SENTIMATE board: getting started and FW examples

Document Revision

REVISION	DATE	AUTHOR	DESCRIPRION
1	10/06/2019	Ing. M.Gualdi	First draft based on schematic V3
2	27/06/2019	Ing. M.Gualdi	Added chapter 2
3	09/07/2019	Ing. L. Moioli	English version
4	23/07/2019	Ing. L. Moioli	Added example 13
5	29/07/2019	Ing. M.Gualdi	Updated board images
6	13/12/2019	Ing. E.Limonta	Added information on the antenna

Index

1. SYSTEM CONFIGURATION

SENTIMATE board is equipped with the following macro elements:

.1.1 AmbiMate multi-sensor module 2314291-2 (TE)

At the following link you can find the description of the module developed by TE.

.1.2 Radio module ZGM130S037HGN1 (Silicon Labs)

At the following link you can find the description of the module developed by Silicon Labs.

- Transmit frequencies: 908.4 MHz - 916.0 MHz
- Maximum TX power 4 dBm

.1.3 Radio module MGM13P12F512GA (Silicon Labs)

At the following link you can find the description of the module developed by Silicon Labs.

- Transmit frequencies: 2400 MHz – 2483.5 MHz
- Maximum TX power 19 dBm

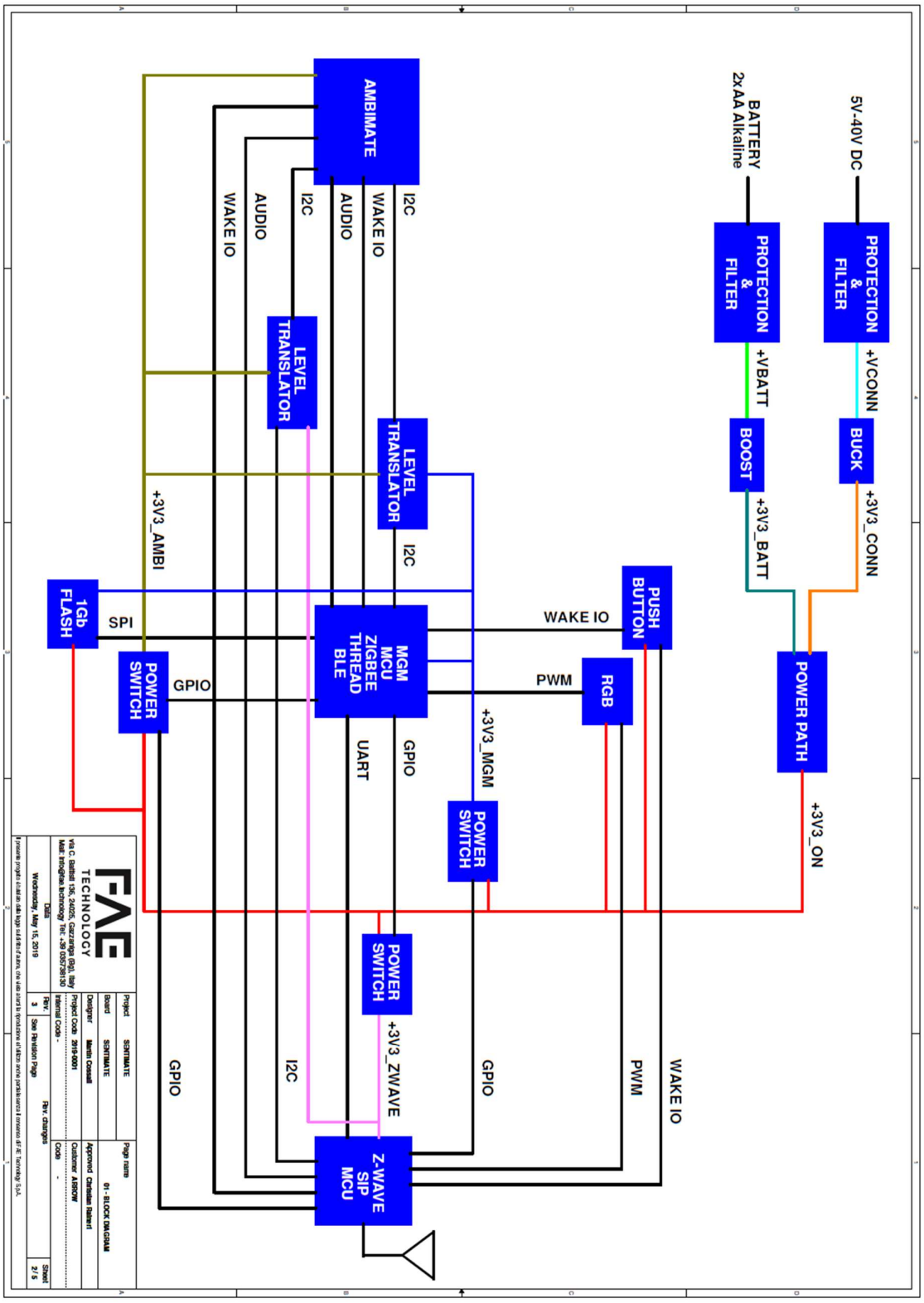
.1.4 RGB LED

At the following link you can find the description of the LED.

.1.5 Antenna

At the following link you can find the description of the antenna .

- Transmit frequencies: 908.4 MHz - 916.0 MHz
- Peak gain: 1.4 dBi
- The ZGM130S and MGM13P modules can transmit simultaneously


5V-40V DC
PROTECTION & FILTER
+VCONN
BUCK
+3V3_CONN
BATTERY 2xAA Alkaline
PROTECTION & FILTER
+VBATT
BOOST
+3V3_BATT
POWER PATH
+3V3_ON
AMBIMATE
I2C
WAKE IO
AUDIO
LEVEL TRANSLATOR
+3V3_AMBI
1Gb FLASH
SPI
POWER SWITCH
GPIO
MGM MCU ZIGBEE THREAD BLE
PUSH BUTTON
WAKE IO
RGB
PWM
+3V3_MGM
POWER SWITCH
GPIO
UART
POWER SWITCH
+3V3_ZWAVE
Z-WAVE SIP MCU
WAKE IO
PWM
GPIO
I2C
GPIO
TECHNOLOGY
Project SSNTMATE
Board SSNTMATE
Designer Martin Cossutt
Project Code 2019-0001
Internal Code -
Rev. 3
Date Wednesday, May 15, 2019
See Revision Page
Rev. changes
Page name 01- BLOCK DIAGRAM
Approved Christian Palmeri
Customer ARROW
Code -
Sheet 2/ 5

.2 DEVELOPMENT ENVIRONMENT SETUP

In order to use the examples, the following tools are required:

- SENTIMATE Board
- Power supply access
- Segger J-Link (any version) or compatible and JTAG 10-Pole cable
- Simplicity Studio V4

.2.1 Sentimate Board Setup

In order to program the SENTIMATE board, two JTAG connectors compliant with 10-pole standard are available. The first one is dedicated to programming the M13P12F512GA module, while the second one is dedicated to programming the ZGM130S module. To access these connectors, it is necessary to remove the upper cover of the board.
Below is a picture of the board with the two JTAG connectors highlighted in red.

.2.2 Power supply Setup

The board can be powered in two different ways:

- External DC power supply (from 5 to 40Vdc on external 2 poles connector)
- Power supply through two AA alkaline type batteries

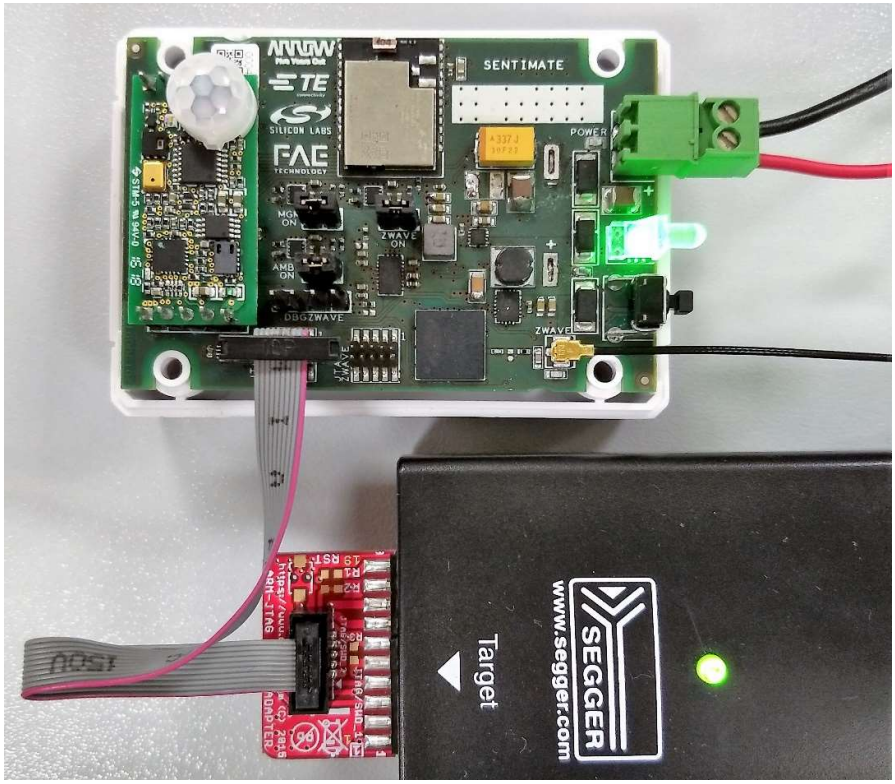
During development phase, it is recommended to use external power supply, to avoid limitation due to battery duration. For example, you can use a power adapter with standard output of 5V, 12V, 19V o 24V, or you can use a power supply with adjustable voltage.
The electric current required by the board depends on the provided supply voltage, in the worst case of 5V supply, it is necessary to provide a maximum current of 500 [mA].
In order to work properly, the board requires that supply jumpers are closed or that instead of a jumper, an ammeter is inserted to monitor power consumption.
The following picture shows two closed jumpers that provide power supply to the AMBIMATE and MGM13P12F512GA modules, while ZGM130S module is not enabled.

.2.3 **Debugger Setup**

It is recommended to use the J-Link debugger with the latest software version installed, downloadable at this link.
It is recommended to use the software package starting from version V6.46g, in which both devices MGM13P12F512GA and ZGM130S are fully supported.

If you are in possession of a Silicon Labs wireless development kit, it is possible to use BRD4001A boards configured for debugging external devices.

Below is a picture that shows the connection between the J-Link and the board through one of the JTAG ports.



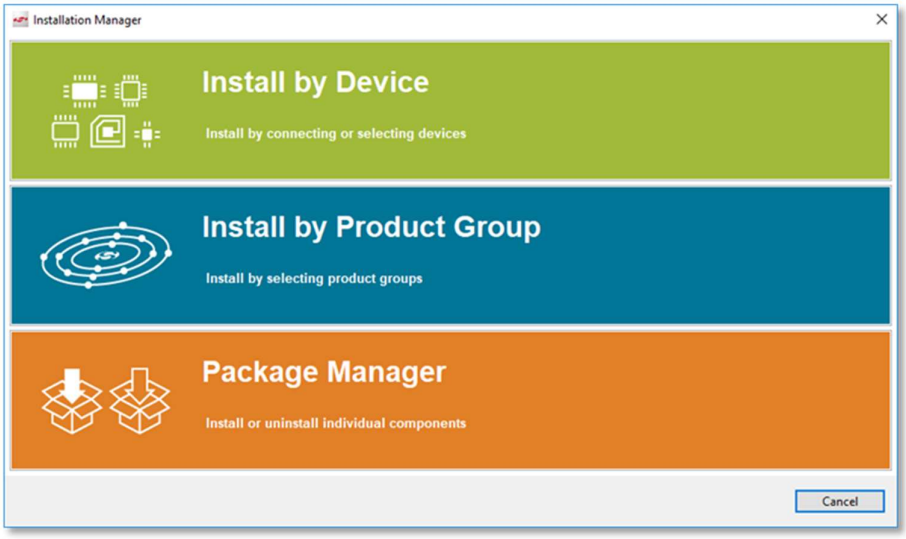
.2.4 IDE Setup

The development environment used is Simplicity Studio V4, provided by Silicon Labs and downloadable from this link.

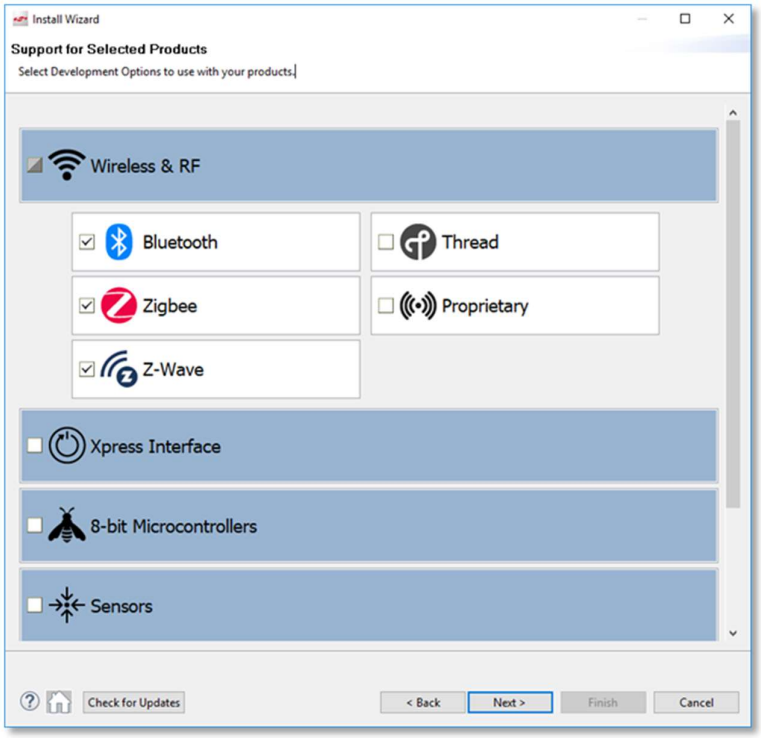
Once you have downloaded the installation file suited for your platform, start it and follow the installation wizard.

Silicon Labs provides AN0822 document on how to use the development environment.

Once the installation phase is completed, you have to install the SDK libraries related to the radio modules mounted on the SENTIMATE board.
To do this, start Simplicity Studio and once the IDE has loaded, use the “Installation manager” tool to select the additional components to install.



In this screen select “Install by Product Group” and go to the next screen.



Here, select the needed radio components and complete the installation.

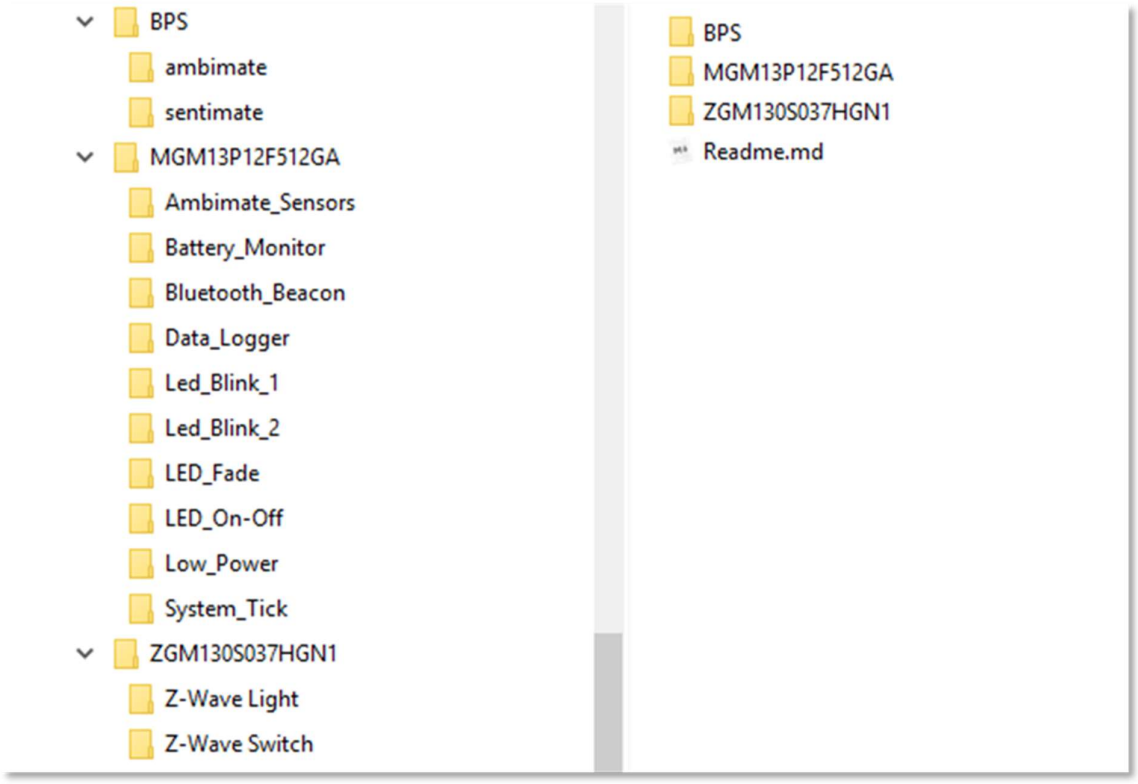
Important Note: *in order to download and access all the functionalities of the Zigbee and Z-Wave libraries it is necessary to register a Silicon Labs development* kit.

.2.5 Sentimate Examples

The first thing to do is downloading the archive containing the examples from the ARROW site and unzip it in any position on the disk.

.2.5.1 Folders structure

Example folders are organized as in the following picture:

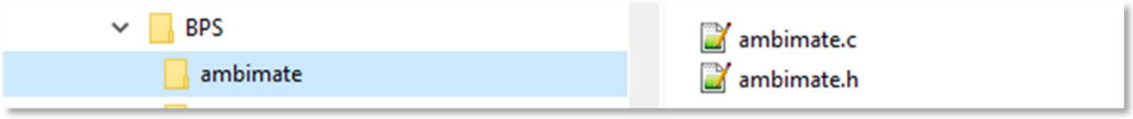


.2.5.2 BSP

Inside this folder, there are the drivers to manage the hardware components mounted on the board.

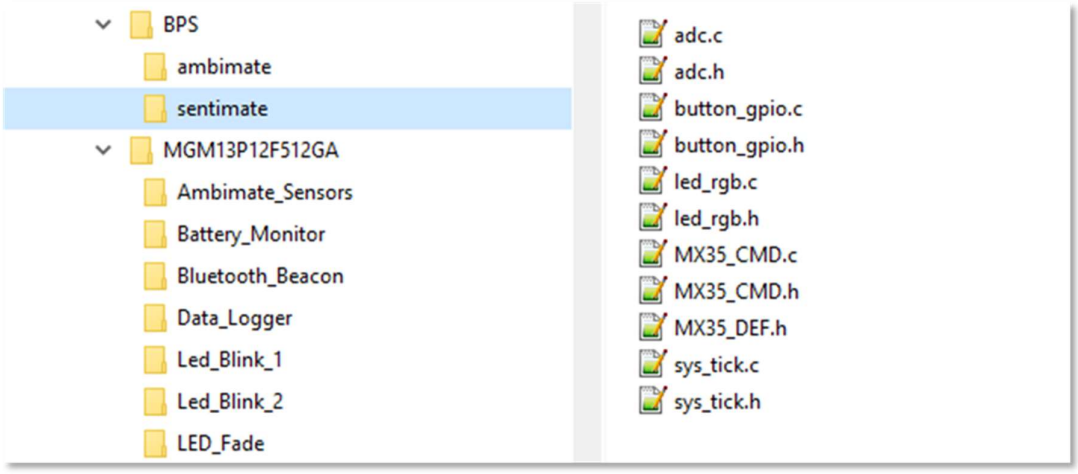
.2.5.2.1 Ambimate

These drivers are dedicated to interfacing with the AMBIMATE module.



.2.5.2.2 SENTIMATE

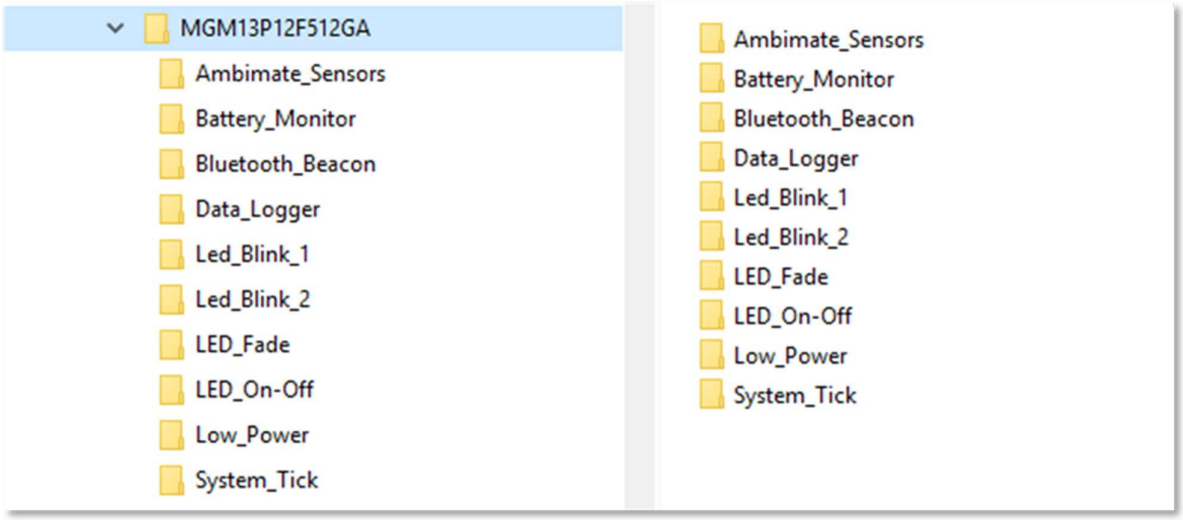
These drivers help interfacing the microcontroller with the remaining peripherals available on the board.



NOTE: *These drivers were developed for the MGM13P12F512GA radio module, but are easily adapted for the ZGM130S module as well.*

.2.5.3 **MGM13P12F512GA**

Inside this folder there are the various examples to use with the MGM13P12F512GA module.



In table 1, of chapter 4, it is reported the example list with a brief description concerning the components involved in each example.

.2.5.4 **ZGM130S**

Inside this folder there are the various examples to use with the ZGM130S module.

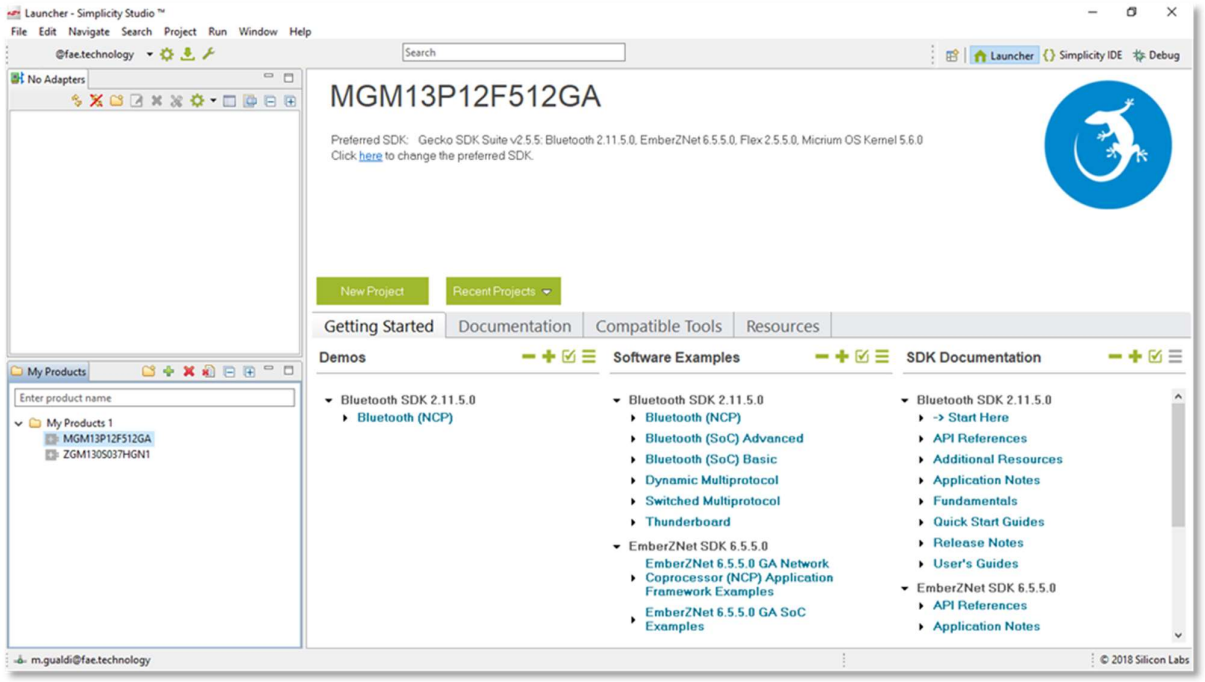
In table 2, of chapter 4, it is reported the example list with a brief description concerning the components involved in each example.

.2.6 How to import the examples

To import the examples into Simplicity Studio, it is sufficient to follow the next steps:

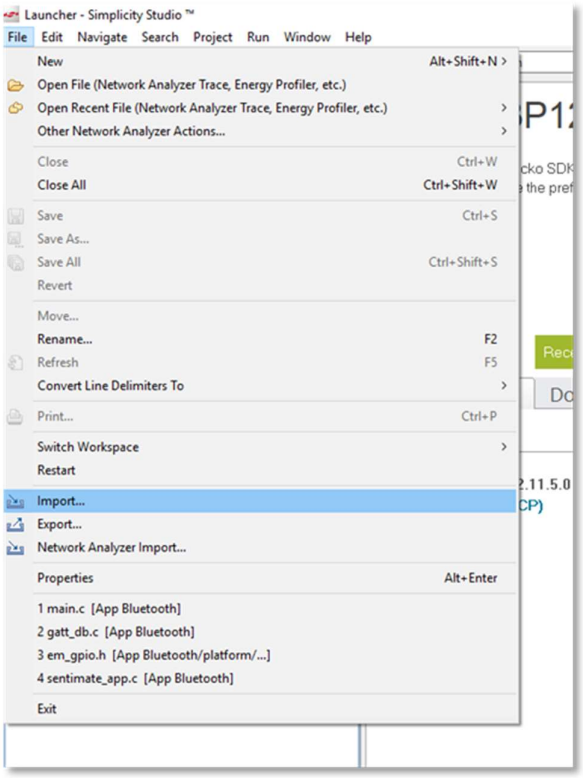
.2.6.1 Start the IDE

Start Simplicity Studio and wait while loading until the “Launcher” is ready



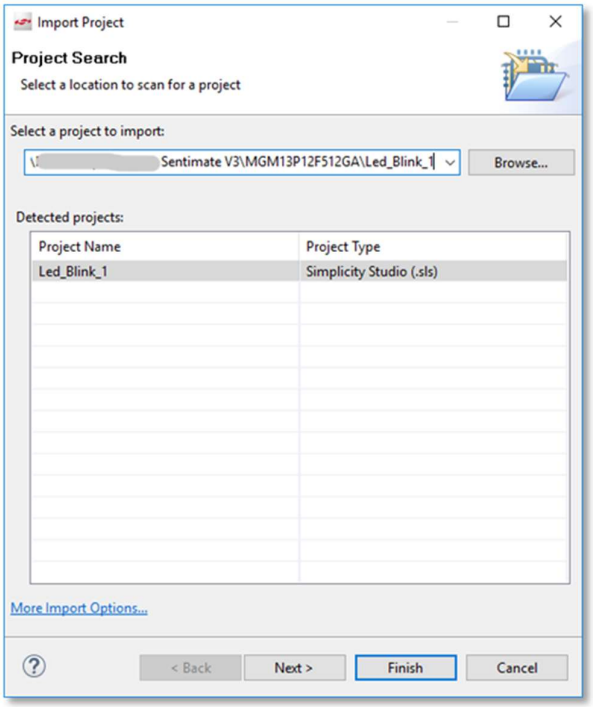
.2.6.1.1 Import Project

From the drop-down menu “File” select the item “Import…”



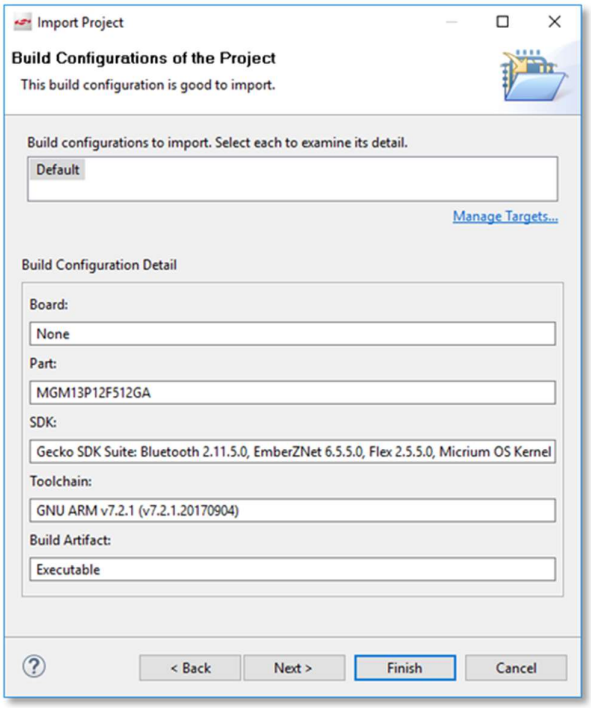
.2.6.1.2 Project selection

With the button “Browse…” go to the location where you unzipped the examples and select the folder containing the example you want to import



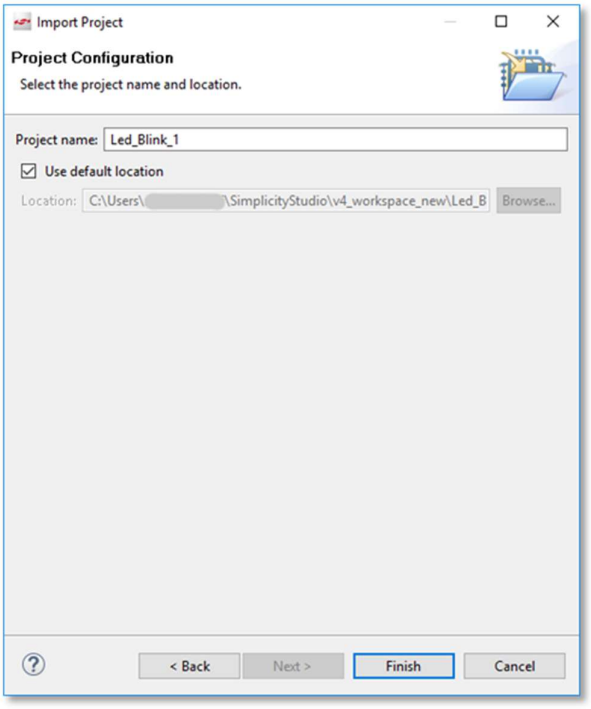
.2.6.1.3 Build Configuration

During this phase it is possible to personalize the building parameters of the project.



.2.6.1.4 Project Configuration

During this phase it is possible to select the workspace and the project name.

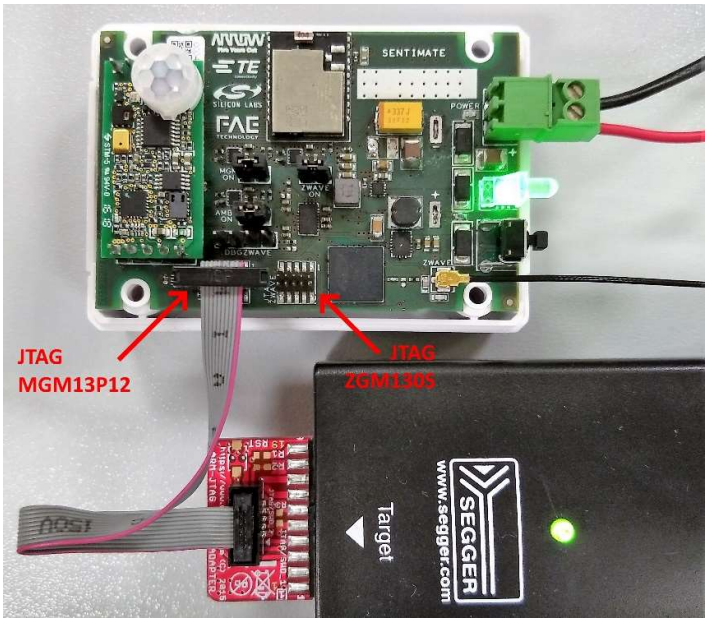


Selecting the “Finish” button the project import operation is concluded.
At this point it is possible to compile and run the example.

.2.6.2 How to connect the debugger

In order to program the microcontroller FLASH memory you need to connect the debugger to the JTAG port present on the SENTIMATE board.

In the following picture you can see the connection between the J-Link debugger and the JTAG port of the MGM13P12F512GA module.



NOTE: *It is recommended to use the JTAG connection into SWD mode*.

.2.7 Examples List

Summary tables of the code examples to develop on the SENTIMATE board.

Table 1 MGM13P12F512GA

Number	MCU	Name	Components
1	BLE / ZigBee / Thread	LED Blink 1	MGM13P12 – LED RGB
2	BLE / ZigBee / Thread	LED Blink 2	MGM13P12 – LED RGB
3	BLE / ZigBee / Thread	LED On - Off	MGM13P12 – LED RGB – Button
4	BLE / ZigBee / Thread	LED Fade	MGM13P12 – LED RGB
5	BLE / ZigBee / Thread	System Tick	MGM13P12
6	BLE / ZigBee / Thread	Battery Monitor	MGM13P12 – Battery
7	BLE / ZigBee / Thread	Low power	MGM13P12 – Power supply
8	BLE / ZigBee / Thread	Ambimate Sensors	MGM13P12 – Ambimate
9	BLE / ZigBee / Thread	Data Logger	MGM13P12 – Ambimate – Memoria Flash Esterna - LED RGB - Button
10	BLE / ZigBee / Thread	Microphone	MGM13P12 – Ambimate
11	BLE / ZigBee / Thread	Bluetooth Beacon	MGM13P12 – Battery
12	BLE / ZigBee / Thread	Bluetooth Sensors	MGM13P12 – Ambimate – LED RGB - Button

13	BLE / ZigBee / Thread	ZigBee Light	MGM13P12 - LED RGB
14	BLE / ZigBee / Thread	ZigBee Switch	MGM13P12 – Button

Table 2 ZGM130S

Number	MCU	Name	Components
1	Z-Wave	Z-Wave Light (End Device)	ZGM130S – LED RGB

.2.8 MGM13P12F512GA Module examples

Example 1	
Name	LED Blink 1
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to configure the GPIOs to achieve the blinking of the RGB LED
Specification	- Configure the GPIOs to pilot the RGB LED in On/Off mode. - The LED is switched on following the color sequence R -> G -> B -> R… - The time interval for the color change is 1 second and is achieved with the “Delay” function - The “Delay” function is implemented using the System Tick and its interrupt - The code execution is blocking
Project file	Led_Blink_1.sls

Example 2	
Name	LED Blink 2
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to configure the GPIOs to achieve the blinking of the RGB LED
Specification	- Configure the GPIOs to pilot the RGB LED in On/Off mode. - The LED is switched on following the color sequence R -> G -> B -> R… - The time interval for the color change is 1 second and is achieved through the TIMER peripheral with interrupt

	- The code execution is non-blocking
Project file	Led_Blink_2.sls

Example 3	
Name	LED On - Off
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to configure the GPIOs to achieve the blinking of the RGB LED through the button press
Specification	- Configure the GPIOs to pilot the RGB LED in On/Off mode. - Configure the GPIOs to read the button press. - At each button press, the LED is switched on following the color sequence R -> G -> B -> Off - Button debounce, wait until the button is released before continuing with the next reading - The button press event is configured with an interrupt on the GPIO - The code execution is non-blocking
File progetto	Led_On_Off.sls

Example 4	
Name	LED Fade
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to configure the Timer and the GPIOs to use the RGB LED via PWM

Specification	- Configure the Timer in PWM mode
	- Configure the GPIOs to pilot the RGB LED in PWM mode.
	- The code manages the fade of the RGB LED following the sequence:
	  ○ R from 0% -> to 100%
	  ○ B from 0% -> to 100%
	  ○ R from 100% -> to 0%
	  ○ G from 0% -> to 100%
	  ○ B from 100% -> to 0%
	  ○ R from 0% -> to 100%
	  ○ G from 100% -> to 0%
	  ○ Start over the cycle
	- The update interval of the PWM is every 10 milliseconds
	- The PWM step resolution is 1%
	- To achieve the time interval a Timer is employed
	- The code execution is non-blocking

Project file	Led_Fade.sls

Example 5	
Name	System Tick
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	Enable the System Tick to generate a time base of 1 millisecond and achieve a global int32 variable “SysTick_Count” that counts milliseconds since start up.
Specification	- Configure System Tick Timer with period 1 milliseconds - Increase the variable “SysTick_Count” in the interrupt handler

Project file	System_Tick.sls

Example 6	
Name	Battery Monitor
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh. Board powered by battery.
Description	The example shows how to configure the ADC and the GPIOs to read the value of the battery voltage.
Specification	- Configure the ADC. - Configure the GPIOs to enable the analog reading. - Convert the ADC value into the value of the battery voltage.
Project file	Battery_Mon.sls

Example 7	
Name	Low Power
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to configure the board to enable the low power mode in order to optimize the consumption when the device is powered by battery.
Specification	- Configure the GPIOs to disable the power supply to the hardware modules. - Configure the Bluetooth module to go in Low power mode with timed awakening every 3 seconds. - Green LED blinking with 2s duration to indicate that the module has awakened and then goes back in low power mode.
Project file	Low_Power.sls

Example 8	
Name	Ambimate Sensors
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to use the Ambimate driver to configure the module parameters and read the sensor values.
Specification	- Configure the communication on I²C bus. - Configure the Ambimate module. - Reading of all Ambimate sensors.
Project file	Ambimate_Sensors.sls

Example 9	
Name	Data Logger
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to use the Ambimate driver to configure the module parameters and read the sensor values.

	At power on, a green LED blinking every 2 seconds indicates that the memory is empty. If the memory already contains some data, the LED blinks in blue. At button press, if the memory is empty, the sensory data are read at time interval of 2 seconds and saved into the memory, while the led is blinking of color red at each writing in memory. If the memory is not empty, a led blinking in yellow indicates that a memory erasing is needed. Pushing again the button, stops the data recording and the led starts blinking in blue every 2 seconds to indicate that recording is not active but there are some data in memory. Holding the button pressed for more than 3 seconds causes the deletion of all the data. This operation is notified with the Led of color white.
Specification	- Configure the communication on I²C bus. - Configure the Ambimate module. - Reading of all Ambimate sensors. - Writing, Reading, Erasing of the memory
Project file	Data_Logger.sls

Example 10	
Name	Microphone
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to configure the ADC to sample the audio analog signal coming from the Ambimate module.

Specification	- Configure the ADC channel.
	- Configure a timer to generate a frequency to sample the audio signal of 8KHz
	- Acquire the audio signal and analyze the volume
	- Feedback of the audio volume through the LED
	  ○ LED switched off for no volume
	  ○ LED green for low volume
	  ○ LED yellow for medium volume
	  ○ LED red for high volume
	Note: Audio thresholds are just empirical

Project file	Microphone.sls

Example 11	
Name	Bluetooth Beacon
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to use the Bluetooth Low Energy radio to enable the Beacon Eddystone_(Google) functionality. The beaconing must show info about the hardware such as temperature and battery level. Battery level is read from the ADC that must be configured, while the temperature is read from the Ambimate module.
Specification	- Configure Bluetooth radio - Configure the ADC channel. - Configure the CRYOTIMER to generate a frequency to acquire the battery level - Feedback through blue LED blink
Project file	Bluetooth Beacon.sls

Example 12	
Name	Bluetooth Sensors
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to use the Bluetooth Low Energy radio to send the read data to the app of the telephone (Blue Gecko). When the board is on but there is no telephone connected, the LED is blinking Green every 2 seconds. When the device is connected to the app, the LED is blinking Blu every 2 seconds to signal that the connection is successful. Sensory data acquisition from the Ambimate module and sending towards the phone starts afterwards the connection with an update frequency of the data every 3 seconds. After the disconnection event the board goes back to standby with green blinking and the reading of the data from Ambimate module is suspended.
Specification	- Configure Bluetooth radio - Configure ADC channel. - Configure the TIMER1 to generate a frequency to acquire the battery level - Acquire sensory data from Ambimate module - Feedback of connection and working status through RGB LED
Project file	Bluetooth Sensors.sls

Example 13	
Name	Ambimate_PIR
Prerequisite	Simplicity Studio V4 + SDK Bluetooth Mesh
Description	The example shows how to use the Ambimate driver to configure just the PIR sensor (movement detection) and give a LED feedback when the movement event is triggered. The movement event is associated with the interrupt of the EVENT_OUT signal, so reading and writing of the Ambimate sensor will be synchronized with that event.
Specification	- Configure the communication on I²C bus. - Configure the Ambimate module to activate just the PIR sensor. - Configure the interrupt bound to the EVENT_OUT signal of the Ambimate module. - Feedback of movement recognition through red LED
Project file	Ambimate_PIR.sls

.2.9 ZGM130S Module examples

The examples for the Z-Wave module are based on using a gateway.
For example, you can create a gateway using the following tools:

- USB UZB-7 Stick
- Z-Wave PC Controller Software (This software is included inside the installation folders of Simplicity studio at "*C:\SiliconLabs\SimplicityStudio\v4\developer\tools\zwave\pc_controller*").

Example 1	
Name	Z-Wave Light
Prerequisite	Simplicity Studio V4 + Z-Wave SDK - 7.11.0.0
Description	Exploiting the libraries for the Z-Wave protocol, enable the functionality of an EndNode light device. Through a Z-Wave gateway it is possible to control the device to switch on or off the LED.
Specification	- Configure Bluetooth radio in ZigBee mode - RGB LED used as bulb light
Project file	Z-Wave_Light.sls

NOTE: *in order to join the SENTIMATE board to the Z-Wave net, it is necessary to start the association by shortly pushing the button*.