

The IoT LoRa Project

Design Report



www.digilentdesigncontest.com

The IoT LoRa Project

Paul Leloup

paul.leloup@outlook.com

Zakaria El Rhosn

eir.zakaria@outlook.fr

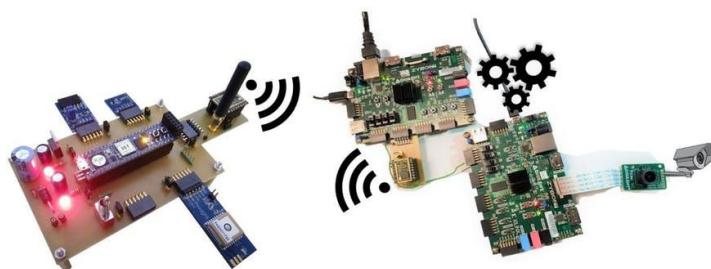
Submitted for the 2019 Digilent Design Contest Europe

05/05/2019

Advisor: Thierry Gil

**Polytech Montpellier
Montpellier, France**

The IoT LoRa Project



1. Summary

2.	Acknowledgements	2
3.	Introduction.....	3
4.	The Smart Gateway	4
4.1.	Smart Cameras	4
4.2.	Hardware and Software Architecture	5
5.	The Smart Sensor	8
5.1.	Specifications.....	8
5.2.	System	14
5.3.	Hardware (FPGA).....	16
5.4.	Software (Flow chart).....	18
5.5.	Result.....	23
6.	Conclusion	26
7.	Reference	27
7.1.	Smart Gateway Reference.....	27
7.2.	Smart Sensor Reference	27
8.	Appendix.....	28

2. Acknowledgements

I thank the Laboratory of Computer Science, Robotics and Microelectronics of Montpellier (LIRMM) for having welcomed me during this Industrial End of Study Project (PIFE).

I would especially like to thank Pascal Benoit, an university lecturer at Polytech and LIRMM, who gave me his confidence to carry out this project and who supervised me every week, as well as Guillaume Patrigeon, PhD at the LIRMM, which helped me greatly in the installation of the workstation and throughout these 200 hours of work.

To do my PIFE at the LIRMM was a pleasure, I could learn a lot thanks to you and to validate my acquired of my curriculum in engineering school.

3. Introduction

The number of connected devices (Internet of Things: IoT) in circulation in the world in 2018 is 23.14 billion [1] and in the next few years the number of devices could reach between 50 and 80 billion. It is estimated that connected devices worldwide in 2013 consumed 616 Twh [2], which is greater than the consumption of France in one year (eg 544 Twh in 2015 [3]). The saving of electric current represents billions of euros if the consumption of these objects was decrease.

Connected and smart objects have become an integral part of our daily lives, thanks to the evolution of the Internet of Things (IoT) domain. Experts from the world of technology, multinationals and startups, seek to make connected objects even smarter, smaller, multitasking and, above all, consuming the least amount of energy possible.

As part of research work carried out at LIRMM on IoT, an experimental platform consisting of connected objects, network gateways and servers for the processing of collected data. My project, part of this work, is to study and design a Smart Camera that ensures the acquisition, analysis and transmission of these data after treatment.

The project is part of an industrial end-of-course project which aims to put the student in the position of a project manager responsible for finding solutions to a concrete problem that is posed to him. The title of PIFE is "Study & Design of a Smart Sensor on FPGA".

The IoT project purpose is very interesting to me, and designing a smart object as well as programming and participating in its improvement is relevant because I want to work in this field.

The Internet of Things (IoT, Figure 1) is the extension of the internet to things and places in the physical world. The connected Internet of Things represents the exchange of information and data from real-world devices with the Internet.

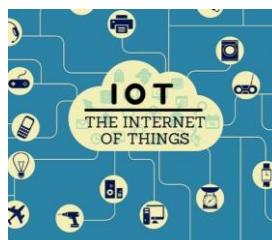


Figure 1 Illustration IoT

The place of study is the Laboratory of Informatics, Robotics and Microelectronics of Montpellier (LIRMM, Figure 2) which is a joint research unit jointly dependent on the University Montpellier (UM) and the National Center for Scientific Research (CNRS).



Figure 2 LIRMM photo

4. The Smart Gateway

4.1. Smart Cameras

Smart Cameras are the new generation of normal cameras that only acquire images and transmit them in video format. The latter are able to process these data, analyze them according to video processing algorithms in order to derive information considered relevant by the user.

More and more companies that manufacture audiovisual devices decide to equip their camera with intelligence like the cameras offered by the American company NEST Labs, a subsidiary of Google and specializing in home automation. NEST cameras are smart cameras that can detect people and recognize familiar faces. These IP cameras transmit information (videos, images and notifications) via Wi-Fi or Bluetooth Low Energy (BLE), with a bandwidth required of at least 2Mbit / s, streaming and video recording as well as notifications on mobile devices require a broadband internet connection and a functional Wi-Fi network. NEST cameras are very demanding in terms of power, bandwidth and flow.

These types of connected cameras, although NESTs have AES encryption systems, but transmitting video data via IP itself increases the risk of hacking, especially with the integration of more and more objects connected the number of IP gateways also increases.



The LIRMM smart camera, connected to an intelligent network that consists of several other sensors, will use much less bandwidth to transmit information. These data are broadcast in the intelligent network and can be used by other connected objects or even by Smart Cameras fixed elsewhere.

Such systems can be used in many areas such as remote monitoring, road traffic and various applications in industrial and domestic environments such as smart homes ...

Smart homes

During the last decade, the field of home automation and smart homes has undergone remarkable development. CCTV is one of the main concepts of Smart Homes, with the installation of surveillance devices for home security, tracking objects (furniture ...) and surveillance of pets.

Smart cameras made up of analysis and processing processors, cameras and communication modules offer intelligent monitoring. While normal cameras are used to capture what is happening in monitored locations, the purpose of smart surveillance is to effectively inform the user of events that may be of interest to them. Because with Smart cameras, the user will no longer have to view video recordings to find out what is going on in the house, but thanks to the processing algorithms that are implemented in the FPGA / Microprocessor, the Smart camera analyzes the information acquired. and warn the user of relevant events (depending on the algorithms used). Thanks to the communication modules, this information can be communicated to other cameras fixed in other corners of the house.

Traffic Road Surveillance

After the integration of smart cameras in homes and industrial settings, their use for traffic monitoring has a lot of benefit for traffic control, accident detection and car tracking.

Roads equipped with these new generation cameras will have the ability to monitor and alert the right people according to the event produced.

Several analysis algorithms are developed and implemented in smart cameras to perform different tasks [2]. We can then detect vehicles [2], determine their speeds, analyze the behavior of cars to detect accidents as soon as they occur, and also track and predict car travel [2].

With these smart cameras, people who monitor the road will be quickly notified of accidents and their location will be even easier. Because to monitor a road, it is necessary to be attentive to a hundred screens at the same time, something which is impossible, whereas with such cameras, the qualified agents will have to be attentive to the flow of messages sent by the smart cameras containing the information on the current event (vehicle traveling at high speed, sudden movement of the vehicle or accident ...).

Smart cameras have a lot of benefit for road monitoring, they will detect accidents as soon as they happen, locate them to act as soon as possible, as they can track cars and plan their trips to control the traffic, traffic and even avoid traffic jams.

Synthesis

Thanks to the modules that make up Smart Cameras, they can ensure maximum surveillance in different environments (houses, industries, roads ...) and with low bandwidth consumption.

Smart cameras transmit messages via low bit rate networks and low power wide area network (LPWAN) scopes [3], which are widely used to connect applications in the IoT domain.

They thus ensure the surveillance of an environment by keeping a confidential nature because they make it possible to warn the user of the relevant events (human presence in a place, to count the number of people entering / leaving ...) without revealing totally what happens in the middle. This guarantee of confidentiality is a very important asset that may interest several infrastructures that rely on camera surveillance to ensure security, event inspection...

4.2. Hardware and Software Architecture

The Smart Camera consists of three main elements: the camera, the development board and the message transmission module. These allow successively acquisition, analysis / processing and the transmission and reception of data.

The hardware part designates the architecture of the Smart camera system with the IPs (Intellectual properties) which compose it: Zynq Processing System, GPIOs, AXI QUAD SPI ... While the software part concerns rather the distribution linux, the programming of the gpios, module of communication...

Camera

Unlike IP cameras that are connected to computer networks, via optical cable or copper or even WIFI connection, the risks of saturation are reduced. In fact, to transmit the data captured by IP cameras, a large bandwidth is required which can vary considerably according to several parameters (type of video compression, image resolution, quality, number of cameras, images per second, etc.). This type of camera is therefore not suitable for the system to which we want to integrate our Smart Camera, which in reality is an intelligent gateway that contains several other sensors.

We therefore chose a Digilent PCam 5C camera designed to be connected to a development platform with FPGA.

LPWAN Networks

The concept of our smart camera is based on the fact that the information published in the network is small, so we choose LPWAN (Low Power Wide Area Network) wireless modules. The LPWAN communication modules allow the integration of a very large number of sensors and other connected objects (IoT) since these are used only for data services that require very few flows.

Several networks have managed to win in the world of IoT and have sparked our interest in the Smart Camera system and its applications, among them we find the LoRa, SigFox and NB-IoT.

The use of a LPWAN module for communication is a major asset for the privacy of environments equipped with Smart Cameras. Indeed, the latter have a capacity of data transmission between 22 and 256 bytes, which makes impossible the transmission of videos captured by the camera.

Each of the three networks offers different benefits and has its limits. QoS quality of service means that NB-IoT is favored for applications that require the guarantee of high QoS, it also offers a maximum data transmission compared to all other LPWANs with a higher QoS. transmission capacity of 16000 against 2430 for LoRa and 120 for Sigfox. However, NB-IoT has higher energy consumption and higher cost (Figure 2).

Depending on the type of application of the Smart Camera, the LoRa proves to be the most suitable, especially since it is less restrictive in terms of the size of the data to be transmitted while having a large reach and a low consumption, without forgetting the security of communications with AES encryption not found in Sigfox

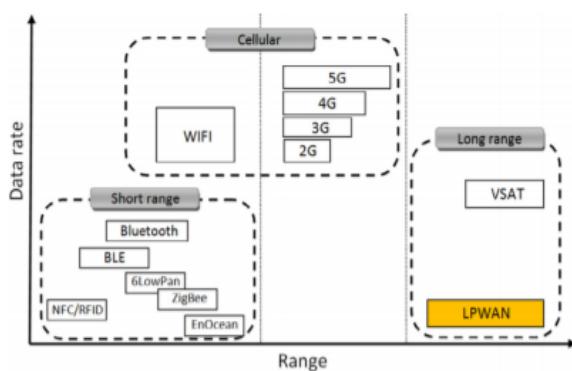


Figure 1 Position of network LPWAN in function of his size of data/range [4]

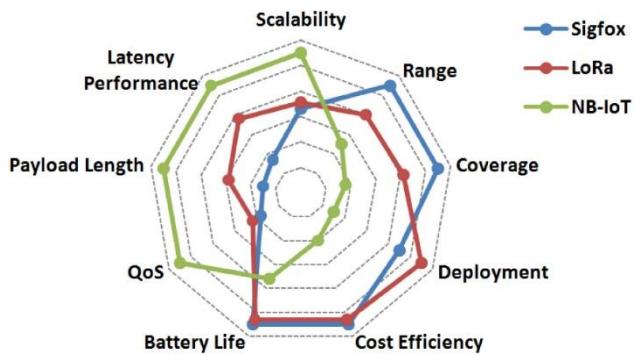


Figure 2 Position of specifications of Sigfox, LoRa and NB-IoT

Data Processing

The Smart Camera processes the images acquired through the two parts of the development platform: Programmable System with an ARM dual core and Programmable Logic representing the FPGA. The treatment consists in recognizing the objects visualized via the camera and to do this, we use image processing algorithms.

There are many image processing algorithms based on openCV library [...] tools to develop different programs such as motion detection and tracking [5]. We can realize and implement several processing

algorithms for different tasks in order to allow the Smart camera to inform us of several events such as object recognition, presence detection, know if a person comes in or out ...

In order to synthesize algorithms designed with C / C ++, SystemC or even OpenCL programming languages, the VIVADO High Level Synthesis (HLS) tool is used. The latter provides an integrated graphical environment for rapid production of IP and systems-on-a-chip assembly [6], these IPs can include logic circuits, signal processing modules and algorithmic descriptions written in C [7].

These systems will be the elements that will be of particular interest after the validation of the first version of the Smart Camera (Acquisition, Processing and Transmission messages). Because they can be equipped with artificial intelligence with the use of neural networks.

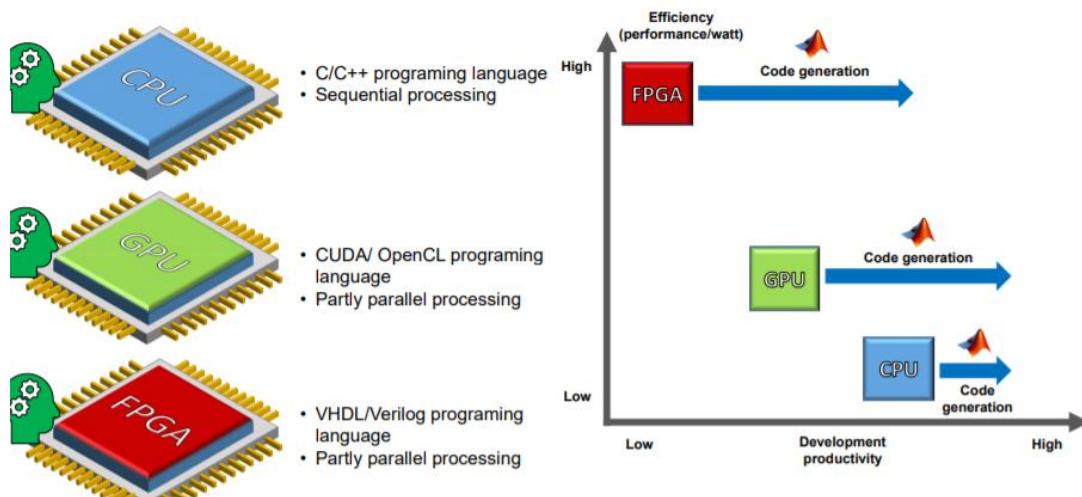


Figure 3 IA on CPY, GPU and FPGA Compare[8]

Machine Learning techniques can be implemented on several embedded targets for IoT. Compared to CPUs and GPUs, FPGAs offer an architecture better suited to the implementation of AI algorithms (Artificial Intelligence) as well as better energy efficiency [8].

FPGAs therefore offer a lot of advantages in the development of machine learning. It is therefore wise to rely on a development platform with FPGA for the evolution of the Smart Camera and here we talk about the scalability of the system.

5. The Smart Sensor

5.1. Specifications

Subject / Goal

Présentation of the project:

The project was proposed by Pascal Benoit, because as part of research work carried out at the LIRMM on the theme of the Internet of Things, researchers develop an experimental platform consisting of connected objects, network gateways, as well as servers. for the processing and visualization of collected data.

For this, they work on a prototype Smart Sensor: it is based on an FPGA card, to which any type of sensor and radio can be connected. This architecture allows rapid prototyping of a connected sensor architecture: for example, one may want to test and compare two types of radio transmission, or configure in the FPGA a Cortex M0 processor or a Cortex M3, or play on the size of embedded memory, all this in order to evaluate the various solutions, and to be able to compare them in terms of performance and energy.

This Smart Sensor prototype is based on a DIGILENT CMODA7 card for which a complete microcontroller architecture is described in VHDL is available: it is composed of a Cortex M0 processor, an internal memory, and various peripherals. In order to quickly prototype various types of technologies we want to create a port expansion card that will connect different types of radio modules and sensors.

The objective of the project is to create a test bench to calculate the consumption and performance of a smart object to optimize them. This is to help Guillaume Patrigeon's thesis on "Ultra-low power adaptive integrated systems for the Internet of Things".

Firstly, it will be necessary to make a bibliographic study of the different types of sensors and radio modules possible, and to specify several cases of use.

Then I have created an electronic board extension ports that allows you to connect different sensors and radio modules in order to customize the smart object. This card will be the Smart Sensor, it will also be called the sensor node. It will connect the CMOD A7, radio modules and sensors.

Once the card is created, there will be a software part that will capture the information from the sensors, which will process them and then send them by radio.

In the end, when the test bench will be functional, we will be in a real application of intelligent object. There will be an architectural exploration later to lower the consumption of microcontrollers implemented in connected objects.

Here are the specifications for the work to be done during this Industrial End of Study Project.

Specifications :

- Make a bibliographic study of the different types of sensors and radio modules possible, and specify several use cases
- Operate the GPS module and send the data received by LoRa with the Cmod A7
- Make a Smart Sensor electronic card that will have these features:
 - Modular thanks to Pmod connectors
 - Possibility to connect at least 5 modules (sensor, radio modules)
 - Reconfigurable and flexible thanks to the Cmod A7 FPGA
 - Reprogrammable by micro USB easily accessible
 - Battery powered for outdoor testing and less power supply noise than USB
 - Easy to measure the consumption and power of each module
 - Voltage regulator and current limiter system to secure the components of the board.
- Make the datasheet of the Smart Sensor card
- Make a Pmod LoRa electronic card that will have these features:
 - Modular thanks to a Pmod connection
 - Possibility to connect a LoRa module
 - Possibility to connect an antenna
 - Easy to measure the power consumption and power of the module
- Make the datasheet of the Pmod LoRa card
- Option: Add temperature, humidity and brightness sensors and program them with the electronic board created to send the data by LoRa
- Option: Add a Bluetooth radio module and send the created frame with it
- Option: Make a consumption measurement of the Smart Sensor

Planification des tâches

See the Gantt created at the beginning of the project in Appendix (Gantt).

System

The system has the same architecture as a connected object, the Smart Sensor architecture is like the diagram below (Figure 3).

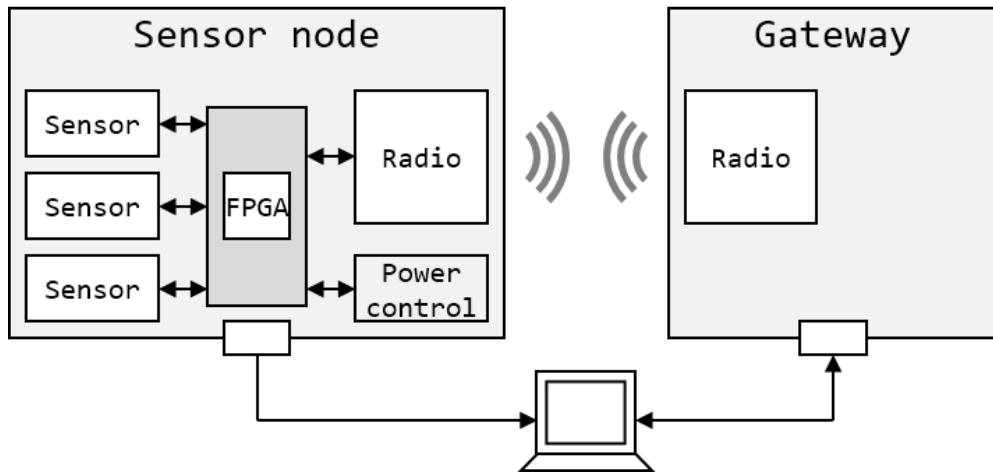


Figure 3 Architecture of Smart Sensor and his Gateway

The card consists of a microcontroller (implemented in an FPGA, the Cmod A7) that will communicate with modules that may be sensors or radio communication systems. Part for battery power is also present.

The particularity of the Smart Sensor that makes it a smart object like no other is that it is modular and therefore reconfigurable. So all possible sensors and radios can be connected to the board if they have Pmod connectivity. Here there are a picture of sensors and radio modules with the Pmod connection (Figure 4)



Figure 4 Pmod modules

We chose to use this connection because there are a very large number of sensors with it. It is also common for prototyping quickly.

These modules also have the advantage of never using analog pins, they communicate with the microcontroller using communication interfaces such as UART, SPI, I2C and GPIO.

Here is the diagram of the Smart Sensor (Figure 5). Each module has Pmod connectivity and they are all interchangeable.

It will just fit the code and indicate the change of the pins.

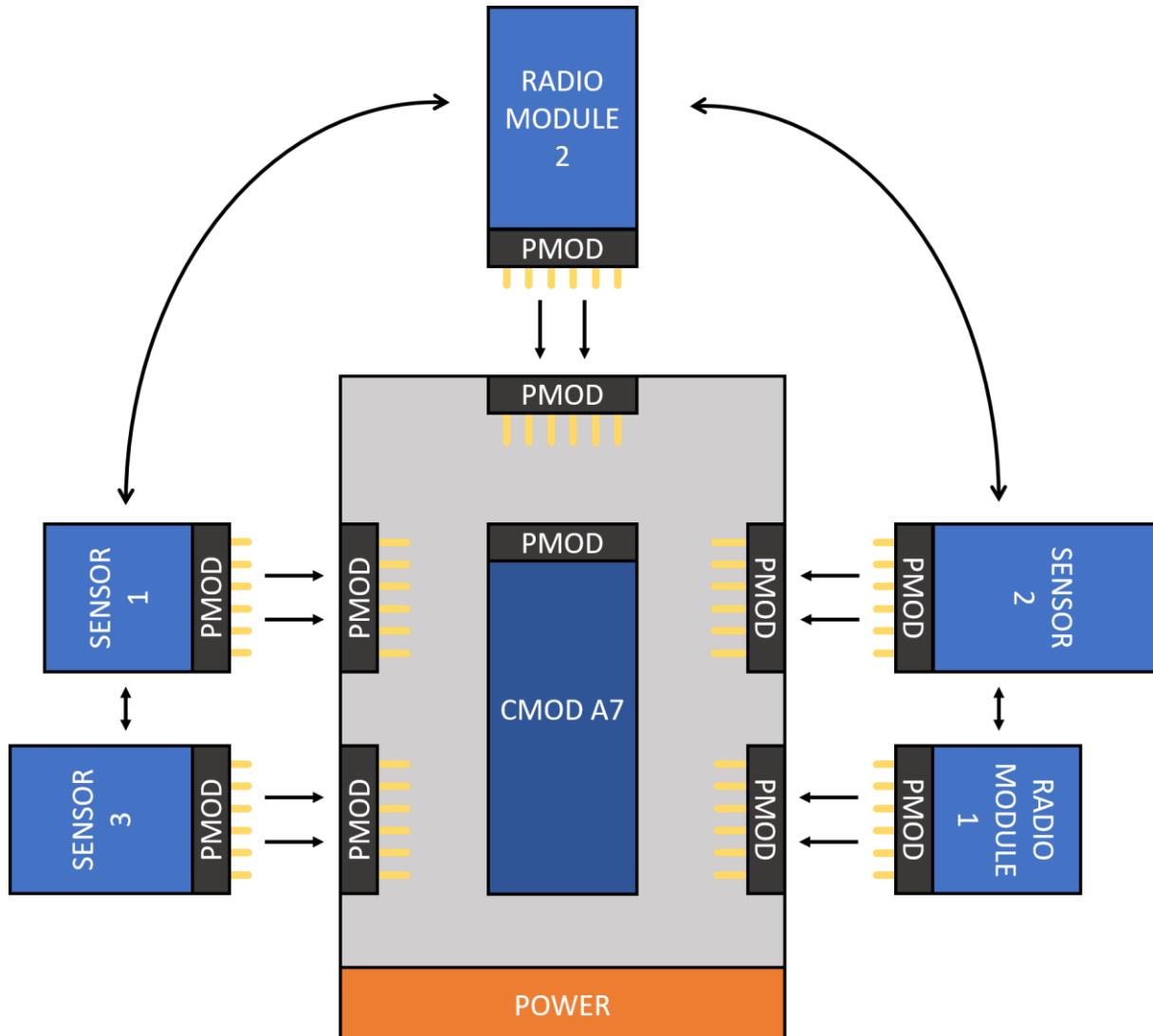


Figure 5 Smart Sensor modulaire

In terms of power, the Smart Sensor is powered by either USB-microUSB cable or battery. The two power systems each have their advantages:

- The USB-microUSB power cable allows you to upload the C code to modify the program. It also allows software development because you can view a UART on the terminal of the computer. This makes it easy to debug and view the data sent and received from the Gateway.
- Battery power has very little noise (high frequencies) and allows for precise and noise-free consumption measurements unlike USB generated by a computer. There is also the advantage of mobility, which allows for scope testing with the Gateway.

Hardware

The main controller of the node is the Digilent CmodA7. It is a compact 48-pin DIP card that features an Artix 7 FPGA from Xilinx. This is the version of the CmodA7 35T that will be used here. As the CmodA7 already has pushbuttons and light-emitting diodes, it will not be necessary to overload the motherboard of the sensor node with these elements.

In our application, the first need is that the hardware is reconfigurable. Thanks to the flexibility of the FPGA, it is possible to explore different architectures and therefore different processor solutions (for example: ARM Cortex-M0 and Cortex-M3, LIRMM's SecretBlaze, etc ...). It is also possible to replace this card with another, with a microcontroller trade to compare the proposed solutions with existing products. This study of microcontroller architectures is part of other projects within the LIRMM.

Here are the pros and cons of using an FPGA chip:

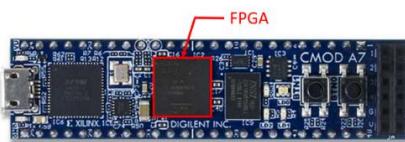


Figure 6 CMOD A7 card of DIGILENT

Advantages	Disadvantages
<ul style="list-style-type: none"> Possibility of prototyping Flexibility Time-to-market weak Adaptability to future evolutions thanks to the reconfiguration 	<ul style="list-style-type: none"> Limited integration by routing resources Performance High unit price for large productions

What interests us the most is the possibility of prototyping and flexibility. Although the Artix 7 FPGA present is limited in terms of performance and resources, the targeted architectures are those of very low-power microcontrollers operating at relatively low frequencies (order of ten MHz) and with a reduced number of peripherals. The limited integration problem due to routing resources has been eliminated because the system is modular with Pmod connectivity and does not require analog pins.

We will integrate in the FPGA chip a microcontroller with ARM cortex-m0 processor. To integrate a microcontroller into an FPGA, it must be described as a VHDL file (.VHD) and follow all the steps summarized in this diagram (Figure 7).

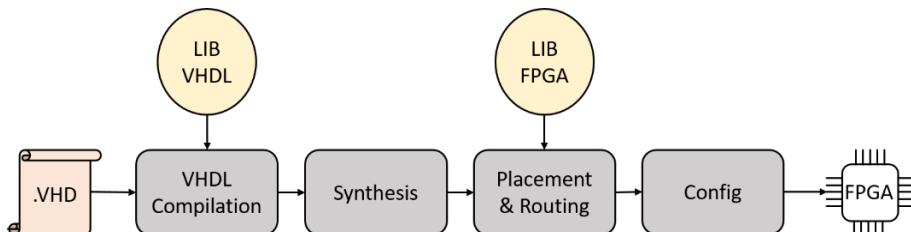


Figure 7 Integrate a microcontroleur in a FPGA

Software

The software is reprogrammable as in most cases. It allows to configure all the GPIO input and output, configure up to 12 communication interfaces (4 UART, 4 SPI and 4 I2C) each pin of these interfaces can be chosen by software.

At the level of the global structure of the code, here is the cyclic functioning of the code (Figure 8).

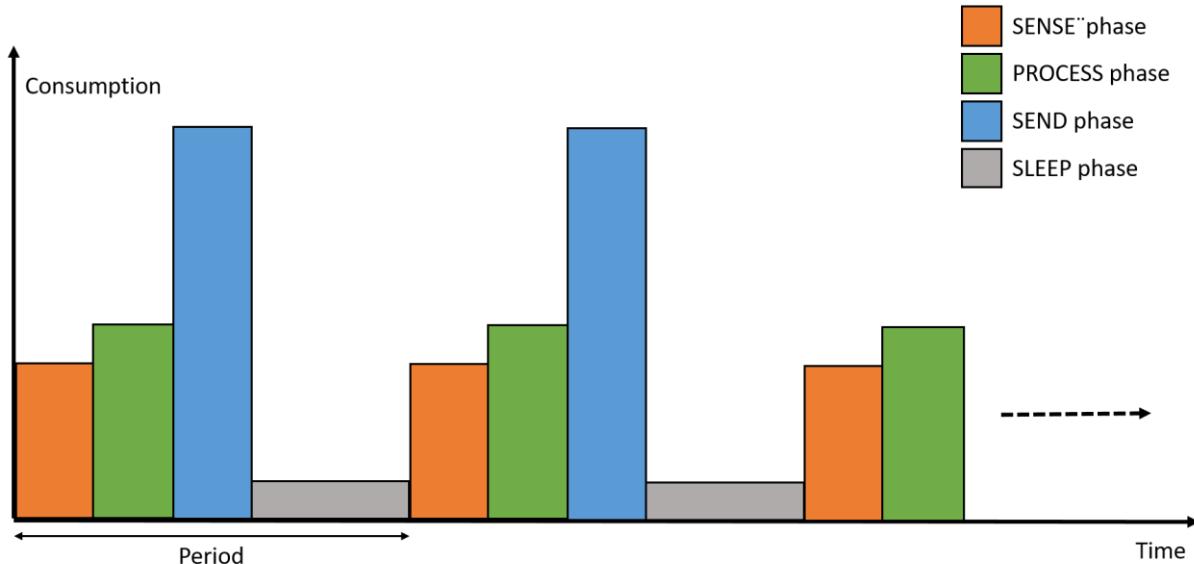


Figure 8 Differents steps of the software

It is composed by 4 parts that are:

- The SENSE phase: It allows to collect the information of the sensors and stores them in a variable.
- The PROCESS phase: It processes the bytes recorded by the previous phase in order to extract the desired information. This information is formatted and grouped in a concise message (a frame to send).
- The SEND phase: This phase sends the previously created frame with one or more radio modules (eg LoRa, BLE).
- SLEEP phase: It puts the microcontroller in sleep mode to save energy because sending the temperature every 5 ms would not be relevant in this application.

5.2. System

Design of the card

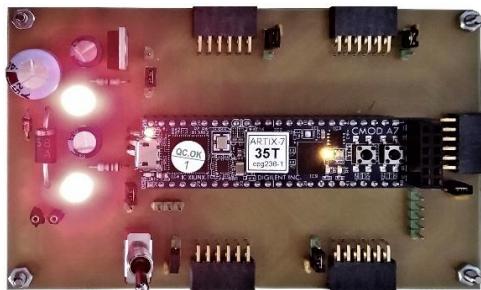


Figure 9 Smart Sensor Card

Electronic card

For more information on the Smart Sensor map, please go to the appendices to view the Smart Card Datasheet. The schematic of the board as well as the board is also in the datasheet.

The electronic board (Figure 9) consists of 5 female Pmod connectors where each pin of each connector is connected to CMOD A7 which is the component in the center of it.

It was chosen a power system to regulate the battery voltage. The voltage of the battery is 7.4V, a voltage higher than 5V to be able to supply all the components of the circuit. There is a first linear voltage regulator to lower the voltage from 7.4 to 5V. 5V is the voltage required by the CMOD A7 to operate. Since there is no 3.3V power pin on the CMOD A7 board, another linear regulator has been placed at 5V to generate a 3.3V supply voltage for all Pmod connectors. Since linear regulators dissipate the extra voltage into heat, it was more appropriate to put them in series to avoid overheating. There are also different components ensuring the robustness of the power supply and the low noise:

- A Schottky diode in series to prevent reverse currents and not to degrade the battery.
- A Zener diode in parallel to have the most constant supply voltage possible.
- Large capacitors in parallel to fully respond to current calls from the Smart Sensor.
- LEDs to visualize the 5V and 3.3V power supply.
- Small capacitors in parallel to limit the high frequencies on the power supply as close as possible to the output pins of the power supply.

I have also positioned pins that allow measurement of current and voltage simultaneously (power). This makes it possible to measure the consumption of each Pmod separately, of all the Pmods together or of the CMOD A7.

Problèmes rencontrés

- Size of the holes for the components to check well (All the holes of the pins of connectors must be in 1mm, the regulators of tension, the connections Pmod also).
- The gap between the two CMOD A7 pin rails was not good.
- The battery connector is in CMS, so on the back, specify it in the software otherwise symmetry problem between VCC and GND.

As solutions to these problems it was made a second version of the card of which no problem has been detected until today.

LoRa



Figure 10 LoRa Pmod

For more information on the Pmod LoRa card, please go to the appendices to view the map datasheet (Datasheet Pmod LoRa). The schematic of the board as well as the board is also in the datasheet.

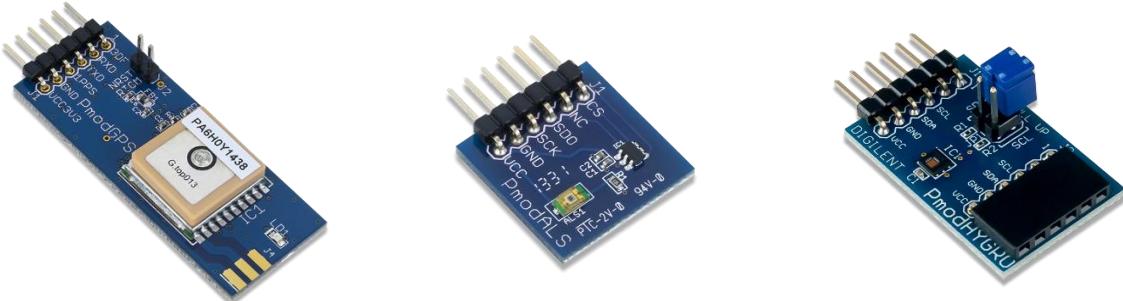
The electronic board (Figure 10) consists of a male Pmod connector where each pin is connected to LoRa RF-LoRa-868-SO module. The 5 extra pins are wired to a single connector.

An 868 MHz SMA antenna can be easily connected using the added connectors.

There is also the same system of measurement of consumption that on the map of the Smart Sensor.

This LoRa module was first used on a breadboard and without an antenna and thanks to this card, the use of it is much easier. It communicates by SPI with the microcontroller.

Pmod Sensors



(a) GPS Pmod

(b) ALS Pmod

(c) HYGRO Pmod

Figure 11 Differents Pmod used

This GPS Pmod (Figure 11 (a)) was the first sensor used with the CMOD A7, it is easy to use and uses the UART to communicate. It has also been used on breadboard.

The ALS Pmod (Figure 11 (b)) captures the brightness, it uses the SPI interface as well.

The latest Pmod sensor is the HYGRO Pmod (Figure 11 (c)), which captures air humidity and temperature. The one is different and uses the I2C interface.

All these sensors will make interesting measurements of consumption by the actual application of the Smart Sensor.

5.3. Hardware (FPGA)

System

The system has a global structure as below (Figure 12), it is the typical structure of a microcontroller.

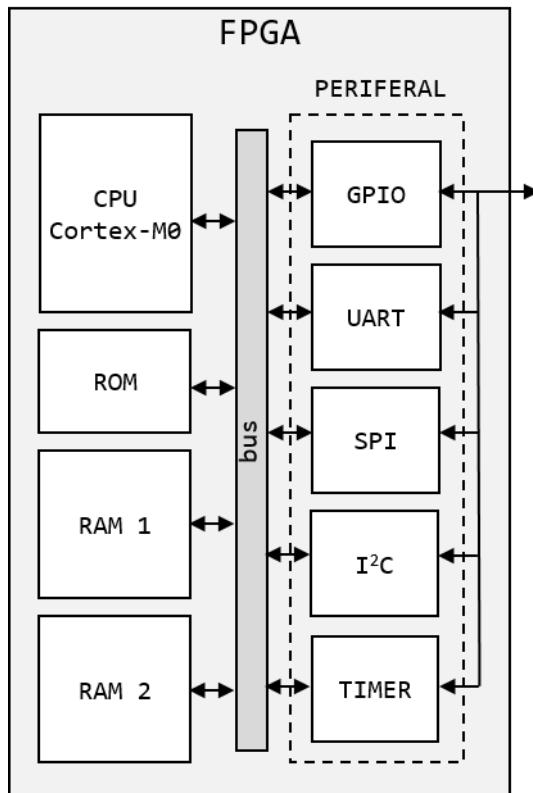


Figure 12 global Structure (FPGA)

So we have inside the FPGA chip a microcontroller composed of a processor, the Cortex-M0 whose specificities will be detailed in the next part, a ROM, two RAM and various peripherals. The TIMER will mainly serve to set a cycle of a constant period with interruption to release the microcontroller from its standby state.

Processor

The processor used in this study is the Cortex-M0 r1p0, from the well-known ARM Cortex-M series. Products based on this processor can be found to validate and compare the operation of the system implemented on the FPGA. The Cortex-M0 is a 3-stage 32-bit RISC processor that implements the ARMv6-M instruction set. It is given for a maximum frequency of 50 MHz, has 32 interrupt lines, a non-maskable interrupt line and a 32-bit single cycle multiplier in its r1p0 version. It has only one hardware interface, type AHB-Lite.

Here is the diagram of the functional blocks of the processor (Figure 13).

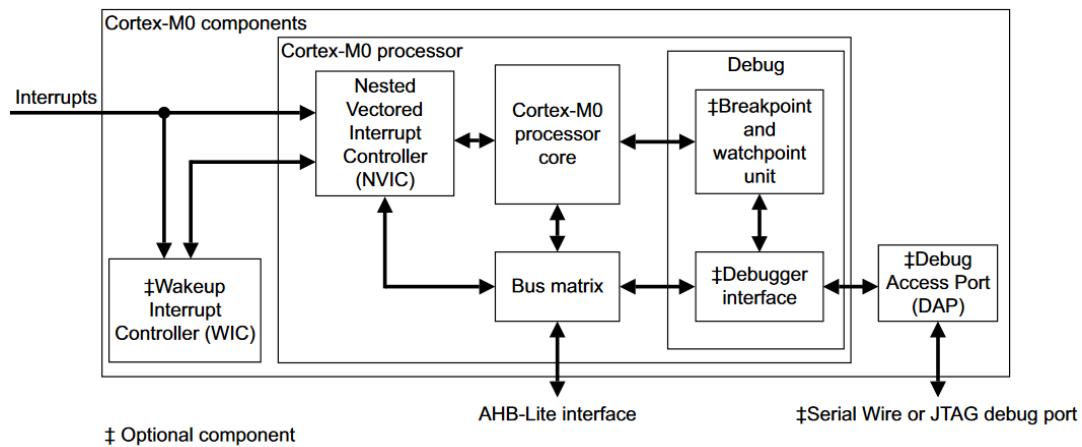


Figure 13 Functional block diagrams of the Cortex-M0

UART, SPI, I²C Interfaces

To communicate between the modules and the microcontroller, we will use communication interfaces such as UART, I²C and SPI.

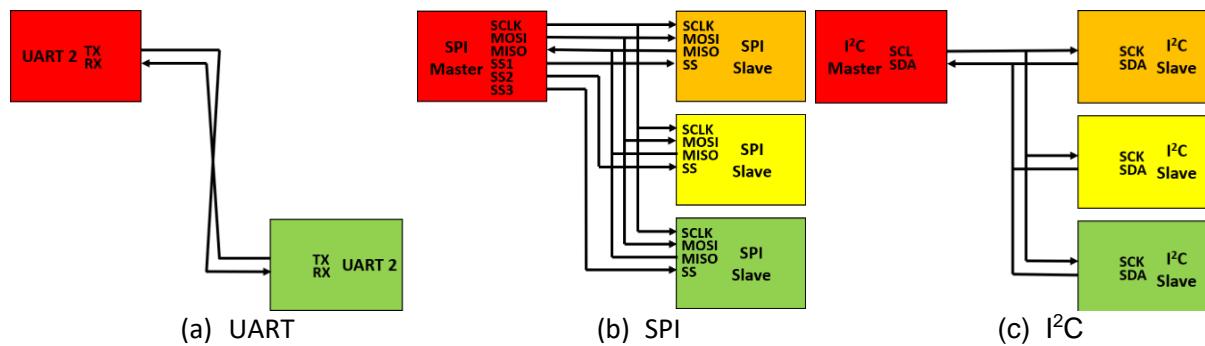


Figure 14 Communications interfaces

A UART (Figure 14 (a)), for Universal Asynchronous Receiver Transmitter, is a universal asynchronous transceiver.

An SPI link (Figure 14 (b)) (for Serial Peripheral Interface) is a synchronous serial data bus. The circuits communicate according to a master-slave scheme, where the master controls the communication. Several slaves can coexist on the same bus, in this case, the selection of the recipient is done by a dedicated line between the master and the slave called Slave Select.

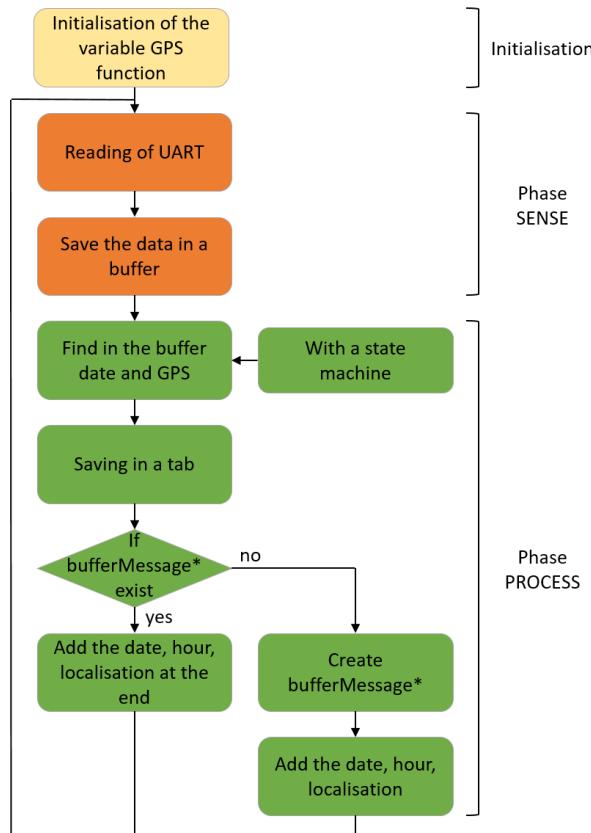
I²C (Figure 14 (c)) is a half-duplex bidirectional synchronous serial bus, where multiple devices, masters or slaves, can be connected to the bus.

UARTs, SPIs, and I²Cs are usually integrated into components like microcontrollers. In this case they are a peripheral function of the microcontroller.

5.4. Software (Flow chart)

GPS

The flowchart below (Figure 15) shows 3 functions, the "initGPS" function that initializes the Pmod, the "readGPS" function (Phase SENSE) and the "processMessageGPS" function (Phase PROCESS) that creates the frame (Figure 16).



*bufferMessage : It's the tab where we save data for send by radio.

Figure 15 Flow chart of the GPS function

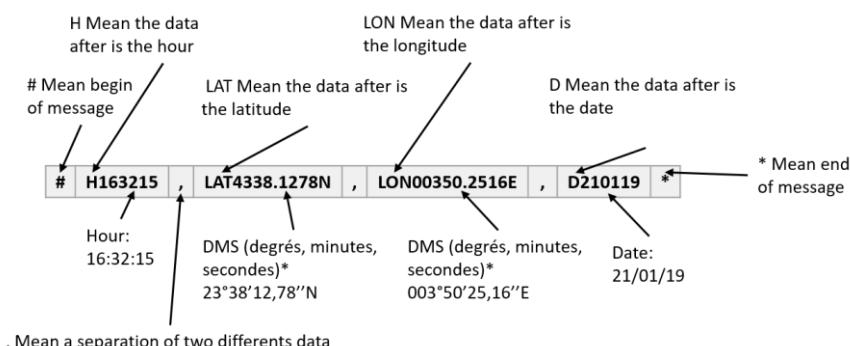
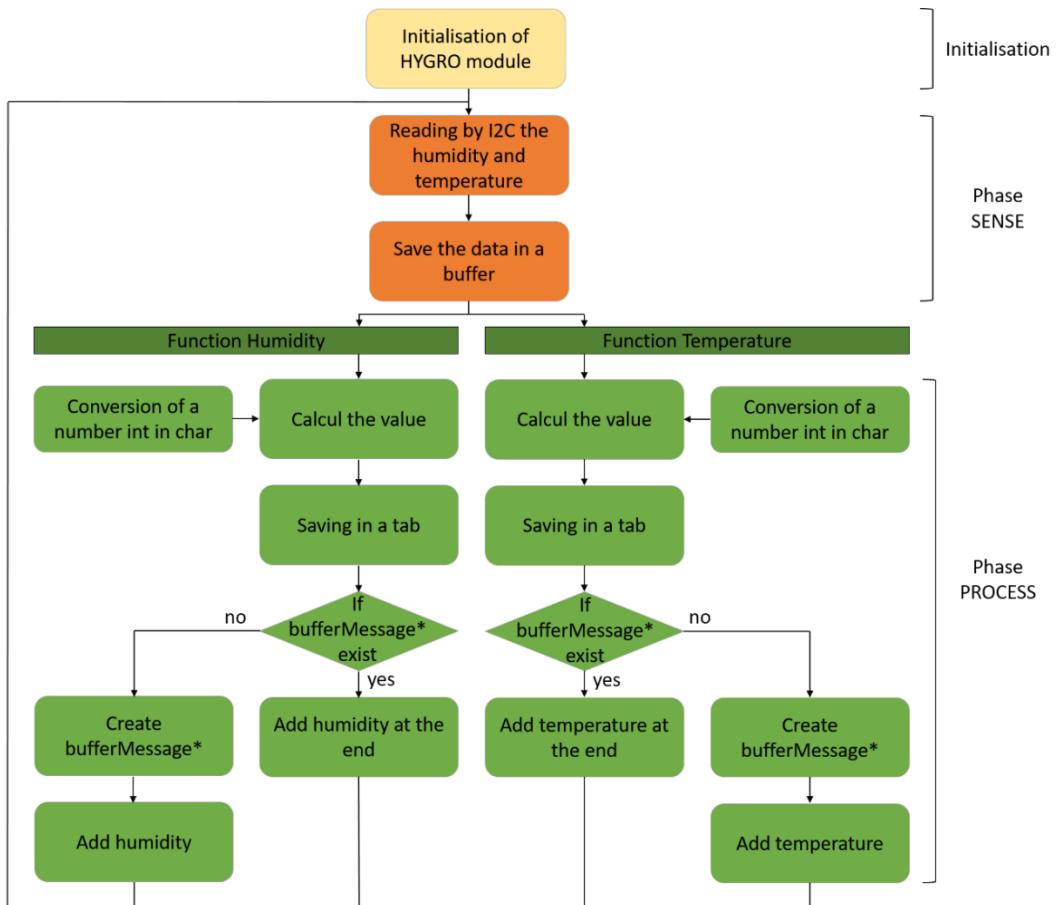


Figure 16 Function GPS frame

HYGRO

As the flow chart shows (Figure 17), there are going to be 4 functions here. The "initHYGRO" function to initialize the Pmod, the "readHYGRO" function that captures the humidity and temperature, then the last two functions "processMessageHumidityHYGRO" and "processMessageTemperatureHYGRO" which records and processes the raw data of the humidity and the temperature to create the frames (Figure 18, Figure 19).



*bufferMessage : It's the tab where we save data for send by radio.

Figure 17 Flow chart of the HYGRO function

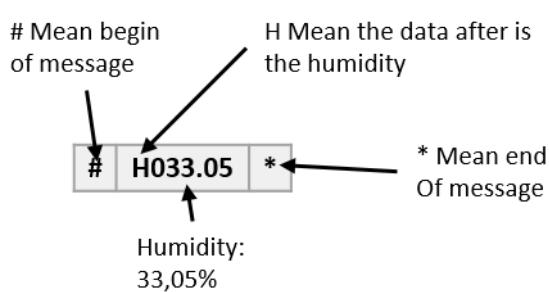


Figure 18 Function Humidity HYGRO frame

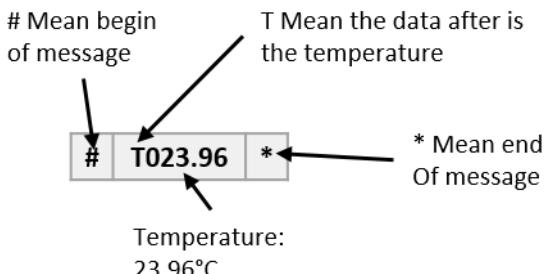
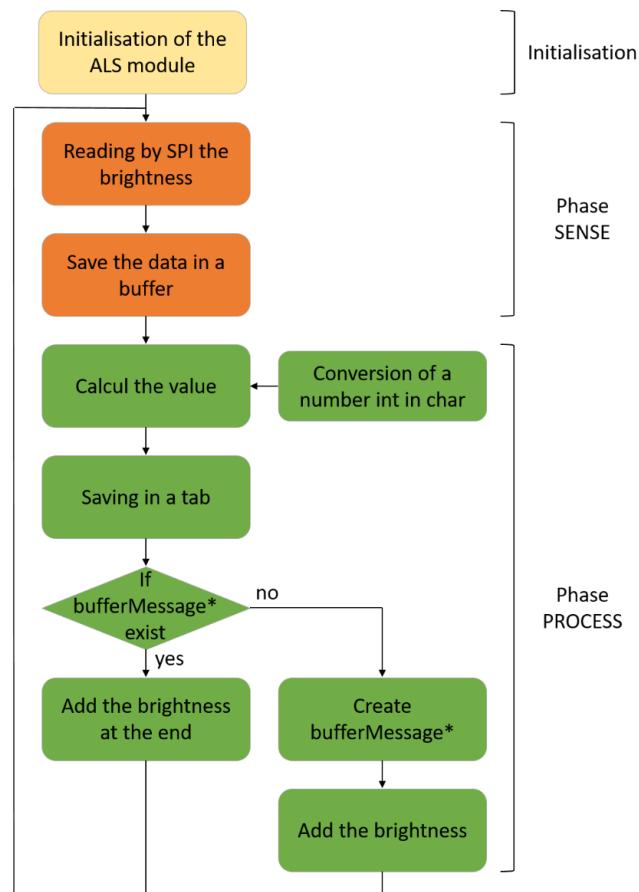


Figure 19 Function Temperature HYGRO frame

ALS

Here is the flow chart of the Pmod ALS, there is a function for the initialization, the phase SENSE of capturing raw data of the sensor and the function of the phase PROCESS which transforms the data.

In all functions of type "processMessage", there is the particularity that if the bufferMessage array is empty, they fill it with their data. Otherwise they add their data at the end of the table. This means that if we interchange the functions between them we will be able to create the frame in the desired order.



*bufferMessage : It's the tab where we save data for send by radio.

Figure 20 Flow chart of the ALS function

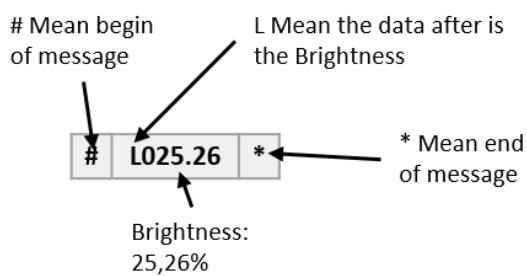


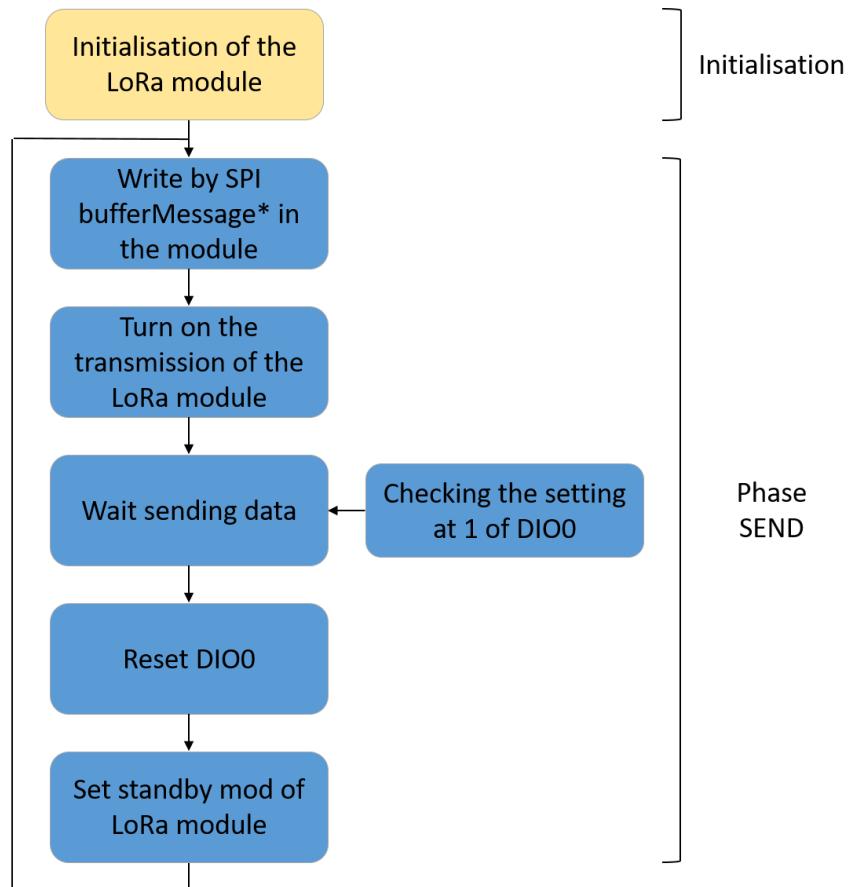
Figure 21 Function ALS frame

LoRa

There are two LoRa functions to run the Pmod:

- The "initLoRa" function initializes the module
- The "sendTabLora1" function that sends the array passed as a parameter.

These two functions are detailed in the flowchart below (Figure 22).



*bufferMessage : It's the tab where we save data for send by radio.

Figure 22 Flow chart of the LoRa function

The message to send in our application is bufferMessage which contains all processed and ordered sensor information.

Smart Sensor

The general flowchart of the code (Figure 23) consists of an initialization phase and 4 periodic phases.

The initialization phase in yellow makes it possible to initialize all the Pmods but also to define the GPIOs, the pins of the communications interfaces, the timers etc ...

The 4 periodic phases are described in the flowchart.

Below is an example of a frame (Figure 24) that the Smart Sensor sends.

The explanations of previous frames allow to decode this frame.

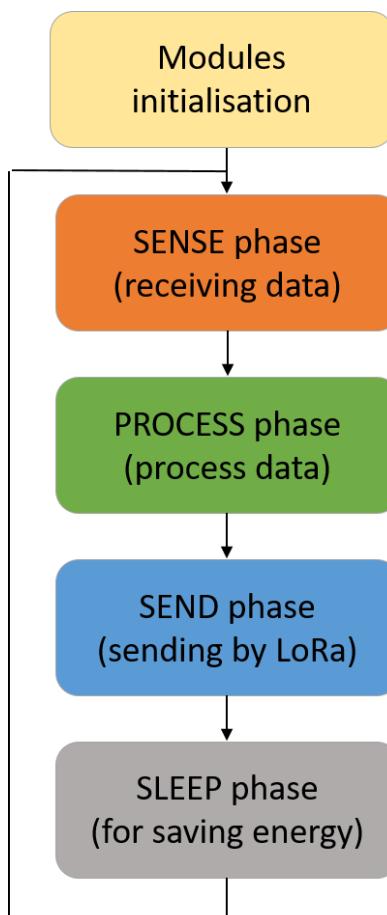


Figure 23 Flow chart of the Smart Sensor

#	H026,10	,	L014,50	,	T023,44	,	H163215	,	LAT4338.1278N	,	LON00350.2516E	,	D210119	*
---	---------	---	---------	---	---------	---	---------	---	---------------	---	----------------	---	---------	---

Figure 24 Frame send by the Smart Sensor

5.5. Result

Flex Node

Here is the final result of the project (Figure 25): we see the CMOD A7, the sensors and the radio module.

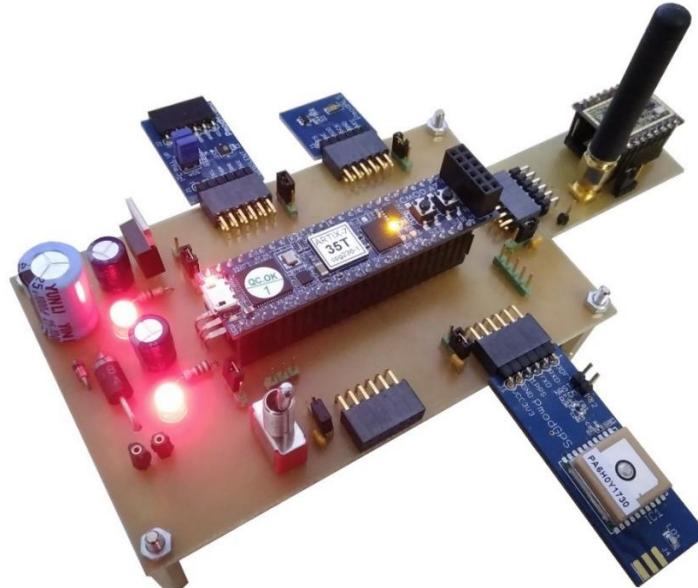
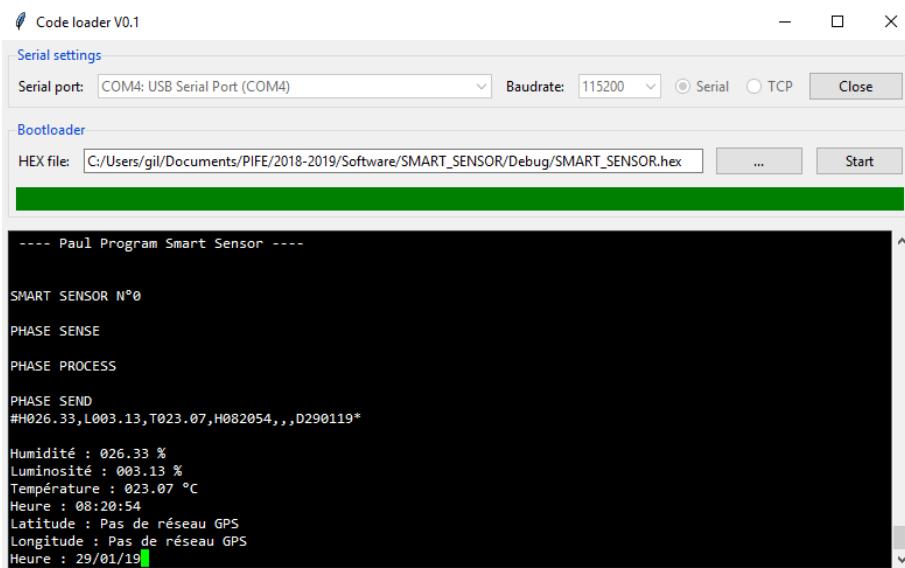


Figure 25 Picture of the Smart Sensor

Terminal

Here is the data displayed by the terminal, it validates the proper operation of the sensors and each step. Just after "PHASE SEND" we can see the frame sent then the sensor data displayed more legibly (Figure 26).



```

---- Paul Program Smart Sensor ----

SMART SENSOR N°0
PHASE SENSE
PHASE PROCESS
PHASE SEND
#H026.33,L003.13,T023.07,H082054,,,D290119*
Humidité : 026.33 %
Luminosité : 003.13 %
Température : 023.07 °C
Heure : 08:20:54
Latitude : Pas de réseau GPS
Longitude : Pas de réseau GPS
Heure : 29/01/19

```

Figure 26 Terminal with print data sensors

Test of consumption

Here is the general consumption of the Smart Sensor (Figure 27). We observe that in these diagrams there are 4 periods of 5 seconds. During the beginning of the period (Phase SENSE, Phase PROCESS and Phase SEND), there is an increase of the power up to 650 mW whereas during the SLEEP Phase the average power is 570mW. The last two curves show that the consumption of the sensors is low ($\approx 130\text{mW}$) compared to the microcontroller implanted in the CMOD A7 board ($\approx 470\text{mW}$).

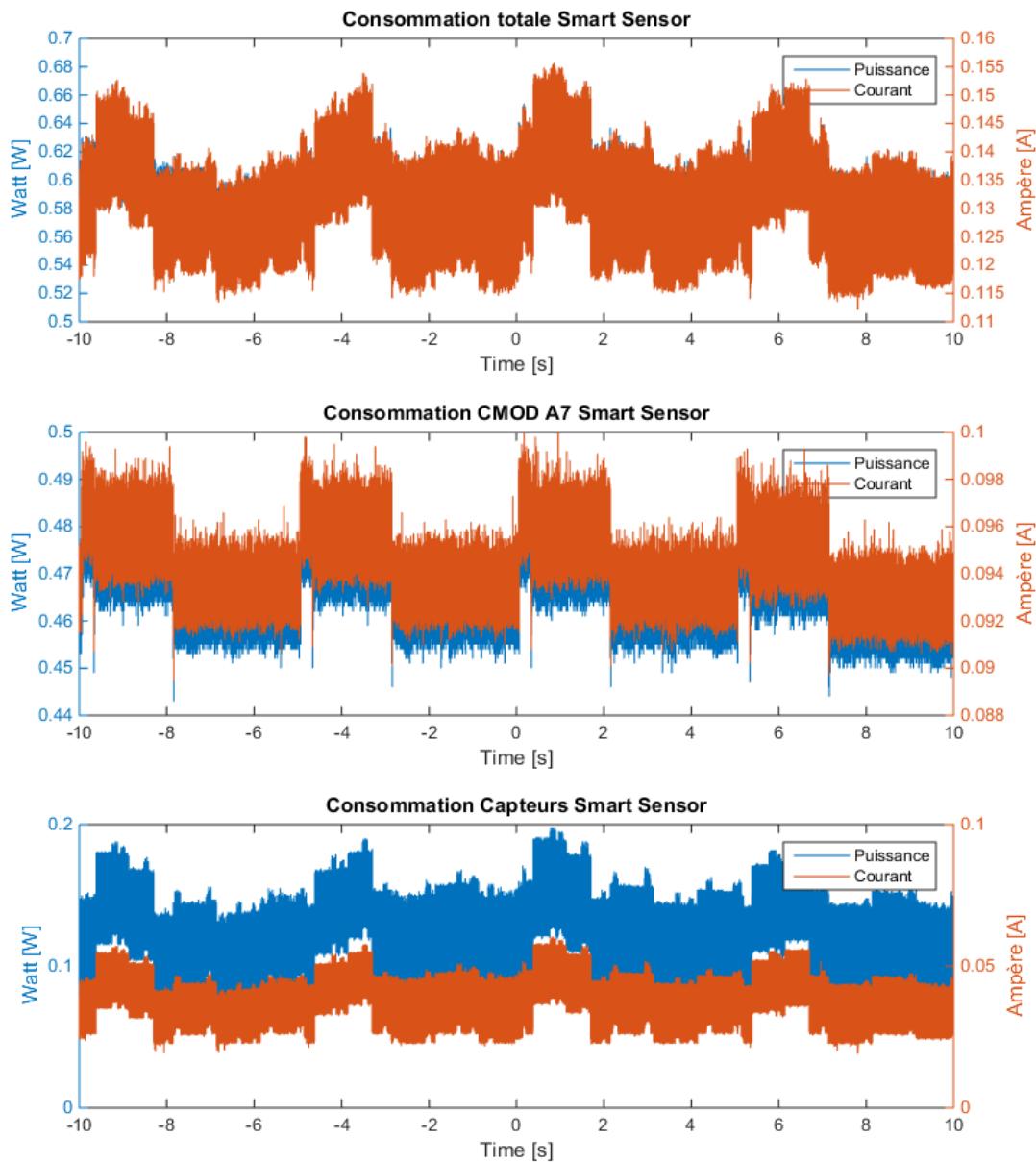


Figure 27 Total consumption, microcontrôleur consumption and sensors (« capteur » in french) consumption

To visualize which sensor consumes the most and when, a measurement of each sensor was made. Thanks to the following 4 curves (Figure 28) we clearly see the SENSE Phase which captures the sensor data on the Pmod HYGRO and Pmod ALS. However the Pmod GPS sends data regularly, every 2 seconds, because it emits data without any requests.

The LoRa of the SEND Phase emits about 1.5 seconds at a power of 28mW.

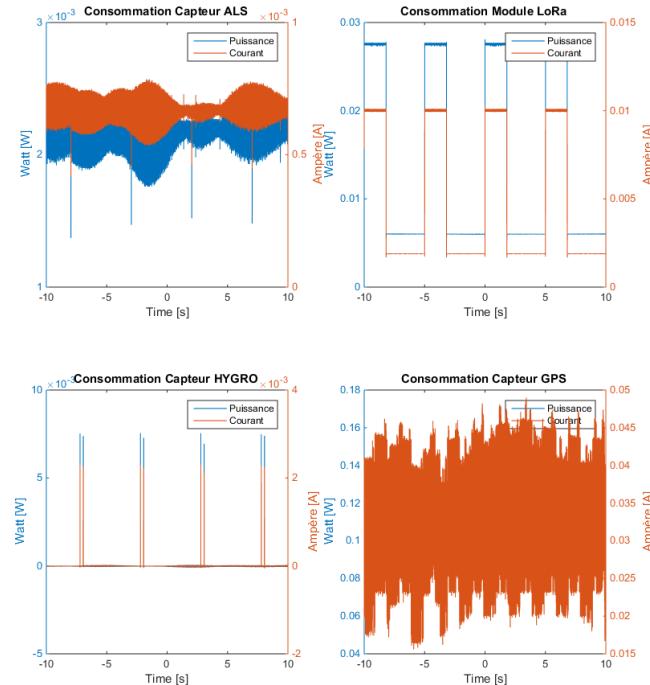


Figure 28 Consumption of each sensors/radio modules

With this chart (Figure 29) with the same scale on each curve, we see that the most important consumption is the CMOD A7. The 4 modules consume very little. In the modules it is the GPS that consumes the most (measurements made without the GPS attached to satellites). Then there is the consumption of LoRa during the sending phase which consume a lot also. The other sensors, HYGRO and ALS, consume very little.

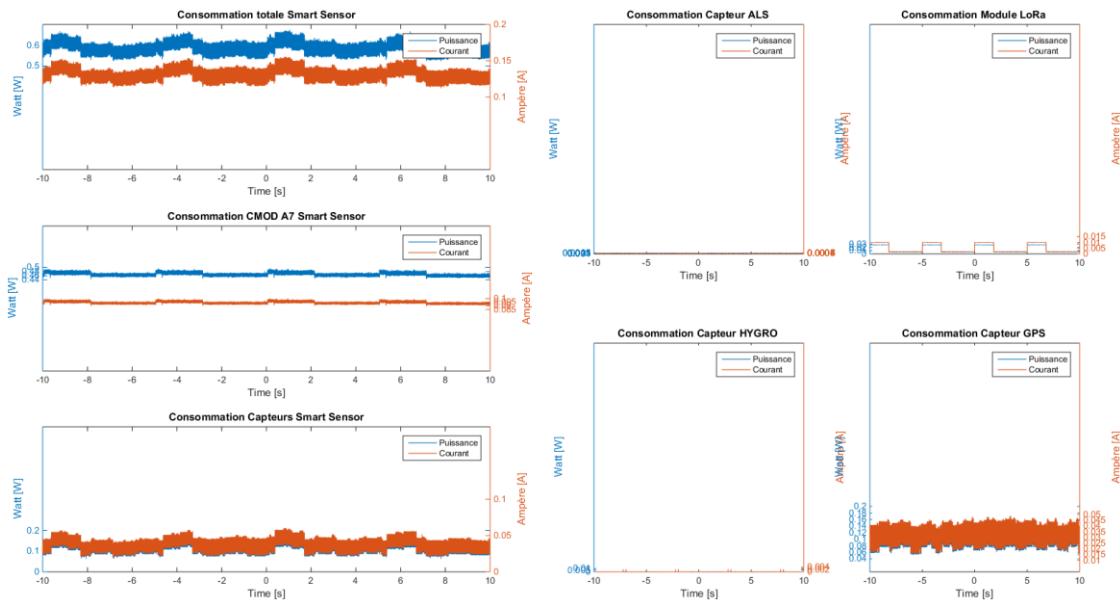


Figure 29 Global view of the consumption

6. Conclusion

This project, led by the LIRMM, is making a significant contribution to the development of smart grid gateways in the IoT domain. It will therefore encourage the integration of cameras in embedded systems networks with several sensors and actuators.

The Smart Sensor is now operational, it meets the specifications and was done on time. Consumption tests show a consistent result. The cost of the Smart Sensor is around 150 € by counting the card and the components.

Indeed, the modules of the Smart camera that perform the image processing acquired (FPGA + ARM processor) and broadcast messages (LoRa), allow a minimal occupation of the network and bandwidth. In addition to these advantages, the Smart Camera guarantees a confidential appearance that makes it preferable applications and areas of use.

The first version of the Smart Camera requires the use of two ZYBO cards, in order to use both parts to perform the tasks of processing and dissemination of information. The next step is to validate the LoRa transmission module and then abandon the second card to keep only one and use its two parts (Programmable System and Programmable Logic).

This industrial project of end of study brought me a lot, either personally or professionally. I was able to discover the laboratory research community. I was able to develop myself personally by working on a project composed of hardware and software.

This experience supported and anchored my knowledge learned during my training. In addition to validating my skills, I was able to develop others such as knowing how to set up an IDE or use communication interfaces other than the UART.

7. Reference

7.1. Smart Gateway Reference

- [1] Ryan Patrick, Nikolaos Bourbakis, "Surveillance Systems for Smart Homes: A Comparative Survey", IEEE Assistive Technologies Research Center, Wright State University, 2009.
- [2] Rickin Patel, Vipul K.Dabhi, Harshadkumar B. Prajapati, "A Survey on IoT based Road Traffic Surveillance and Accident Detection System (A smart way to handle traffic and concerned problems)", International Conference on Innovations in Power and Advanced Computing Technologies, 2017.
- [3] Low-Power Wide Area Network (LPWAN), Les réseaux bas débit pour l'Internet des Objets EFORT, 2015.
- [4] Olga ABDALA, Gabriel MABILLE, Rémy TRIBHOUT, Claire MEYMANDI, "Etude comparative LoRa, Sigfox, GM, UMTS",
- [5] Khoukha ADRAR, "Conception d'une Smart Camera", encadré Pascal BENOIT, Gil THIERRY. LIRMM, 2018.
- [6] M. Santarini, Xilinx unveils vivado design suite for the next decade of 'all programmable' devices, Xcell Journal-Solutions for a Programmable World, no 79, 2012
- [7] Roland Christian Gamom Ngounou Ewo. Déploiement d'applications parallèles sur une architecture distribuée matériellement reconfigurable. Automatique. Université de Cergy Pontoise, 2015. Français.
- [8] Pierre NOWODZIENSKI, "Embarquez votre Intelligence Artificielle (IA) sur CPU, GPU et FPGA" – Application Engineer. The MathWORKS Inc, 2018

7.2. Smart Sensor Reference

- [1] <https://fr.statista.com/statistiques/584481/internet-des-objets-nombre-d-appareils-connectes-dans-le-monde--2020/>
- [2] https://www.iea.org/publications/freepublications/publication/MoreData_LessEnergy.pdf
- [3] <https://www.objetconnecte.net/objets-connectes-energie-101116/>
- [4] <https://store.digilentinc.com/pmod-modules-connectors/>
- [5] <https://fr.farnell.com/>
- [6] <http://www.lirmm.fr/>

8. Appendix

State of the art

Etude & Conception d'un Smart Sensor sur FPGA

Projet Industriel de fin d'étude

AUTEUR : PAUL LELOUP

TUTEUR : PASCAL BENOIT

ENCADRANTS : PASCAL BENOIT & GUILLAUME PATRIGEON

2018



Table des matières

Table des matières	1
1. Introduction	1
2. Gantt	2
3. Capteurs	3
3.1. Définition, types de capteurs.....	3
3.2. Analogique / Numérique	3
4. Communication.....	5
4.1. Réseaux sans fil pour l'IOT.....	5
4.2. Energie.....	5
5. Conclusion.....	6
Références	7

1. Introduction

Le lieu d'étude est le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) qui est une unité mixte de recherche dépendant conjointement de l'Université Montpellier (UM) et du Centre National de la Recherche Scientifique (CNRS) [1].

Le projet a lieu dans le cadre d'un projet industriel de fin d'étude qui a pour objectif de mettre l'élève dans la situation d'un chef de projet chargé de trouver des solutions à un problème concret qui lui est posé [2].

L'internet des objets (IOT) est l'extension d'Internet à des choses et à des lieux du monde physique. L'internet des objets connectés représente les échanges d'informations et de données provenant de dispositifs du monde réel avec le réseau Internet.

L'architecture de l'appareil est présentée sur le schéma ci-dessous (Figure 1). Le projet a pour but de créer la carte électronique qui reliera les capteurs, la carte FPGA et les modules de communication ainsi que de les programmer pour les faire fonctionner ensemble. Cela a pour objectif d'aider une thèse sur l'exploration d'architecture de microcontrôleur afin de baisser la consommation d'énergie.

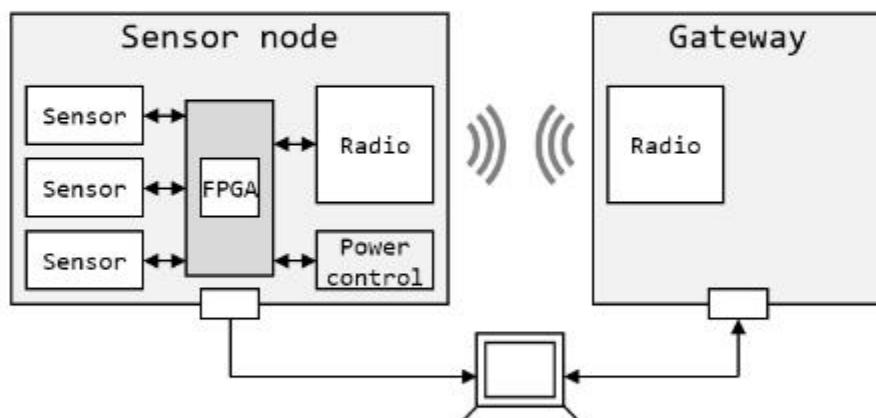
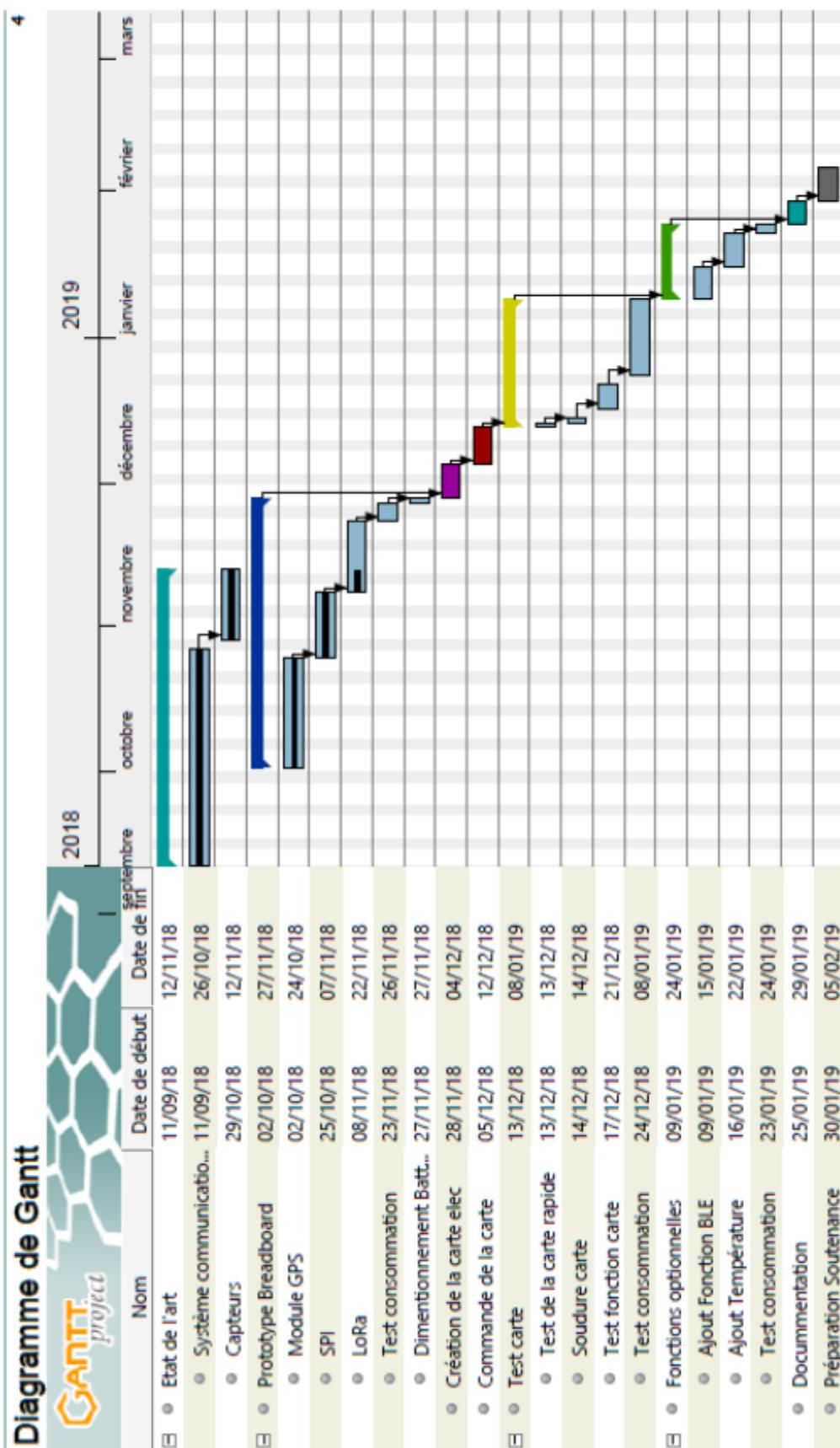


Figure 1 Architecture du Smart Sensor

2. Gantt



3. Capteurs

3.1. Définition, types de capteurs

Un capteur est un dispositif qui fournit à partir d'une grandeur physique prélevée, une autre grandeur physique de nature différente (souvent électrique) [3].

Il y a deux types de capteurs différents. Les capteurs actifs sous l'action du mesurande, génère un signal électrique (courant ou tension), on peut l'appeler générateur. Les capteurs passifs sous l'action du mesurande font apparaître une variation d'impédance. Le capteur passif ne produit pas directement un signal électrique, il sera indispensable d'utiliser un circuit électrique extérieur pour déterminer la valeur du mesurande.

Quelques exemples de capteurs passifs : température, flux optique, déformation, humidité...

Quelques exemples de capteurs actifs : flux optique, force, pression, vitesse...

Le capteur de température utilisé est le PmodTMP2 de Digilent [4] (Figure 3) qui fonctionnera dans les conditions suivantes : Vdd = 3.3V, lecture périodique. Les avantages sont les suivants :

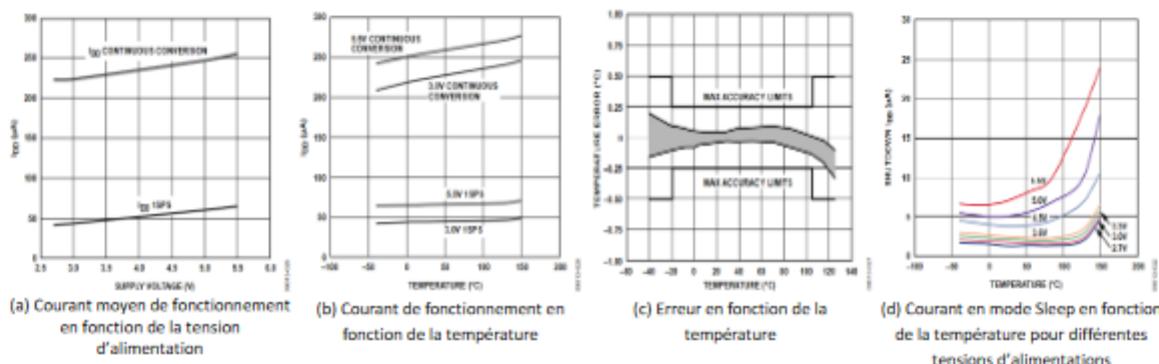


Figure 2 Caractéristique du capteur de température PmodTMP2 de Digilent

Le courant moyen consommé est de $225\mu\text{A}$ en fonctionnement continu (Figure 2 (a)), il est cependant légèrement variable en fonction de la température (Figure 2 (b)). Le capteur est très précis car il a une erreur maximale de 0.5°C entre -20°C et 105°C (Figure 2 (c)). Le mode Sleep consomme seulement $5\mu\text{A}$ maximum (Figure 2 (d)) et permet d'économiser de l'énergie ce qui est très important dans notre application [5].

3.2. Analogique / Numérique

Il faut transformer ces données analogiques en données numériques. On va créer des circuits électriques pour recevoir des valeurs numériques ou utiliser, comme dans la plupart des cas, un microcontrôleur. Il existe beaucoup de capteurs sous forme de module qui délivre la donnée numérique. Capteurs utilisés :

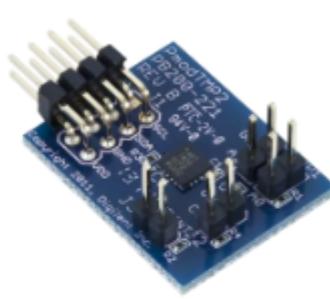


Figure 3 Capteur de température PmodTMP2 de Digilent



Figure 4 Capteur GPS PmodGPS de Digilent

Pour faire l'interface entre le module et le microcontrôleur central, il faut communiquer par des câbles. On va utiliser des interfaces de communications tels que l'UART, l'I²C, le SPI et d'autres encore.

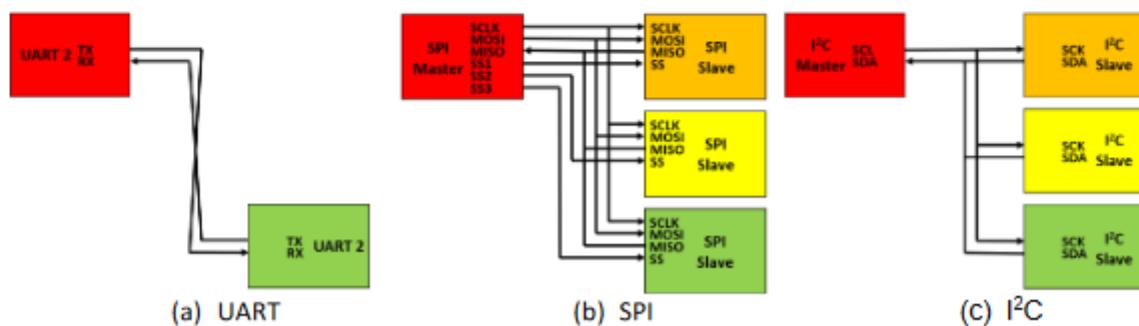


Figure 5 Interfaces de communications

Un UART (Figure 5 (a)), pour Universal Asynchronous Receiver Transmitter, est un émetteur-récepteur asynchrone universel.

Une liaison SPI (Figure 5 (b)) (pour Serial Peripheral Interface) est un bus de données série synchrone. Les circuits communiquent selon un schéma maître-esclaves, où le maître contrôle la communication. Plusieurs esclaves peuvent coexister sur un même bus, dans ce cas, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée Slave Select.

I²C (Figure 5 (c)) est un bus série synchrone bidirectionnel half-duplex, où plusieurs équipements, maîtres ou esclaves, peuvent être connectés au bus.

Les UART, SPI et I²C sont généralement intégrés dans des composants comme des microcontrôleurs. Ils ne sont dans ce cas plus un composant à proprement parler, mais une fonction périphérique du composant central.

4. Communication

4.1. Réseaux sans fil pour l'IOT

Pour l'IOT il est important que le plus d'appareils soient reliés pour avoir plus d'informations et agir de la meilleure manière possible. L'intérêt du réseau sans fil est qu'il est pratique (pas de fil) et que l'installation d'un nouvel appareil sera rapide et simple. Voici quelques réseaux sans fil adapté à l'IOT qui vont être comparés au niveau de la consommation énergétique :

Le **SIGFOX** est une technologie UNB (Ultra Narrow Band) qui opère à 868 MHz avec un débit de 1000 b/s pour une bande passante de 1 KHz et une sensibilité de -140 dBm avec une portée annoncée de 40 km. Les appareils utilisant cette technologie peuvent envoyer 140 messages par jour à une station. C'est une technologie asynchrone, donc les nœuds ne se réveillent pas pour se synchroniser. Il est donc plus simple de calculer la consommation [6].

LoRa est faite pour les applications IOT à longue portée. LoRa Alliance propose une station qui reçoit des paquets depuis un appareil et retransmet les données à un serveur par connexion TCP. Le LoRa utilise principalement de la modulation basée sur le CSS (Chirp Spread Spectrum) qui permet un débit de 0,25 à 11 kb/s avec une bande passante de 7 à 250 KHz. Dans cette comparaison seule LoRa classe A sera pris en compte [6].

Le **Bluetooth Low Energy** est une version du Bluetooth standard. Il est le plus souvent utilisé pour des applications tel que des moniteurs de fréquence cardiaque ou des contrôles à distance de température d'une pièce. L'avantage de cette technologie est qu'elle est très populaire (dans la majorité des smartphones) et que les dernières versions (4.2, 5.0) permettent d'avoir plusieurs nœuds. BLE a une fréquence d'utilisation de 2,4 GHz, le débit maximal est de 2 Mb/s. Sur la dernière version 5.0, on peut utiliser 4 schémas différents qui permettent d'avoir plusieurs débit (125 kb/s, 500 kb/s, 1 Mb/s et 2 Mb/s) pour moins consommer en fonction des applications [6].

La technologie **NB-IoT** (Narrowband Internet of Things) est une technologie de communication sans fil qui fonctionne à 180KHz avec un débit de 250 kb/s. Appliquée à ce jour principalement au secteur de l'industrie, le NB-IoT, porté par la Chine, souhaite conquérir maintenant celui de la Smart Home, du Smart Building et de la Smart City. Les atouts de cette technologie sont nombreux : faible consommation d'énergie, couverture étendue et faible coût de production et d'utilisation [7].

4.2. Energie

L'IOT est intéressant par le fait de son autonomie et de sa source d'énergie sans fil. Les résultats présentés se basent sur un article scientifique : *Comparison of the Device Lifetime in Wireless Networks for the Internet of Things* [6].

Les calculs de la durée de vie des objets connectés sont faits à partir des valeurs des variables de ce schéma (Figure 6).

t_a signifie la période d'application

t_{on} signifie la période d'émission

t_{cl} signifie la période de la synchronisation

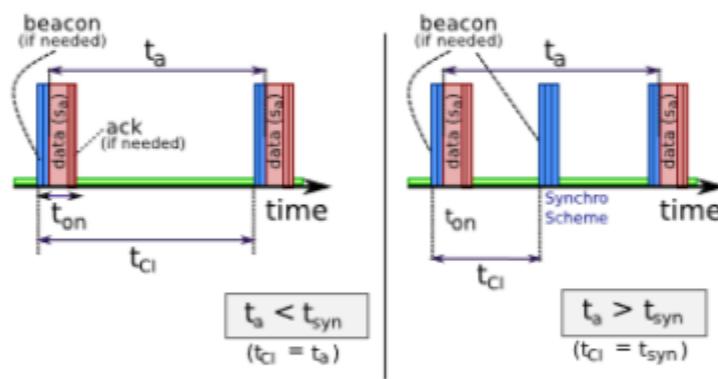


Figure 6 Calcul de consommation

Les auteurs de l'article ont créé l'algorithme qui va traiter ces valeurs et nous donner la consommation.

Voici la consommation de chaque mode (Figure 7) (il ne faut pas prendre en compte uniquement ce tableau car chaque protocole utilisera pendant des temps plus ou moins long ces modes) :

Power	P_Tx	P_Rx	P_Idle	P_Sleep
SIGFOX	147mW	39mW	Ø	4.32uW
LoRa	419.6mW	44mW	Ø	4.32uW
BLE	24.11mW	9.26mW	4.67mW	3.24uW
NB-IOT	Ø	Ø	Ø	Ø

Figure 7 Récapitulatif des consommations

NB-IOT n'a pas de valeur dans le tableau car ce système de communication n'était pas dans l'article étudié et prendre des valeurs d'autres études ne serait pas objectif étant donné que les conditions de test ne seraient pas les mêmes.

Il y a un autre algorithme qui permet de calculer la consommation en fonction de la probabilité de paquet perdu que l'on doit renvoyer.

5. Conclusion

L'enjeu majeur de l'IOT est l'autonomie, d'où la préférence d'utiliser des capteurs actifs (qui génèrent une tension ou un courant). L'intérêt est de faire de l'exploration d'architecture sur FPGA de microcontrôleur pour baisser la consommation.

Voici les résultats de mes recherches sur la consommation des protocoles radio (rappel : sauf NB-IOT) :

- Lorsque les paquets ne sont pas perdus :

Pour une émission tous les jours, toute les 100s, toute les secondes et millisecondes le BLE est le plus efficace en termes d'économie d'énergie.

- Lorsque 20 % des paquets sont perdus :

Le BLE reste plus économique en énergie mais la durée de vie diminue d'environ 10 %.

- La dérive d'horloge :

Elle ne fait que très peu baisser la durée de vie du LoRa et du SIGFOX à une faible intensité de trafic, contrairement au BLE.

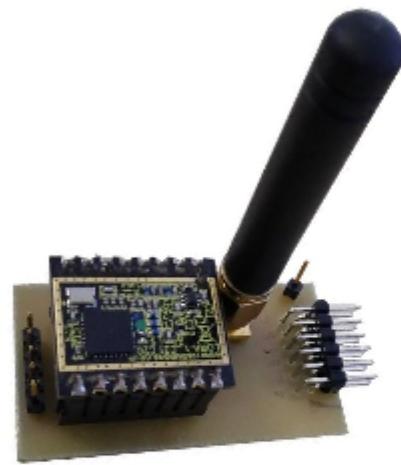
Pour notre application on va donc préférer le LoRa car il consomme finalement assez peu mais a une portée bien plus grande que le BLE. On va préférer utiliser des capteurs Pmod pour faciliter les tests de différents capteurs lorsque la carte sera terminée. Des capteurs tels que la température, la localisation vont être utilisés afin d'avoir des données pertinentes à transmettre.

L'utilisation du LoRa est intéressante pour le LIRMM car elle permettra d'obtenir des données sur cette technologie récente.

Datasheet Pmod LoRa

LoRa Pmod V2.0

Pmod designed by Paul LELOUP



Additional features

- Integrated RF-LoRa-868 of RF Solutions
- A Pmod Connector for plug and play (using FPGA card for prototype)
- An SMA connector for the antenna
- Small card dimensions: 27.9mm x 50.9 mm
- 4 connectors for measure current and tension of the modules
- 5 connectors for use optional PINS of the RF-LoRa-868

RF-LoRa-868 Features

- Up to 16KM Range
- Integrated LoRa Modem Semtech SX1272
- Highly Efficient Integral Impedance Matching Network
- Provides Full Functionality of the RFIC:
- 157 dB maximum link budget
- +20 dBm at 100mW constant RF output vs. V supply
- -14 dBm high efficiency PA
- Built in RF switch
- High sensitivity: down to -130 dBm
- Bullet-proof front end: IIP3 = -12.5 dBm
- 89 dB blocking immunity
- Programmable bit rate up to 300kbps
- Low RX current of 10 mA, 100nA register retention
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation
- Built-in bit synchronizer for clock recovery
- Preamble detection
- 127 dB Dynamic Range RSSI
- Automatic RF Sense and CAD with ultra-fast AFC
- Packet engine up to 256 bytes with CRC
- Built-in temperature sensor and low battery indicator

Application

- Prototype with FPGA
- Smart Sensor
- Home Automation
- RF Alarms
- Sensor networks
- Long Range Telemetry
- Meter Reading
- Irrigation Systems
- Wireless Applications

Part Numbers

Pmod connector

PIN	Definition	Direction	Function
1	RESET	In	Reset Trigger Input
2	DIO0	In/Out	Digital I/O software configured
3	TX_SWITCH	In	Enable TX RF Path Active High
4	RX_SWITCH	In	Enable RX RF Path Active High
5	GND	-	Ground connection
6	VCC	In	Power connection
7	SCLK	In	SPI Serial Clock Input
8	NSEL	In	Device Select Active Low
9	SDI	In	SPI Serial Data Input
10	SDO	Out	SPI Serial Data Output
11	GND	-	Ground connection
12	VCC	In	Power connection

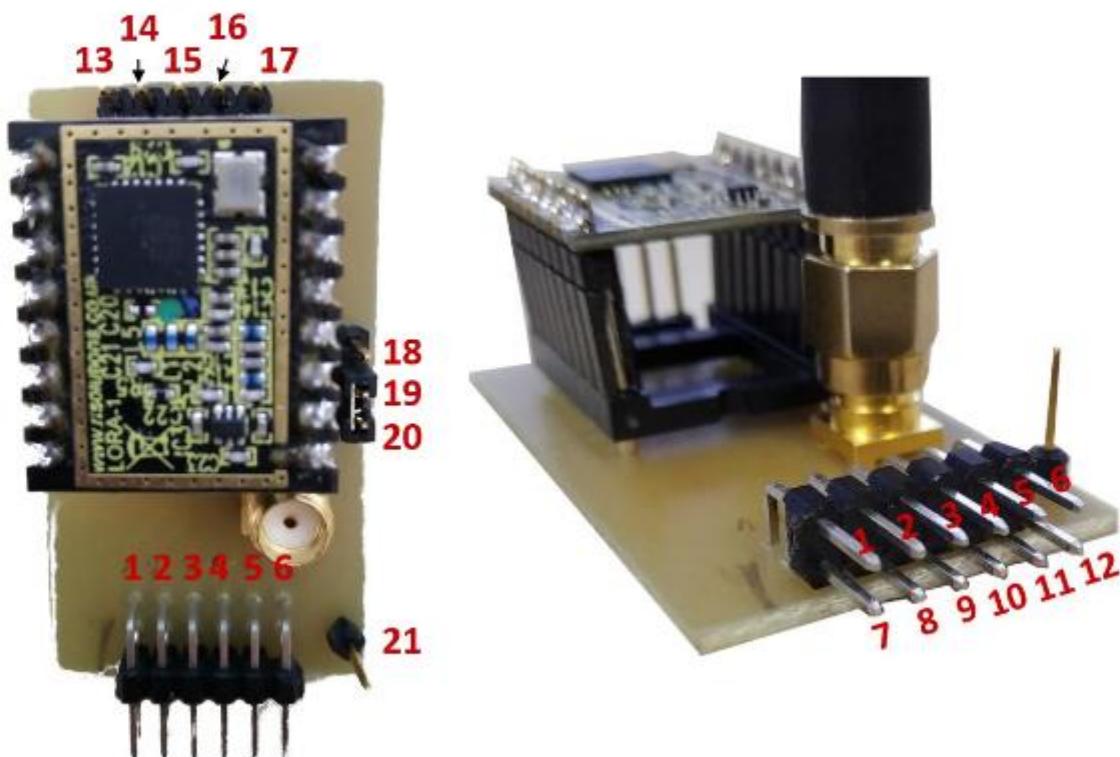
5 Pins connector

PIN	Definition	Direction	Function
13	DIO3	In/Out	Digital I/O software configured
14	DIO4	In/Out	Digital I/O software configured
15	DIO5	In/Out	Digital I/O software configured
16	DIO1	In/Out	Digital I/O software configured
17	DIO2	In/Out	Digital I/O software configured

4 Pins connector

PIN	Definition	Direction	Function
18	VCC Module	-	Pin for tension measure
19	VCC Module	-	Pin for current measure
20	VCC Pmod	-	Pin for current measure
21	GND	-	Pin for tension measure

Pin Description



Using the Pmod

First you can plug the RF-LoRa-868 like this in the Pmod card:



Then for turn ON the alimentation of the Pmod, you must put a bridge between the Pin 19 and 20.
Now there is 3.3V to the module.

It's highly recommended to put an SMA antenna (868 MHz).

Go on your keyboards, you can use the Pmod!

Schematic

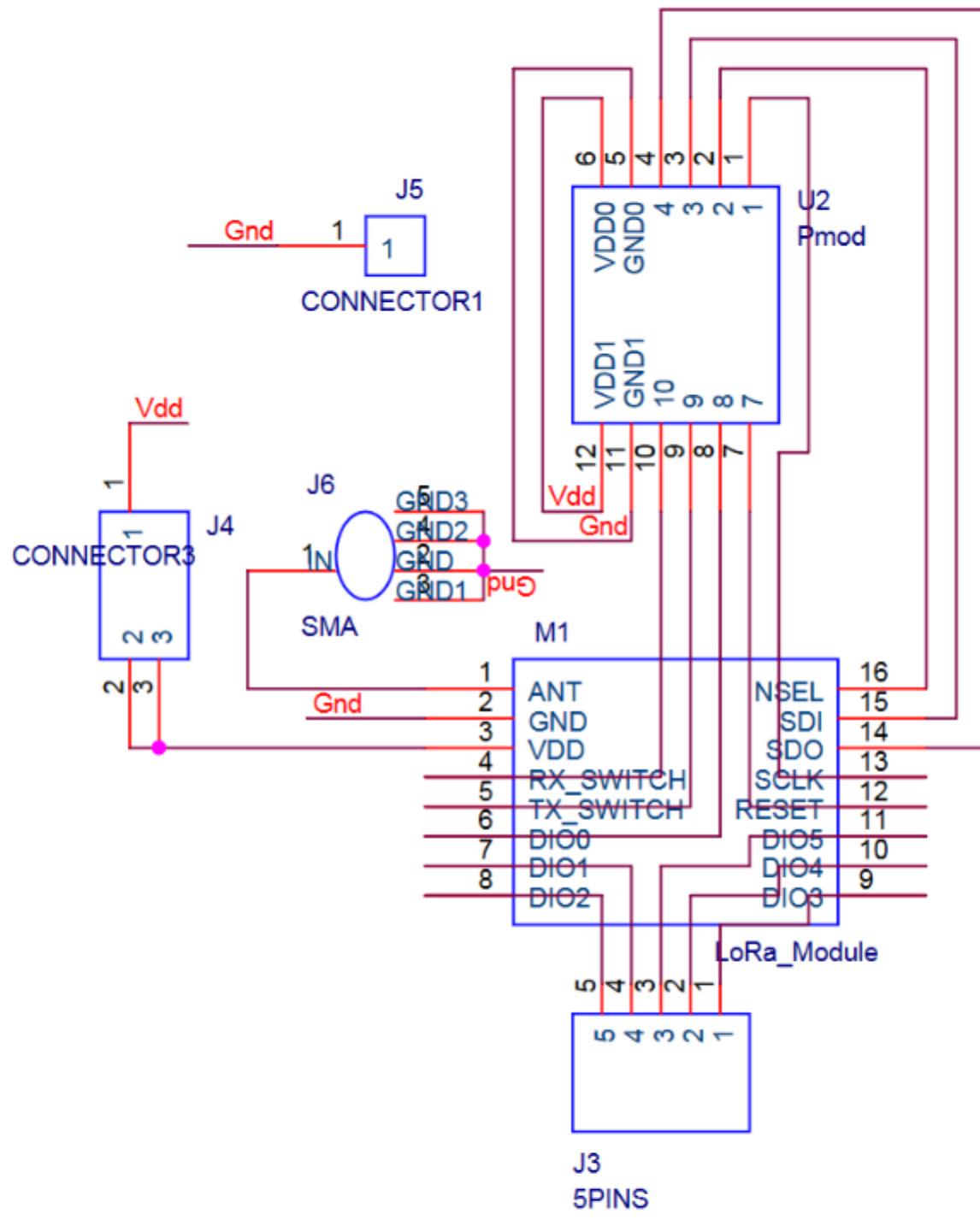


Figure 1 Schematic of Pmod v2.0

Board

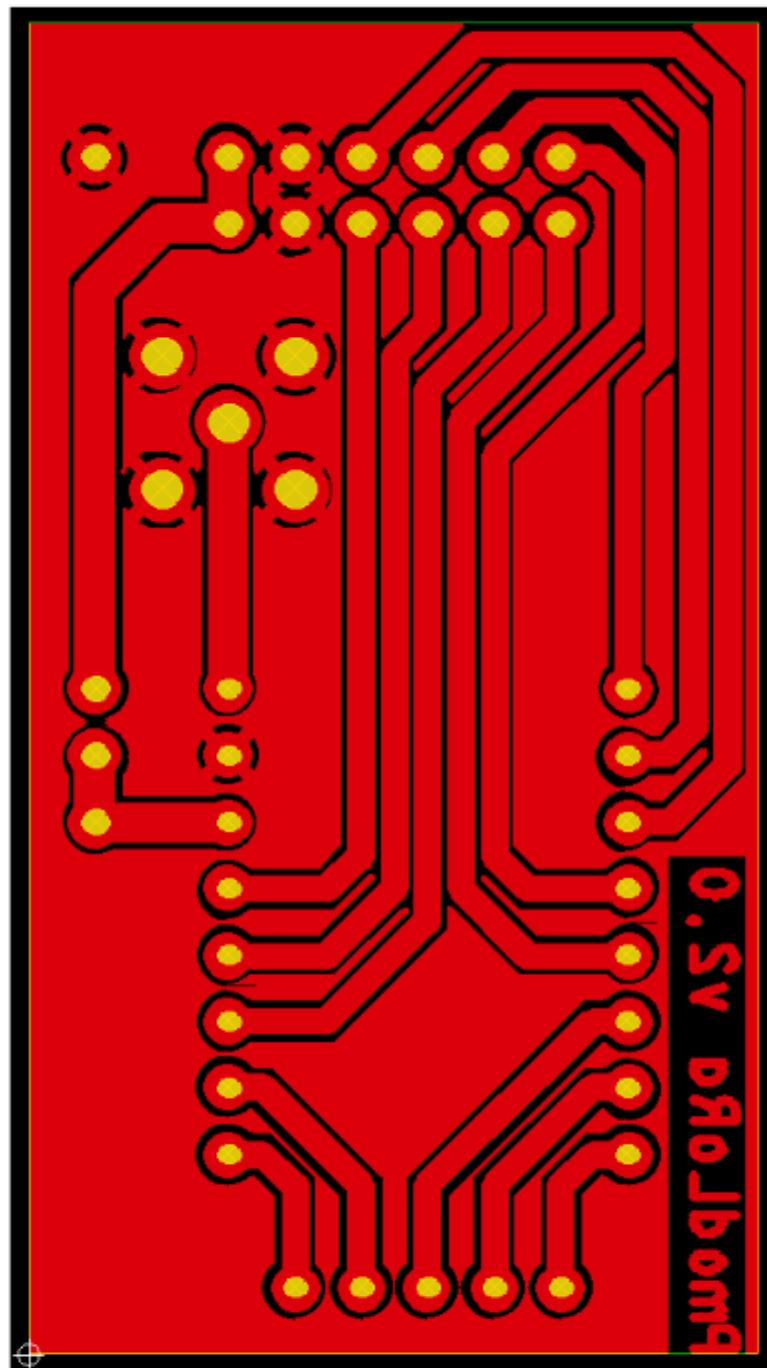


Figure 2 Bottom view Board of Pmod v2.0

References

- Datasheet RF-LORA-868
http://www.farnell.com/datasheets/2162976.pdf?_ga=2.254649036.578254260.1548165701-413366784.1537950013
- Datasheet SX1272/73
<https://www.semtech.com/uploads/documents/sx1272.pdf>

Tutoriel decouverte

Tutoriel de prise en main du CmodA7 FPGA

Etude & Conception d'un Smart Sensor sur FPGA

AUTEUR : PAUL LELOUUP

TUTEUR : PASCAL BENOIT

THESE DE : GUILLAUME PATRIGEON

2019



Table des matières

Table des matières	1
1. Prérequis	2
1.1. Matériel.....	2
1.2. Logiciel.....	2
2. Tutoriel.....	3
2.1. Mettre en place l'IDE, le Bootload	3
Installer JAVA.....	3
Installation IDE Eclipse.....	3
Paramètres du projet (IDE Eclipse)	7
Installation du Codeloader	13
Utilisation du Codeloader	14
Faire clignoter la LED 2	16
2.2. Utiliser l'interface UART avec le module GPS	16
Initialiser l'UART2	16
Les afficher	16
Les stocker	16
2.3. Mettre sous un bon format	16
Machine d'état	16
Créer une trame	16
3. Pour approfondir	16
3.1. Utiliser les autres interfaces du CmodA7, SPI et I2C	16
3.2. Envoyer la trame par LoRa.....	16

1. Prérequis

1.1. Matériel

Pour le matériel vous aurez besoin de :

- Un ordinateur
- Un FPGA CmodA7 de DIGILENT
- Un module Pmod GPS de DIGILENT
- Un câble USB-microUSB

1.2. Logiciel

Au niveau du logiciel vous aurez besoin de :

- Un IDE, Eclipse CDT :
 - Lien vers le site : <https://www.eclipse.org/cdt/downloads.php>
 - Téléchargement d'Eclipse version 2018.2 :
<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2018-12/R/eclipse-inst-win64.exe>
- Le compilateur gcc-arm-none-eabi :
 - Lien vers le site :
<https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads>
Télécharger la version « 7-2017-q4-major » et cliquer sur « window zip »
L'extraire et le déplacer dans un dossier à ne pas supprimer.
- Python 3 :
 - Lien vers le site : <https://www.python.org/downloads/>
 - Téléchargement de Python 3.7.2 :
<https://www.python.org/ftp/python/3.7.2/python-3.7.2.exe>
- Java JRE :
 - Lien vers le téléchargement
<https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
Accepter les droits
Aller dans « Java SE Runtime Environment 8u201 » et cliquer sur la version « x64 » en « .exe »
- La librairie du cortex M0 est présente dans l'archive du tutoriel
- Les fichiers sources sont présents dans l'archive du tutoriel
- Les fichiers *linker.ld* et *startup.asm* sont présent dans l'archive du tutoriel
- L'outil Codeloader V0.1 en python est présent dans l'archive du tutoriel
- Le datasheet du CmodA7 pour avoir les PINS est présent dans l'archive du tutoriel

2. Tutoriel

2.1. Mettre en place l'IDE, le Bootload

Installer JAVA

Double cliquer sur le .exe

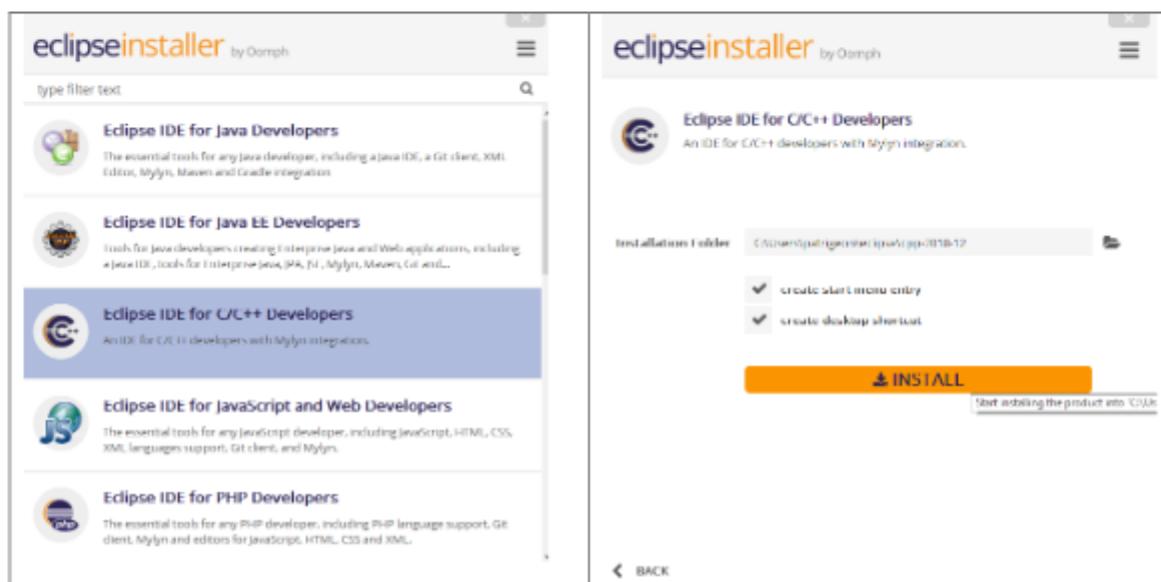
Cliquer sur installer

Cliquer sur fermer

Java est maintenant installé

Installation IDE Eclipse

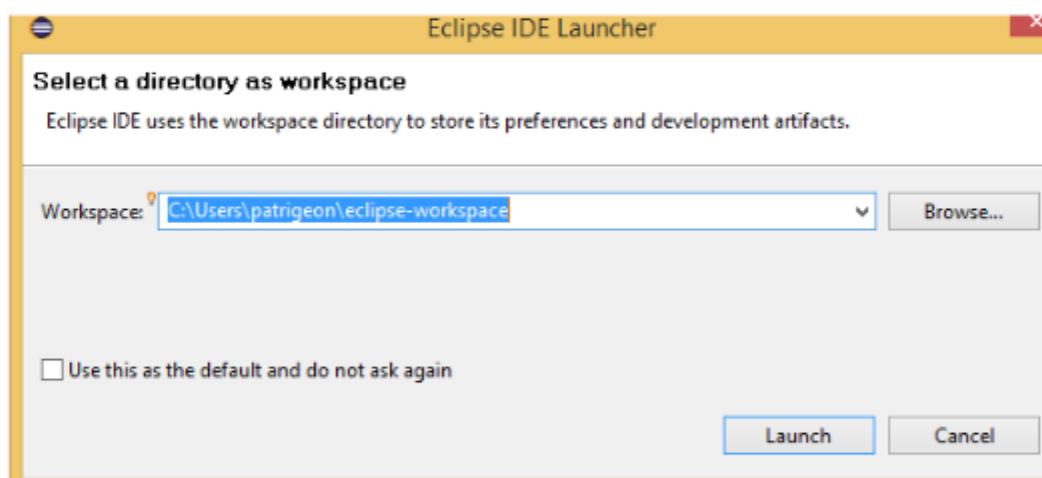
Double cliquer sur le .exe



Accepter toutes les conditions

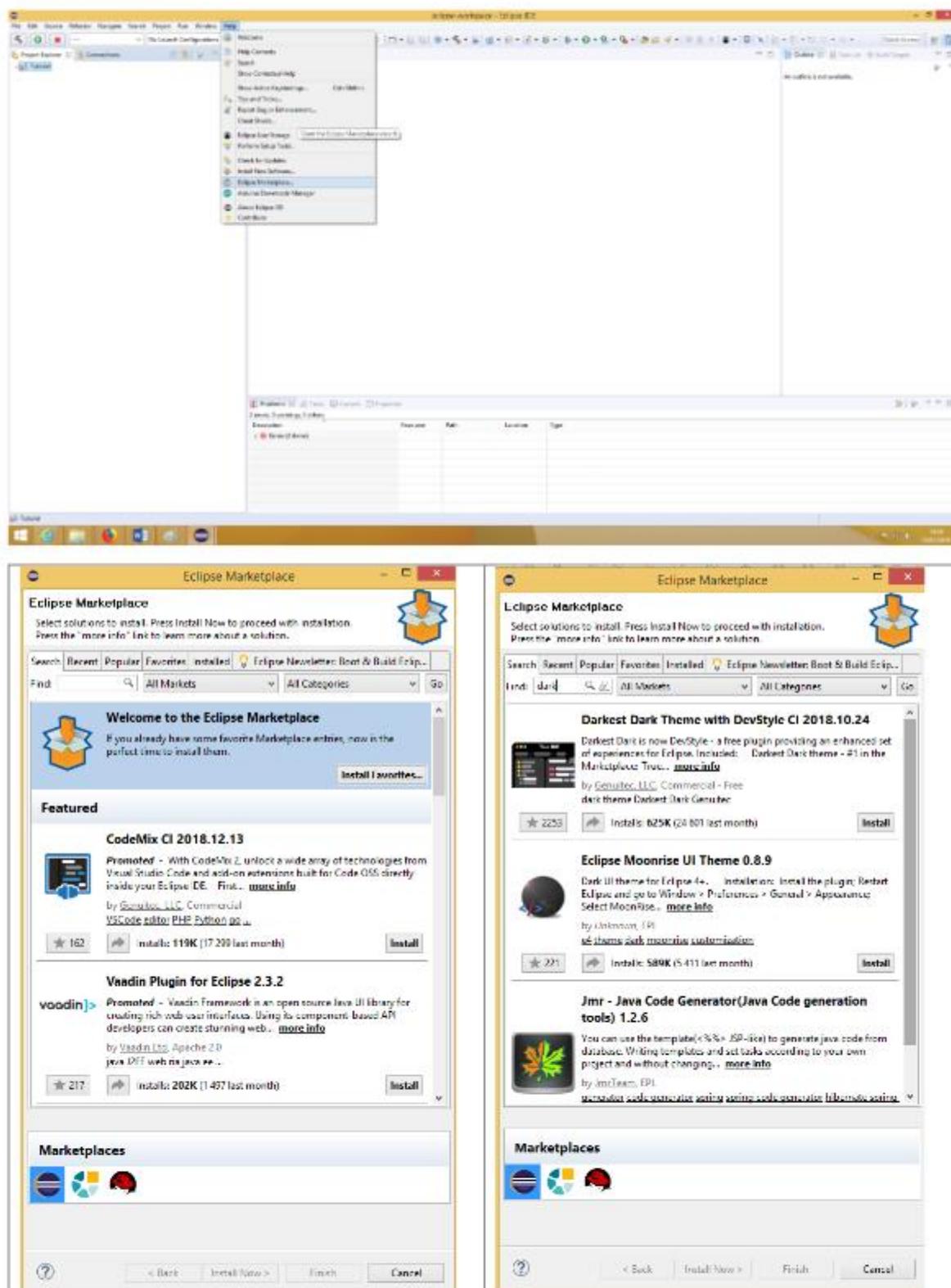
Définir le dossier où tous les projets seront enregistrés :

(Bien conserver le chemin qui mène aux projets)



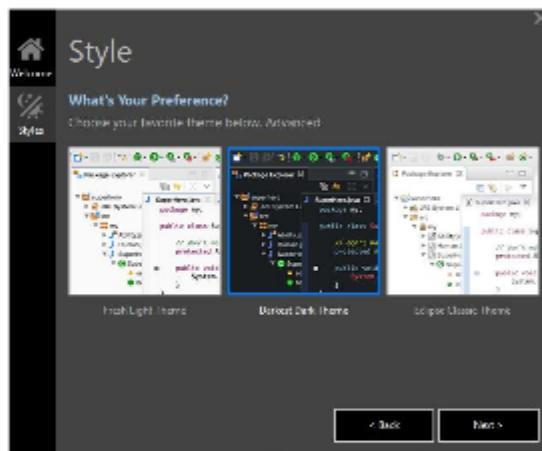
Pour le Dark theme ;

Help/marketplace



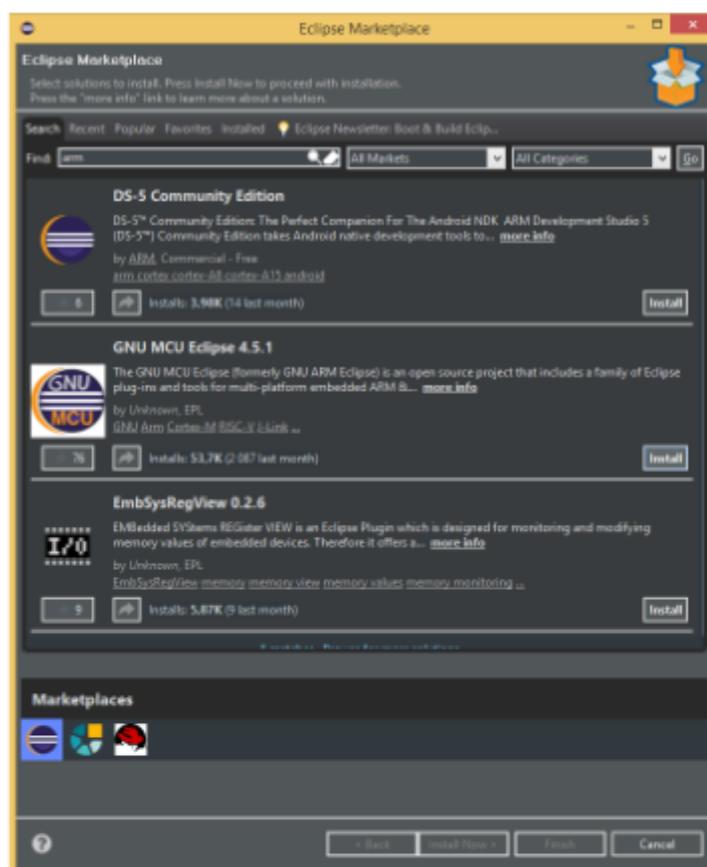
Confirmer les conditions

Restart

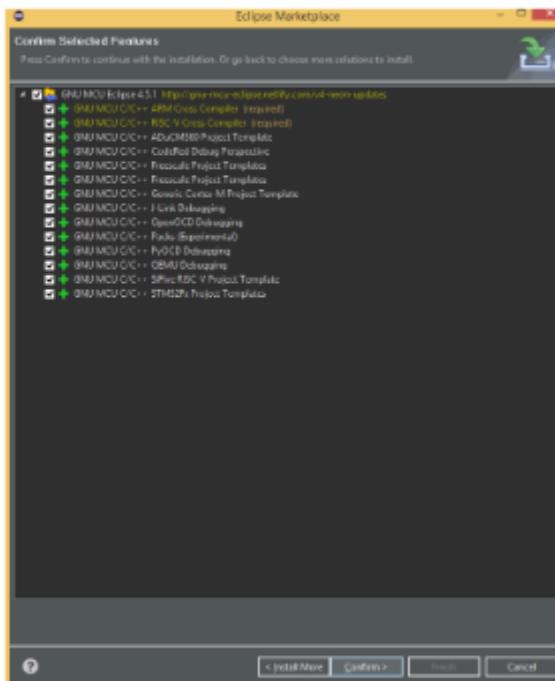


Installer GNU ARM ;

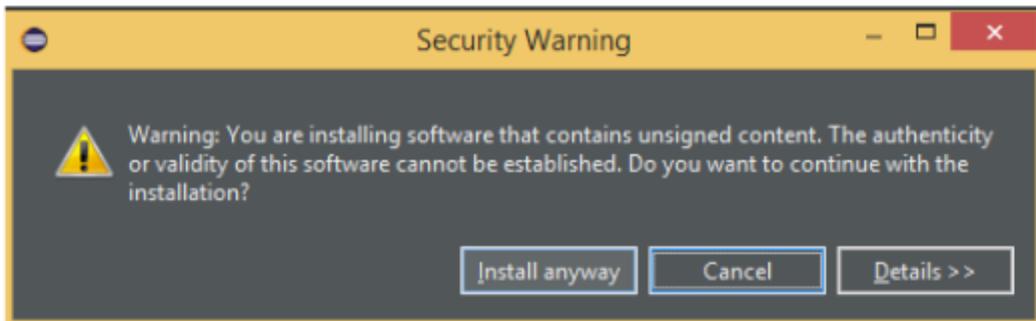
Help/marketplace



Confirmer les conditions



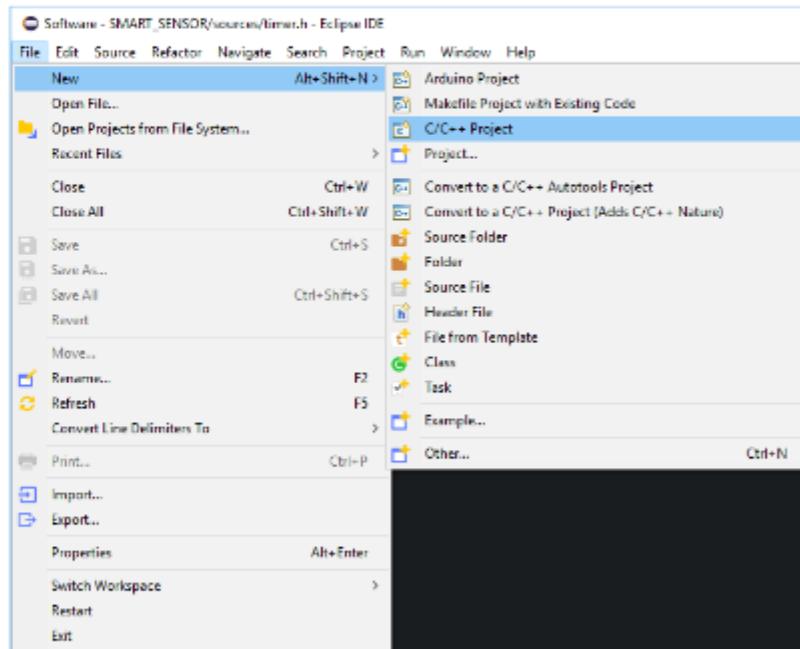
Install anyway



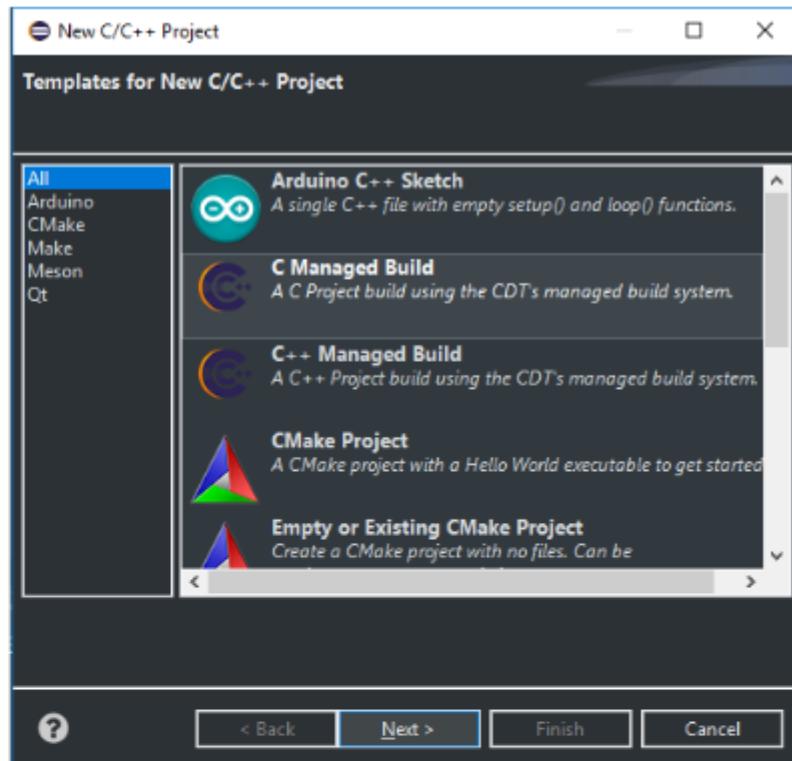
Restart

Paramètres du projet (IDE Eclipse)

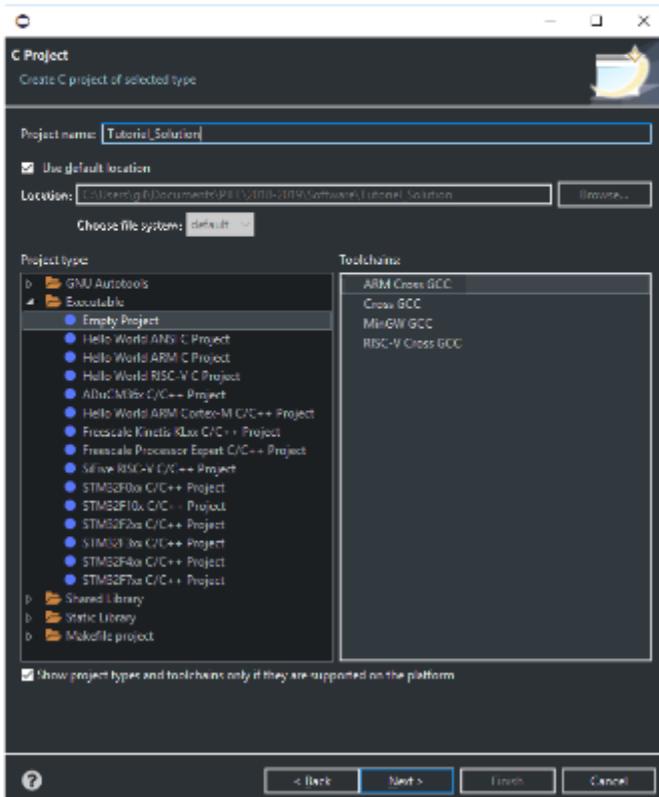
Faire un nouveau projet C/C++



Sélectionner « C Managed Build »



Choisir le nom du projet, sélectionner ARM Cross GCC et cliquer sur « Next »



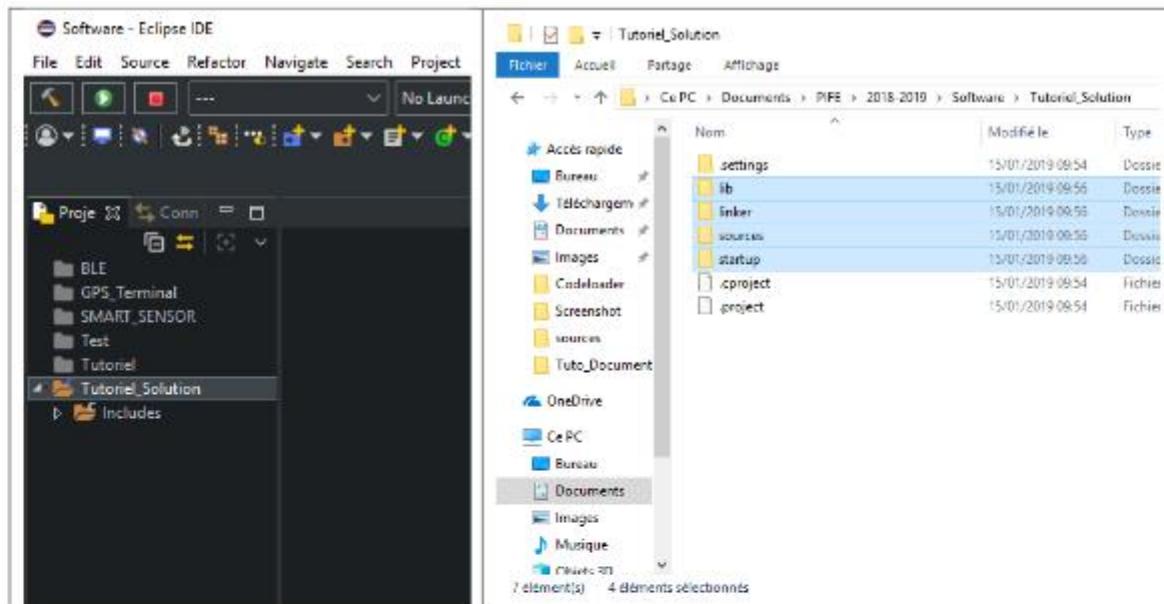
Puis cliquer encore sur « Next »



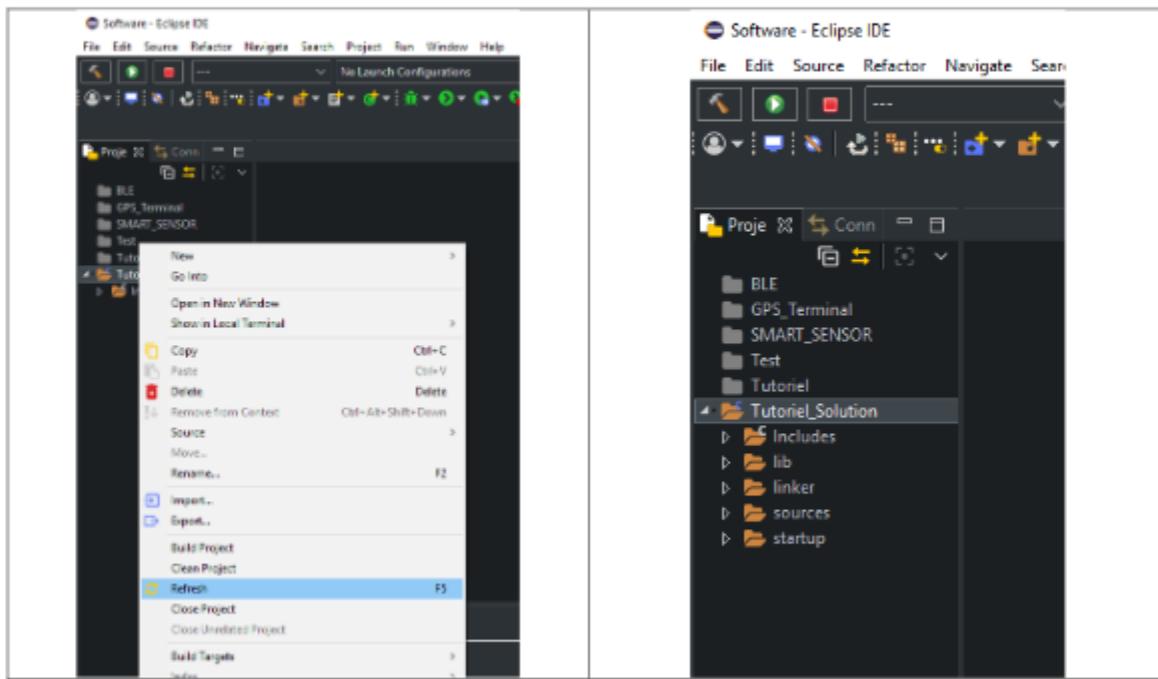
Pour le GNU ARM Cross Toolchain, sélectionner le path de gcc-arm-none-eabi-7-2017-q4-major-win32 préalablement téléchargé.



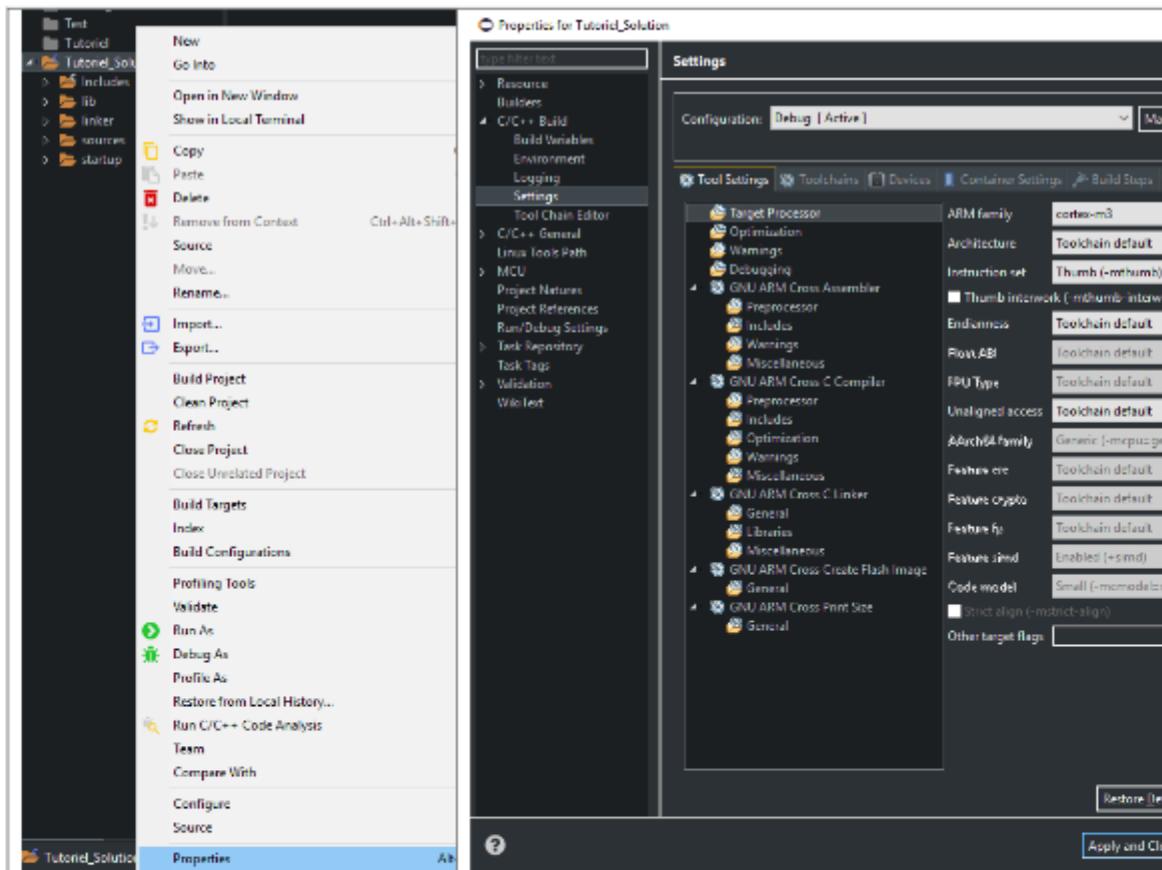
Le dossier compressé « Tutoriel_vX.X » qui vous a été envoyé par mail contient le dossier « Software ». A l'intérieur de celui-ci, vous trouverez 4 dossiers : les fichiers sources, la librairie, le linker et le startup. Ensuite copier/coller ces dossiers dans le dossier de votre projet



Clic droit sur le projet dans Eclipse et cliquer sur Refresh.

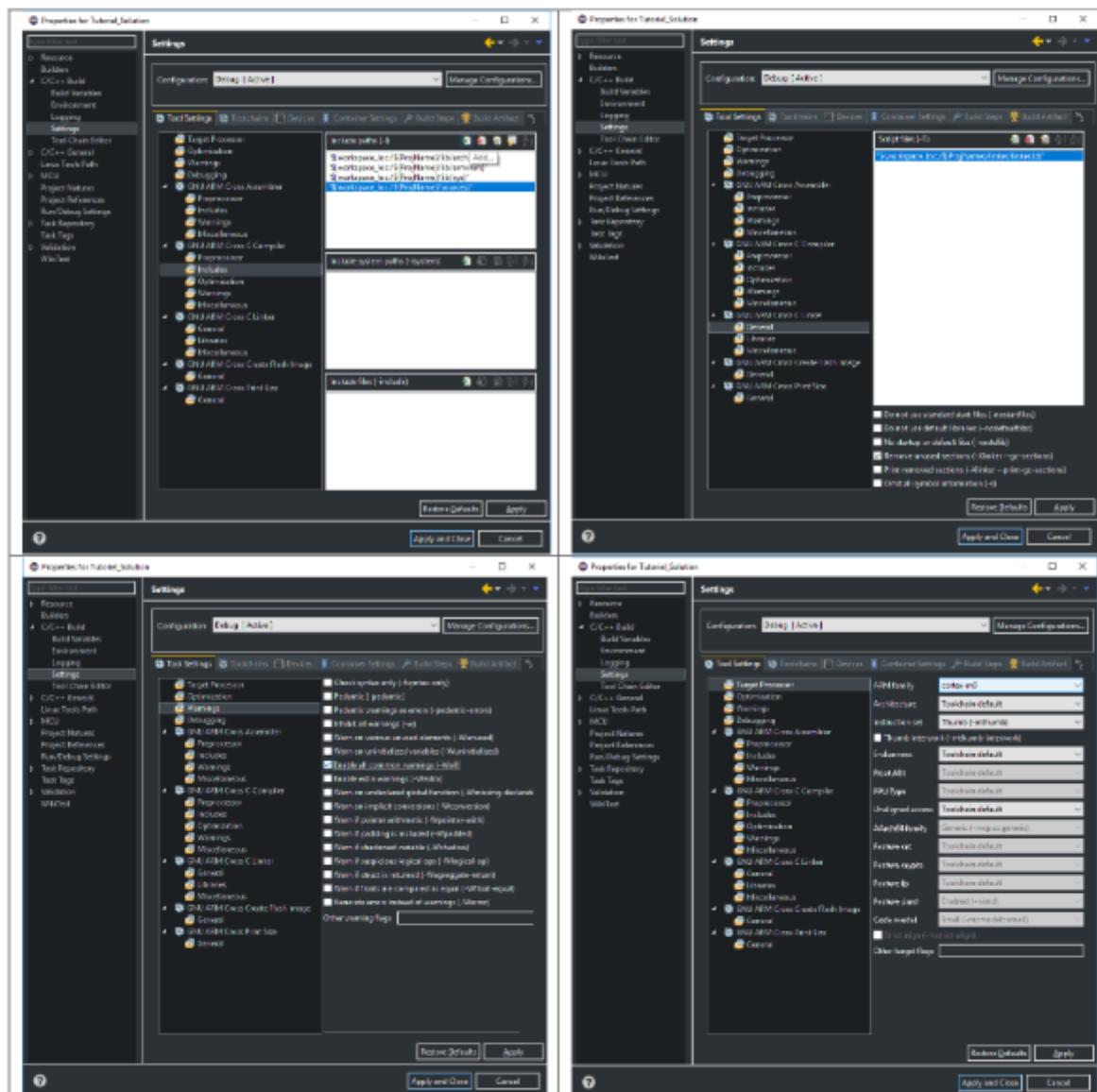


Dans l'IDE, sélectionner le projet et aller dans les Properties :



- Properties/C/C++ Build/Settings/Tool Settings/GNU ARM Cross C Compiler/Includes/Include paths
- Ajouter le dossier source et les 3 dossier librairies.
- Properties/C/C++ Build/Settings/Tool Settings/GNU ARM Cross C Linker/General
- Ajouter le fichier linker
- Properties/C/C++ Build/Settings/Tool Settings/Warnings
- Cocher la case « Enable all common warnings (-Wall) »
- Properties/C/C++ Build/Settings/Tool Settings/Target Processor
- Selectionner le cortex-m0 dans ARM family

Voici le résultat attendu :



Compiler le projet en cliquant sur le marteau en haut à gauche :

The screenshot shows the Eclipse IDE interface. In the top-left, there's a code editor with C code for a timer interrupt. In the bottom-right, a terminal window displays the build process:

```

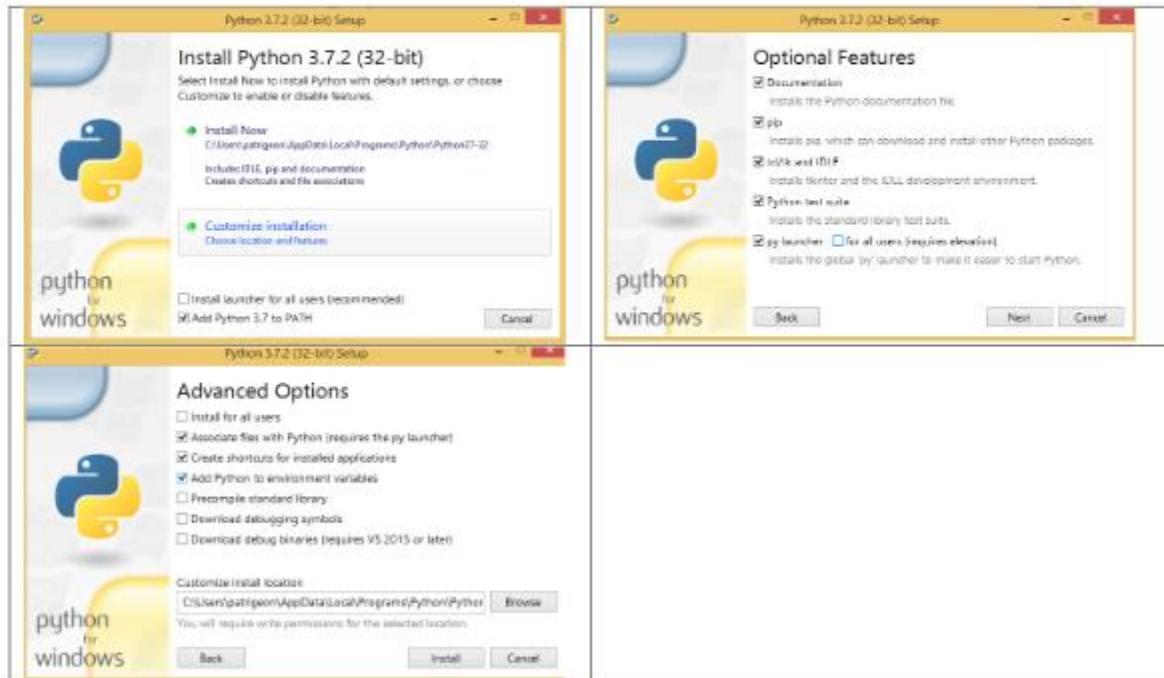
d:\Software\Tutorial_Solution\tutorial_main.c - Eclipse IDE
File Edit Search Refactor Navigate Project Run Window Help
No Local Configuration
File Edit View Insert Run Project Tools Window Help
Tutorial_Solution
  +-- Main.c
  +-- Includes
  +-- lib
  +-- startup
  +-- Tutorial_Solution.dfx
  +-- Tutorial_Solution.his
  +-- Tutorial_Solution.map
  +-- Tutorial_Solution.mk
  +-- Tutorial_Solution.elf
  +-- Tutorial_Solution.hex
  +-- Tutorial_Solution.siz
  +-- GPS.h
  +-- GPS.c
  +-- main.c
  +-- system.c
  +-- TIMER.h
  +-- TIMER.c
  +-- usrt.c
  +-- usrt.h
  +-- usrtsettings.h
  +-- startup

Main.c
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195

```

Installation du Codeloader

Lancer Python .exe



Taper cmd dans la barre de recherche windows

Taper : pip install pyserial et taper enter

```
Microsoft Windows [version 6.3.9600]
(c) 2013 Microsoft Corporation. Tous droits réservés.

P:\>pip install pyserial
Collecting pyserial
  Downloading https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c0907414e2a01a7c88bb3915cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any.whl (193kB)
    100% |████████████████████████████████| 194kB 2.2MB/s
Installing collected packages: pyserial
Successfully installed pyserial-3.4

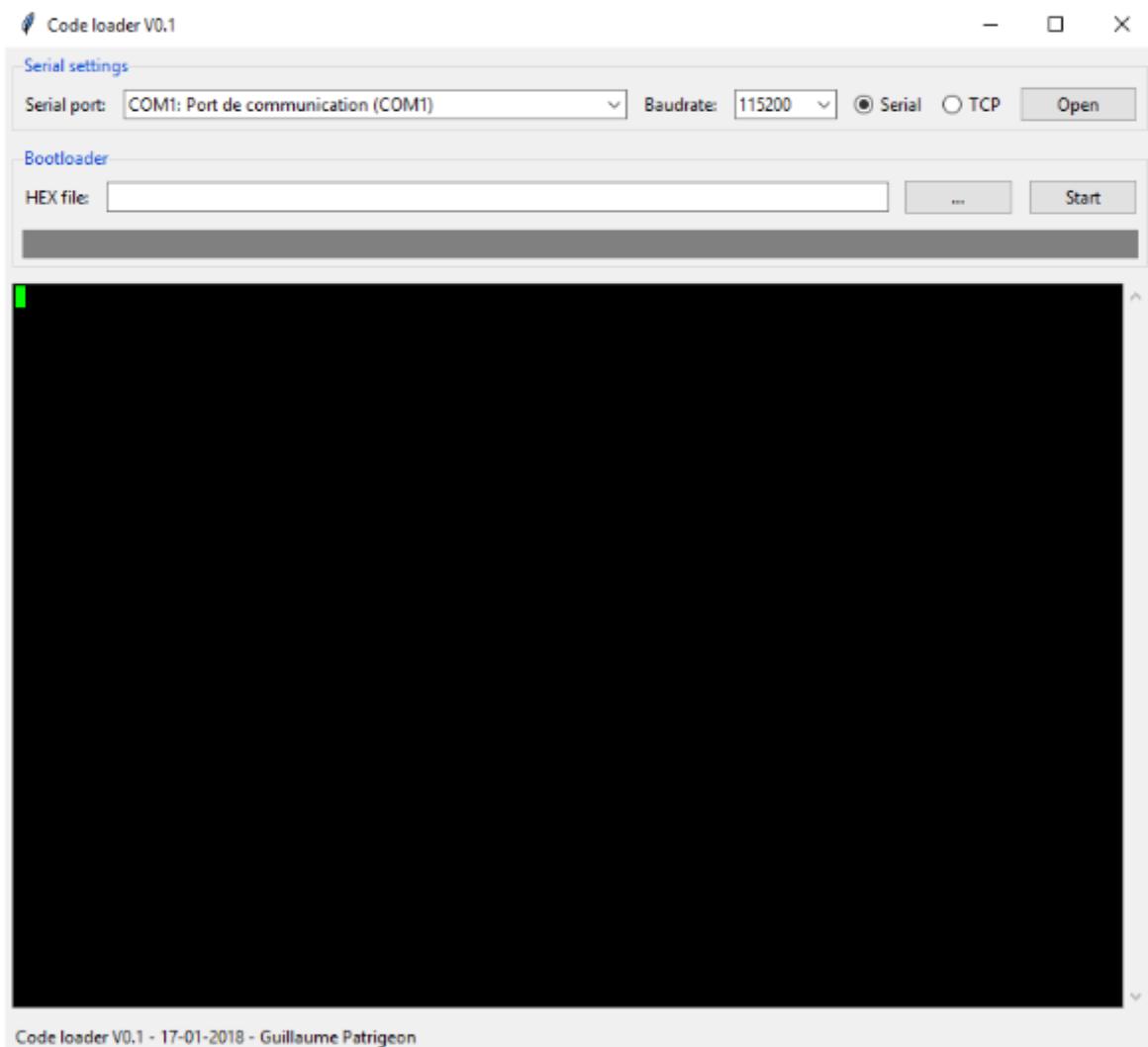
P:\>
```

Lancer main.pyw pour vérifier la bonne installation !!!!

Il se trouve dans le dossier compressé du tutoriel dans le dossier Codeloader. Décompresser le dossier préalablement.

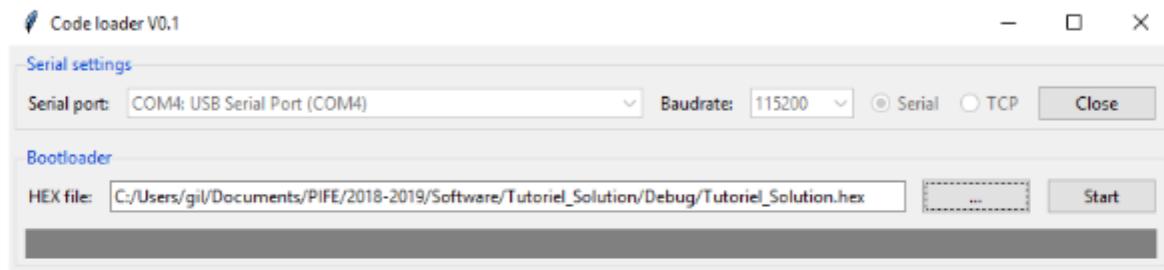
Utilisation du Codeloader

Double cliquez sur le main.pyw pour ouvrir le codeloader, cette fenêtre devra apparaître :

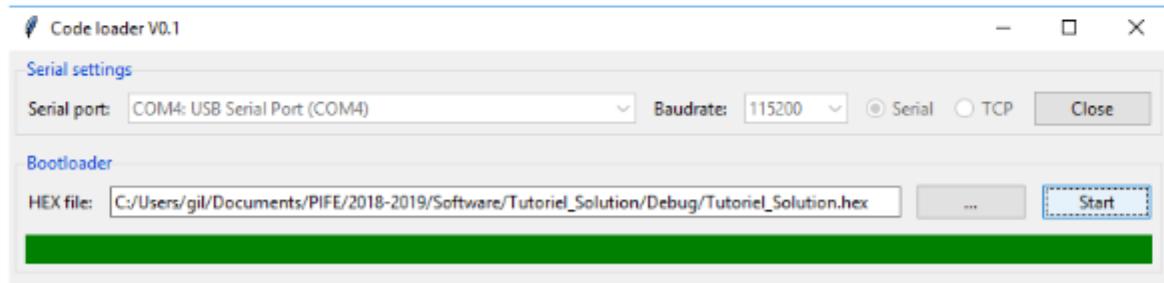


Une fois le Codeloader lancé, sélectionner le port COM où est branché le CmodA7, mettre un Baudrate de 115200 en liaison Serial puis ouvrir la liaison série.

Sélectionner ensuite le fichier hex file et télécharger le code.

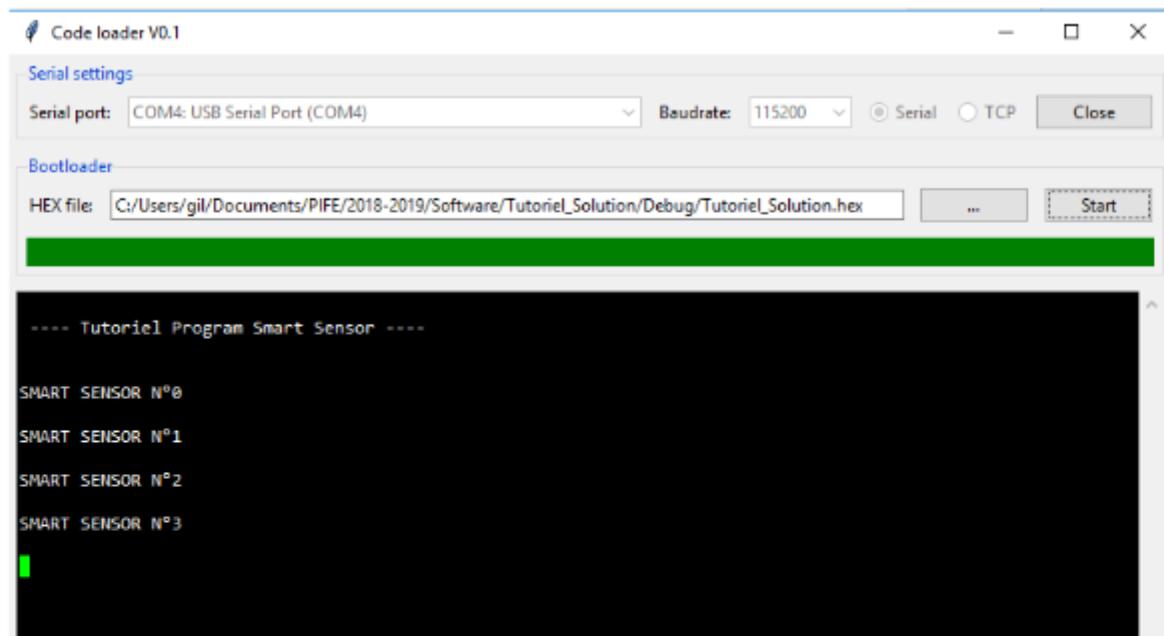


Pour télécharger le code, il faut appuyer sur start et rapidement appuyer sur le reset du CmodA7 qui est le bouton le plus proche de la connectique Pmod.



Si le téléchargement est réussi, la barre de chargement est verte, si le reset n'a pas été appuyé à temps, la barre deviendra rouge.

Si tout a été fait correctement, des informations s'affichent à l'écran et lorsque l'on appuie le bouton BTN1, la LED1 s'allume.



Faire clignoter la LED 2

Adapter le code pour faire clignoter la LED2 avec une période de 2 secondes

2.2. Utiliser l'interface UART avec le module GPS

Initialiser l'UART2

Il faut aller dans le system.c pour initialiser les PIN en entrée/sortie de l'UART2 puis définir l'utilisation de l'UART2 dans uart.setting.h

Les afficher

Renvoyer les données de l'UART2 sur l'UART1 pour l'afficher à l'écran.

Les stocker

Renvoyer les données de l'UART2 dans un tableau, puis afficher le tableau à l'écran.

2.3. Mettre sous un bon format

Machine d'état

Faire une fonction dans GPS.c pour n'afficher qu'une seule trame. Celle qui est intéressante pour le projet (date, heure, latitude, longitude). Voir la signification des trames sur le datasheet du module.

Mettre cette trame dans un tableau et afficher ce tableau.

Ensuite afficher l'heure, la date de manière lisible pour quelqu'un qui ne connaît pas la structure de la trame.

Créer une trame

Créer une trame avec les informations intéressantes dans le but de l'envoyer plus tard par un module LoRa.

3. Pour approfondir

3.1. Utiliser les autres interfaces du CmodA7, SPI et I2C

3.2. Envoyer la trame par LoRa