords

2.6.1 Device Features
The ATECC508A device includes an Electrically Erasable Programmable Read-Only Memory (EEPROM) array that can be used for key storage, miscellaneous read/write data, read-only, secret data, consumption logging, and security configuration.

Access to the various sections of memory can be restricted in a variety of ways, and the configuration can then be locked to prevent changes.

The ATECC508A features a wide array of defense mechanisms specifically designed to prevent physical attacks against the device itself or logical attacks against data transmitted between the device and the system.

Hardware restrictions on the way keys are used or generated provide further defense
against certain styles of attack.

Access to the device is made through a standard I2C interface.

Each ATECC508A ships with a guaranteed unique 9-byte (72-bit) serial number. Using the cryptographic protocols supported by the device, a Host system or remote server can prove that the serial number is authentic and is not a copy. Serial numbers are often stored in a standard Serial EEPROM, which can be easily copied with no way for the Host to know if the serial number is authentic or if it is a clone. The entire serial number must be utilized to guarantee uniqueness.

### 2.6.2 Cryptographic Operation

The ATSHA204A supports a standard challenge-response protocol to simplify programming. In its most basic installation, the Host system sends a challenge (i.e. a number) to the device in the Client, which combines that challenge with a secret key by using the Message Authentication Code (MAC) command from the system and sends that response back to the system.

This basic operation can be expanded in many ways because of the flexible command set of the ATECC508A.

For a complete explanation about the possible Cryptographic Operations check the Data Sheet (See the link in Chapter 6)

### 2.7 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 Chapter 6).

### 2.8 Bluetooth Low Energy (Microchip RN4871)

The RN4871 is a small form factor, Bluetooth 4.2 Low-Energy module measuring just 9 x 11.5 x 2.1 mm. This fully-integrated module is designed for easy implementation into a broad range of applications. Supporting the latest Bluetooth standard, it delivers up to 2.5x throughput improvement and more secure connections vs. Bluetooth 4.1 based products. Developers can easily interface to the device via a standard UART interface, available on most Microcontrollers and Processors.
2.8.1 Characteristics

- FCC Certified: Yes
- Min Temp Range: -20°C
- Max Temp Range: +70°C
- Op Voltage Min: 1.9V
- Op Voltage Max: 3.6V

More information can be found on relevant Data Sheet (See the link in Chapter 6).

The module on the board is equipped with the firmware able to communicate with the ATMEL SAMD21 MCU.


2.9 32Kx8 Bits EEPROM (AT24C256C)

The Atmel® AT24C256C provides 262,144-bits of Serial Electrically Erasable and Programmable Read-Only Memory (EEPROM) organized as 32,768 words of 8 bits each.

The device’s cascading feature allows up to eight devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential.

The device operates from 1.7V to 5.5V.

More information can be found on relevant Data Sheet (See the link in Chapter 6).

2.9.1 Characteristics

- Low-voltage and Standard-voltage Operation–VCC = 1.7V to 5.5V
- 400kHz (1.7V) and 1MHz (2.5V, 2.7V, 5.0V) Compatibility
- Write Protect Pin for Hardware Protection
- 64-byte Page Write Mode–Partial Page Writes Allowed
- High Reliability–Endurance: 1,000,000 Write Cycles–Data Retention: 40 Years

2.9.2 EEPROM Address

The EEPROM is mapped to the following I2C address 0x57
### 2.10 Board Interfaces and Connector

#### Headers

![Figure 4: Headers pinout](image)

<table>
<thead>
<tr>
<th>Conn.</th>
<th>Pin</th>
<th>Description</th>
<th>Arduino pin label</th>
<th>Driven by</th>
</tr>
</thead>
<tbody>
<tr>
<td>J4</td>
<td>1</td>
<td>Digital I/O D8 / PWM</td>
<td>~8</td>
<td>MCU Port PA06</td>
</tr>
<tr>
<td>J4</td>
<td>2</td>
<td>Digital I/O D9 / PWM</td>
<td>~9</td>
<td>MCU Port PA07</td>
</tr>
<tr>
<td>J4</td>
<td>3</td>
<td>Digital I/O D10 / PWM SS</td>
<td>~10</td>
<td>MCU Port PA18</td>
</tr>
<tr>
<td>J4</td>
<td>4</td>
<td>Digital I/O D11 / PWM MOSI</td>
<td>~11</td>
<td>MCU Port PA16</td>
</tr>
<tr>
<td>J4</td>
<td>5</td>
<td>Digital I/O D12 / PWM MISO</td>
<td>~12</td>
<td>MCU Port PA19</td>
</tr>
<tr>
<td>J4</td>
<td>6</td>
<td>Digital I/O D13 / PWM SCK</td>
<td>~13</td>
<td>MCU Port PA17</td>
</tr>
<tr>
<td>J4</td>
<td>7</td>
<td>Ground pin</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>8</td>
<td>Analogue Reference (used by ADC)</td>
<td>AREF</td>
<td>MCU Port PA03</td>
</tr>
<tr>
<td>J4</td>
<td>9</td>
<td>I2C Interface SDA</td>
<td>SDA</td>
<td>MCU Port PA22</td>
</tr>
<tr>
<td>J4</td>
<td>10</td>
<td>I2C Interface SCL</td>
<td>SCL</td>
<td>MCU Port PA23</td>
</tr>
<tr>
<td>J5</td>
<td>1</td>
<td>Digital I/O ATN</td>
<td>ATN</td>
<td>MCU Port PB06</td>
</tr>
<tr>
<td>J5</td>
<td>2</td>
<td>IOREF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>3</td>
<td>Reset</td>
<td>RESET</td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>4</td>
<td>3.3V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>5</td>
<td>5V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>6</td>
<td>Ground pin</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>7</td>
<td>Ground pin</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>8</td>
<td>External Power Supply Input</td>
<td>VIN</td>
<td></td>
</tr>
<tr>
<td>J6</td>
<td>1</td>
<td>Analog I/O A0</td>
<td>A0</td>
<td>MCU Port PA02</td>
</tr>
<tr>
<td>J6</td>
<td>2</td>
<td>Analog I/O A1</td>
<td>A1</td>
<td>MCU Port PB08</td>
</tr>
<tr>
<td>J6</td>
<td>3</td>
<td>Analog I/O A2</td>
<td>A2</td>
<td>MCU Port PB09</td>
</tr>
<tr>
<td>J6</td>
<td>4</td>
<td>Analog I/O A3</td>
<td>A3</td>
<td>MCU Port PA04</td>
</tr>
<tr>
<td>J6</td>
<td>5</td>
<td>Analog I/O A4</td>
<td>A4</td>
<td>MCU Port PA05</td>
</tr>
<tr>
<td>J6</td>
<td>6</td>
<td>Analog I/O A5</td>
<td>A5</td>
<td>MCU Port PB02</td>
</tr>
<tr>
<td>J7</td>
<td>1</td>
<td>Digital I/O D0 / Serial1 (RX)</td>
<td>RX&lt;0</td>
<td>MCU Port PA11</td>
</tr>
<tr>
<td>J7</td>
<td>2</td>
<td>Digital I/O D1 / Serial1 (TX)</td>
<td>TX&gt;1</td>
<td>MCU Port PA10</td>
</tr>
<tr>
<td>J7</td>
<td>3</td>
<td>Digital I/O D2 / PWM</td>
<td>~2</td>
<td>MCU Port PA14</td>
</tr>
<tr>
<td>J7</td>
<td>4</td>
<td>Digital I/O D3 / PWM</td>
<td>~3</td>
<td>MCU Port PA09</td>
</tr>
<tr>
<td>J7</td>
<td>5</td>
<td>Digital I/O D4 / PWM</td>
<td>~4</td>
<td>MCU Port PA08</td>
</tr>
<tr>
<td>J7</td>
<td>6</td>
<td>Digital I/O D5 / PWM</td>
<td>~5</td>
<td>MCU Port PA15</td>
</tr>
<tr>
<td>J7</td>
<td>7</td>
<td>Digital I/O D6 / PWM</td>
<td>~6</td>
<td>MCU Port PA20</td>
</tr>
<tr>
<td>J7</td>
<td>8</td>
<td>Digital I/O D7 / PWM</td>
<td>~7</td>
<td>MCU Port PA21</td>
</tr>
</tbody>
</table>

**Table 4: Arduino Compatible Headers**
<table>
<thead>
<tr>
<th>Description</th>
<th>Driven by</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLE TXD</td>
<td>MCU Port PB16</td>
</tr>
<tr>
<td>BLE RXD</td>
<td>MCU Port PB17</td>
</tr>
<tr>
<td>BLE RTS</td>
<td>MCU Port PB22</td>
</tr>
<tr>
<td>BLE CTS</td>
<td>MCU Port PB23</td>
</tr>
<tr>
<td>BLE P10</td>
<td>I/O Expander GPIO4</td>
</tr>
<tr>
<td>BLE P11</td>
<td>I/O Expander GPIO5</td>
</tr>
<tr>
<td>BLE P32</td>
<td>I/O Expander GPIO6</td>
</tr>
<tr>
<td>BLE P33</td>
<td>I/O Expander GPIO7</td>
</tr>
<tr>
<td>BLE RST</td>
<td>I/O Expander GPIO1</td>
</tr>
</tbody>
</table>

Table 5 BLE port description

<table>
<thead>
<tr>
<th>Description</th>
<th>Driven by</th>
</tr>
</thead>
<tbody>
<tr>
<td>LORA TXD</td>
<td>MCU Port PB12</td>
</tr>
<tr>
<td>LORA RXD</td>
<td>MCU Port PB13</td>
</tr>
<tr>
<td>LORA RTS</td>
<td>MCU Port PB14</td>
</tr>
<tr>
<td>LORA CTS</td>
<td>MCU Port PB15</td>
</tr>
<tr>
<td>LORA GPIO0</td>
<td>MCU Port PB07</td>
</tr>
<tr>
<td>LORA GPIO1</td>
<td>MCU Port PB10</td>
</tr>
<tr>
<td>LORA GPIO2</td>
<td>MCU Port PB11</td>
</tr>
<tr>
<td>LORA RST</td>
<td>I/O Expander GPIO0</td>
</tr>
</tbody>
</table>

Table 6 LORA port description

<table>
<thead>
<tr>
<th>Description</th>
<th>Driven by</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS TXD</td>
<td>MCU Port PA12</td>
</tr>
<tr>
<td>GPS RXD</td>
<td>MCU Port PA13</td>
</tr>
<tr>
<td>GPS RST</td>
<td>I/O Expander GPIO2</td>
</tr>
<tr>
<td>GPS FORCE ON</td>
<td>I/O Expander GPIO3</td>
</tr>
</tbody>
</table>

Table 7 GPS/GNSS port description
Chapter 3 Mechanical Information

3.1 Main components layout

Figure 5: Main Component layout (top)

3.2 Mechanical Characteristics

The maximum length and width of the Smarteverything Lion PCB are 2.7 and 2.1 inches respectively, with the USB connectors, power jack 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 SmartEverything Lion fully compatible with most shields designed for the Uno, Diecimila or Duemilanove.

Figure 6: Dimensions
Chapter 4 Software Development

The following chapters provide an overview about how users can develop their software and run it on the ASME Lion 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, having some tools to easily debug the code and fix what does not work as expected is really important.

The possibility to use an external debugger like the Atmel-ICE (http://www.atmel.com/tools/atmel-ice.aspx) is 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 link in Chapter 6 )
The first time you run the ArduinoIDE it is necessary to load the Arduino Zero &
SmartEverything Core.
Click on the Tools → Boards → Boards Manager… menu entry

![Launch Board Manager](image_url)

*Figure 7: Launch Board Manager*
The two steps of installation shall be done as follows:

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

![ArduinoZero Core Installation](image1)

2) Select from “Type” combo Partner and choose the ARROW Boards in order to install the core of the SmartEverything.

![SmartEverything Core Installation](image2)

The installation procedure will automatically install all the necessary USB drives. Once the installation is completed, you will be able to connect the Board and start coding. If the Driver installation process does not complete successfully, follow the steps described in Chapter 7.
### 5.3 Compile the Example project

This procedure explains how to compile and download the code on the SmartEverything Lion board.

As every other Arduino board, the SmartEverything Lion has the basic examples code. To try those examples, just go to File->Examples and select one under the 01.Basics menu.

![Image showing Arduino IDE: Examples menu](image)

**Figure 10:** ArduinoIDE: Verify the code

### 5.4 Run the software

Verify the code

![Image showing Arduino IDE: Verify](image)

Load the software on the Connect the SmartEverything Lion board.
5.5 Importing the Lora SmartEverything Board Library

The SmartEverything Lion board will provide, a few days after its release on the market, some useful libraries to interface with some of its main components.

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

Once the Library Manager is started, you can filter the available libraries writing SmartEverything, or the name of the component, in the right top text box and then you can select the necessary library.
The procedure to download the other SmartEverything Lion libraries is identical.

![ArduinoIDE: Library Manager](image)

*Figure 13: ArduinoIDE: Library Manager*

Another way to install the Library of the supported components is going to the GitHub repository and downloading the zip file.

![GitHub repository](image)

Once it is downloaded, go to “include Library” menu and use the “Add .ZIP Library…” item.
5.6 Official GitHub library repository

Here is the official repository where the zip file to download can be found:

- LoRa Library (https://github.com/axelelettronica/sme-rn2483-library)
- GPS Library (https://github.com/ameltech/sme-se868-a-library)
- BLE Library (https://github.com/axelelettronica/sme-rn4870-library)
- EEPROM Library (https://github.com/PaoloP74/EEPROM)

The EEPROM shall be initialized with these parameters

```c
new extEEPROM(kbits_256, 1, 64, 0x57);
```
Chapter 6 References and Useful Links

6.1 Data sheets

- **ATMEL SAMD21 Ultra low-power ARM® Cortex®-M0+ MCU**

- **Crypto Authentication Chip (ATMEL ATECC508A)**

- Dynaflex 868Mhz Antenna (915/2)

- **LoRa Module (Microchip RN2483)**

- **GPS Module with Embedded Antenna (Telit Jupiter SE868-A)**

- **BLE Microchip RN4871(specification)**

- **E2Prom ATMEL (AT24C256C)**

6.2 Tools

- Arduino IDE

- Atmel Studio

6.3 Web Sites

- Axel Elettronica – [www.axelelettronica.it](http://www.axelelettronica.it)

- Arduino – [www.arduino.cc](http://www.arduino.cc)

- Atmel – [www.atmel.com](http://www.atmel.com)


Chapter 7 Troubleshooting

7.1 Driver installation problems

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

If this happens, it is necessary to install the drive manually.

The correct drivers can be found on the following directory:
C:\Users\<login name>\AppData\Local\Arduino15\packages\Arrow\hardware\samd\<installed Rel>\drivers.

![Figure 14 Correct USB Driver installation](image)