

Time Synchronization in network

2ndYear Internship at Northumbria University

* + - * + **MORAND Baptiste**
        + **2A – ELEC / SATE**
        + **Academic year 2015/2016**
        + **Superviseur : Xuewu Dai**
        + **Teacher : Miloud FRIKEL**

ACKNOWLEDGEMENT

# Contents

[ACKNOWLEDGEMENT 1](#_Toc457045936)

[I. Contents 2](#_Toc457045937)

[II. Introduction 3](#_Toc457045938)

[III. University and lab. presentation 4](#_Toc457045939)

[IV. Requested work and required skills 5](#_Toc457045940)

[A. State of the art of time synchronization 5](#_Toc457045941)

[B. Project Presentation 9](#_Toc457045942)

[1. AIM 9](#_Toc457045943)

[2. Objectives 9](#_Toc457045944)

[3. Hardware selection 9](#_Toc457045945)

[C. My work 11](#_Toc457045946)

[1. Software selection 11](#_Toc457045947)

[2. Application construction 11](#_Toc457045948)

[3. Correction 13](#_Toc457045949)

[4. Bench Test 14](#_Toc457045950)

[V. Conclusion 17](#_Toc457045951)

[VI. References 18](#_Toc457045952)

[VII. Annexes 19](#_Toc457045953)

[VIII. French resume 1](#_Toc457045954)

# Introduction

Actually more and more device are connected together with different network, this is what we call “internet of thing” (IOT), with more than 2 milliard of embedded system sell in the word. This is why there is more and more sensor networks used to communicate with the real word. A wireless sensor network (WSN) is a group of sensor nodes that are connected through wireless communication links and each node is equipped with some kind of capabilities of sensing, data storage, data processing and communication. WSN have been widely used in commercial and home applications, such as environment monitoring, building/home automation. The trend in WSN is the time-sensitive industrial applications that require precise timing among WSN nodes. For such applications physical time play a crucial role. There are three principal applications:

* Measurement Observer: In automation the thing the most important is the sample time, but to have a good accuracy of a sample time we need time synchronization.
* Real time: One other thing important is to link the data from sensor to the real time.
* Synchronization: Time is also a valuable tool for intra-network coordination among different sensor nodes. Many applications of time known from traditional distributed systems also apply to wireless sensor networks. There is too the synchronization of sensor, how to acquire some data at the same time.

The actual problem is that time synchronization have a cost. Time synchronization force to have exchange of packet through a network, this exchange of packet will create a decreasing of the bandwidth for the real data sending. It will have the cost of consumption cost by the communication, and calculation of time and correction. If really you need a high accuracy you will need have a extra cost due to some change of the hardware.

My project consists in making a time protocol in a SAMR21 Atmel board, through a wireless IEEE 802.15.4 protocol and to correct my clock. For this I will read some paper to understand how time synchronization work and how to configure the board. And I will think about a protocol to make the synchronization and an algorithm to make the correction. I will finish by testing my system in real condition in an industry. And finally I will make some test for having an idea of the accuracy of my project.

# University and lab. presentation

I have work in the University of Northumbria in Newcastle Upon Tyne (United Kingdom). This university has 3 campuses, one at Newcastle, another at Coach Lane and finally one at London.

Northumbria has a student population of around 32,000. Teaching and research activities are managed by four faculties: Arts, Design and Social Sciences; Engineering and Environment; Business and Law; and Health and Life Sciences.



I have work in the department of Engineering and Environment, this department is a building of five flours with a lot of laboratory, there is some research in optical, microelectronic, microwave, automation, and embedded system.

# Requested work and required skills

## State of the art of time synchronization

Time synchronization is to synchronize the drifting clock of a slave node to the reference clock at the master node. The master is chosen because it has the best clock. For example, a GPS clock can work as the master node. In general the master node is synchronize with a clock that is given by a GPS. The GPS generate a pulse per second (PPS).

GPS

Master node

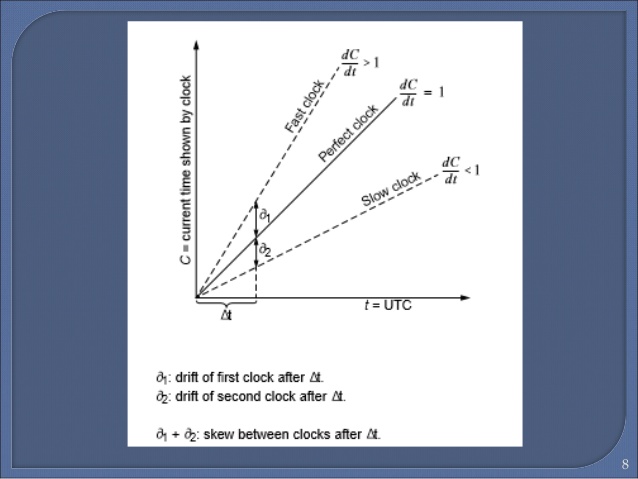
Slave node

Slave node

PPS

Network

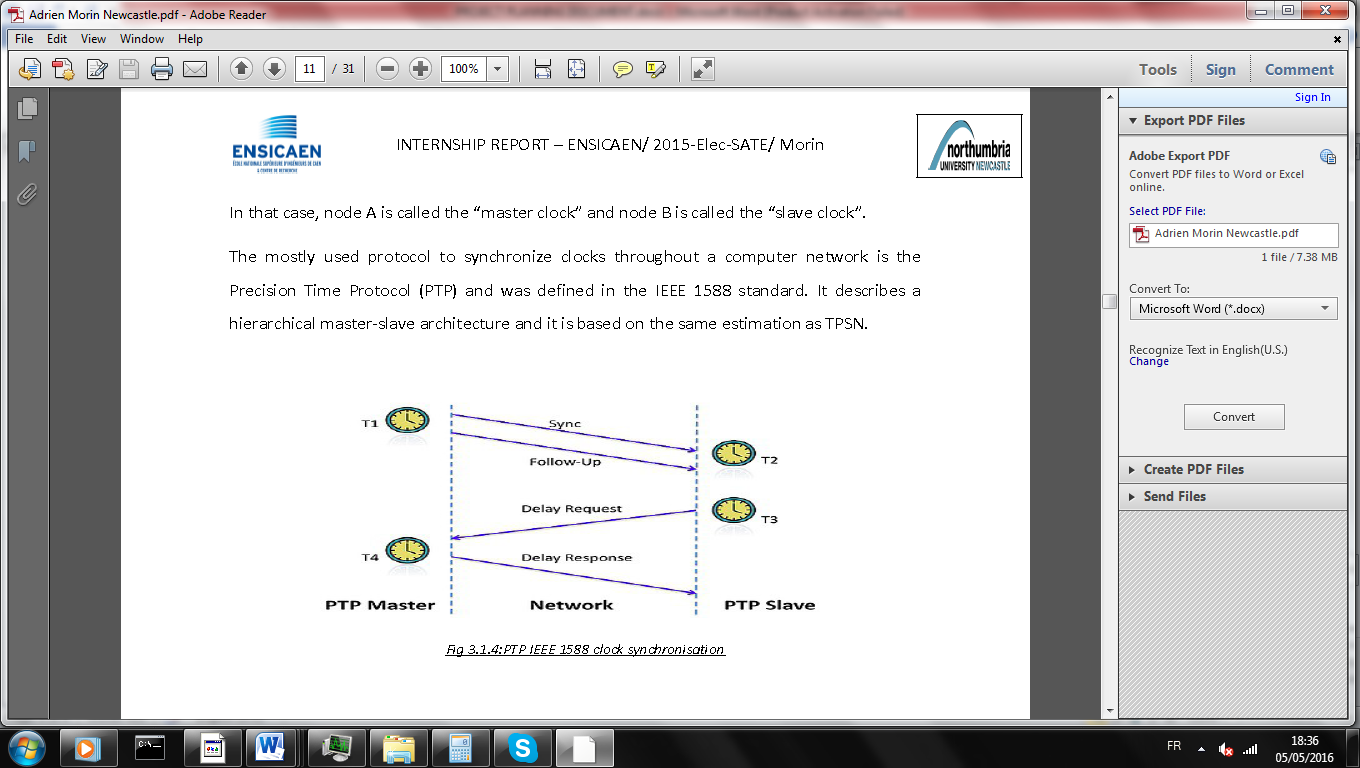
As I say tine synchronization is very important, and the main problem is the oscillator of the embedded system. Because we need to reduce the cost of our system and we cannot put a GPS on all your devices. The oscillator of our system depends of the constructor, the temperature, and the alimentation (see anexe number 1 p ??). All this things will create a disturbance. To measure the differences of frequency we use the skew. The skew is the differences of frequency between the master node and the slave node. This difference of frequency will create a temporal shift.



Before correcting the skew we need to measure it this is why we use a precision time protocol (PTP) who is specified in the standard IEE1588. This algorithm is use to know this 2 unknow time :

* the offset : who is the differences of time between the master and the slave.
* The delay: the delay is time between sending the information and the time to receiving the information he is the sum of 3 times: the time to code the information, the propagation time and the time to decode the information.

The algorithm work like this:



Four types of packet have been defined in the PTP:

1. **Sync:** The process is started when the master send Sync packet with the timestamp who correspond to the time where master send. Sometimes generic hardware and software stacks that implement PTP don't have the ability to add a precise timestamp to a message they are in the process of generating this is why the timestamp T1 is send in a **follow up** command.
2. The Slave save the time when the packet is received (T2).
3. **Delay request:** The slave send a delay request to the master with no data. And he save the time of sending (T3)
4. **Delay Response:** The master request with the time when he have receive the request (T4).

Sync, Delay Request (DReq) and Delay Response (DRes) are three compulsory packets and the Follow Up is a optional one. When the time stamp T1 cannot be sent in packet Sync, a Follow Up packet will be used to send T1 to the slave

In practice, a slave node cannot accurately estimate the local time on the target node due to varying message or network delays between the nodes. But we can make an approximation with a little error this is why we consider that the delay is constant in the time, and the frequency of the slave and the master will not change during the period of synchronization. With the time given by the PTP we can calculate the delay and the offset.

Master

Slave

d + Xi +

d + Xi -

T4

T3

T2

T1

d is the delay, Xi a random delay, is the offset.

T2-T1=d+Xi + T4-T3= d+Xi-

With the information we can show that:

When can conclude that time:

Tslave=Tslave+.

## Project Presentation

### AIM

The aim of this project is to develop and implement a precision time synchronization protocol for industrial sensor networks. Then an IEEE 1588-like time synchronization algorithm will be developed and validated to have precision time synchronization among at least two wireless sensor nodes.

The C/C++ language will be used for an ATSAMR21G18A microcontroller.

### Objectives

For arrive to the aim there is some objective:

* Understanding IEEE 1588 Precise time protocol.
* Understanding wireless IEEE 802.15.4 protocol and Universal Asynchronous Receiver Transmitter (UART) and use it with the microcontroller.
* Write code in C/C++ language to realize the algorithm.
* Test the algorithm and test his accuracy.
* Use the GPS to be at the time of Coordinated Universal Time (UTC).
* Synchronize the master node with the PPS of the GPS.

### Hardware selection

**Microcontroller:**

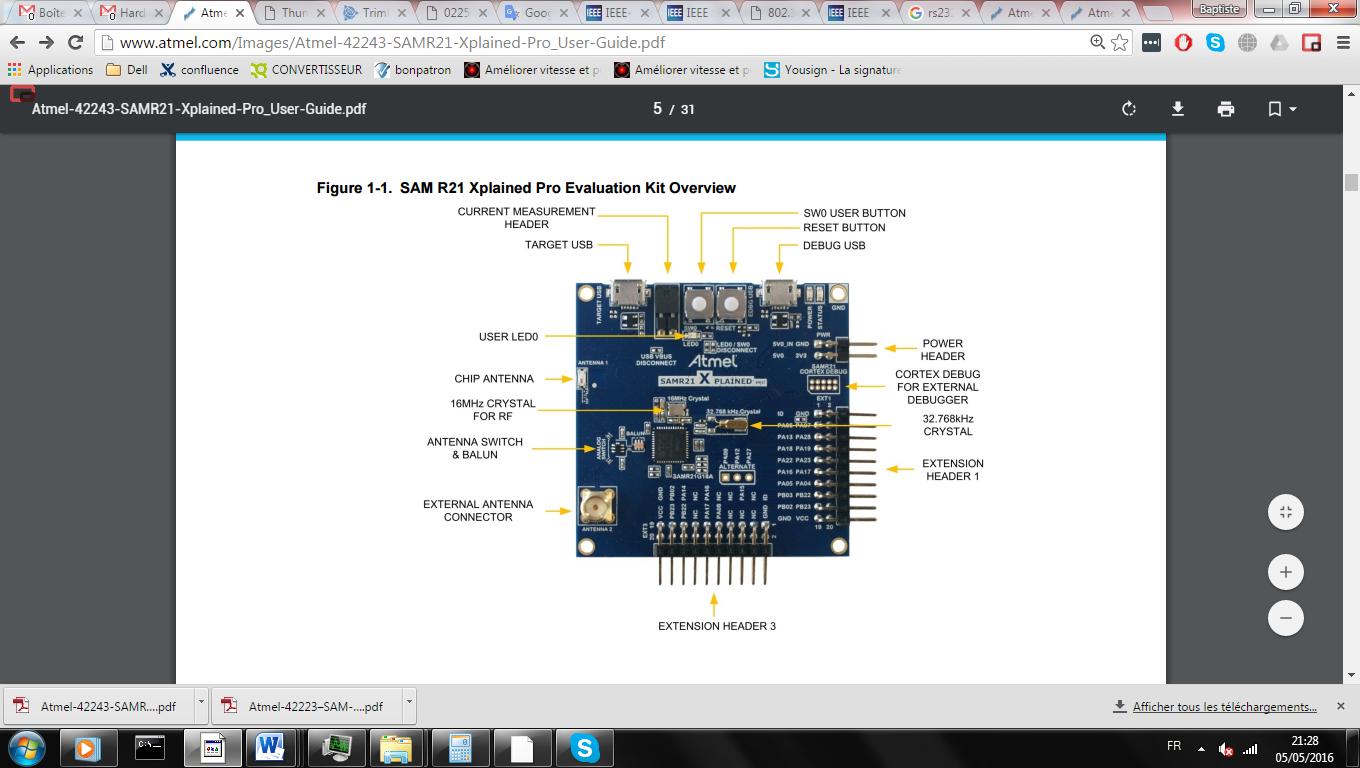
As I said before my microcontroller will be the ATSAMR21G18A this microcontroller have the particularity to have IEEE 802.15.4 controller integrated inside the chip.

**ATSAMR21G18A main characteristic:**

* Supply Voltage : 1.8V to 3.6V
* Memories :
  + 7256KB in-system self-programmable Flash
  + 32KB SRAM
* Peripherals
  + 28 GPIO
  + 3 timer counter
  + 1 usb interface
  + 5 communication interface (UART,I2C,…)
  + 8 ADC channel
  + Watchdog timer
  + Integrated Ultra Low Power Transceiver for 2.4GHz ISM Band

The IEEE 802.15.4 is a communication protocol defined by IEEE. It is designed for wireless networks of the family WPAN LR (Low Rate Wireless Personal Area Network) due to their low consumption, low-range and low-flow devices using this protocol.802.15.4 is used by many implementations based on proprietary or IP (Internet Protocol), such as ZigBee and 6LoWPAN.

This microcontroller will be use with industrial board SAMR21 who have a clock at 32KHz.



**Global Positioning System (GPS):**

For generate a PPS we will use a ThunderBolt® E GPS Disciplined Clock this particularity is the PPS that is generate with a high accuracy.

**ThunderBolt® E GPS Disciplined main characteristic:**

* +24 VDC power supply.
* BNC connector :
  + Equipment to analyze the 10 MHz output frequency.
  + 1 PPS accuracy (0 to 2,4V).
* RS-232 through a DB-9/M connector.
* Antenna interface.

## My work

### Software selection

My first things to do is to select my integrated development environment (IDE) I will use the compiler and the software Atmel Studio 7 who is the software given by the constructor. I have choose it because it is free of charge and is integrated with the Atmel Software Framework (ASF) a large library of free source code with 1,600 project examples.

After, I have to select the software library and the different use in the project. The big issue when you make time synchronization is knows well how work your peripherals in particular:

* **Timer:** The timer is very important because it’s him that will keep the time and all the application will depend of his accuracy. This is why I choose to configure him directly with my own library.
* **Network:** The network is very important because it is him that will manage the time Protocol. The library will depend of the network choose. I have choose to use (Atmel Studio Framework) ASF who is a library given by the constructor that is specific to our microcontroller, it is just a driver to configure easily the peripherals.

I have chosen to add a Real Time Operating System of my project (RTOS) this Operating System (OS) because it’s easy to modulate the application, for example if you want to add a functionality you just have to add a task, you can modulate your application as you want, and because it’s offer a lot of services (Queue, task,…). It is very easy to use because you just have to create some task (it is like a function) and to define priority on his task. And this is the OS that will manage your application. I have choose FREERTOS as RTOS because it is free, the most use RTOS for embedded system and because I already know it because I have study in course.

### Application construction

**Software:**

My project has 2 tasks:

* **Time protocol task:**

This task will be use to communicate with the network to calculate the offset and the delay. He will be use too for correcting the frequency. This task will have the maximum of priority due to its sensitivity to delay. This task is the task the most important in my project it will consist in 3 main subtasks: Sending receiving and correction. The Sending and the receiving subtask will manage the time protocol through the network like this we can know the offset and the delay. But the main objective is to correct the frequency this is why we will use a correction subtask, the correction will use the offset to calculate the average skew, and to correct the frequency.

I will change a little the PTP protocol to use a protocol with less of packet exchange and in keeping the same accuracy. Like this we can synchronize lot of slave node without improvement of the network. And due to this it will decrease the power consumption of the device.

My protocol is constituted with 2 different processes:

* + Delay : This will be 2 packet one delay request, and one delay response
  + Sync: this command is use to correct all the device at the same time with one packet come from the master.

After define this protocol I have make some graphical of my application. (see annexes 2)

* **HMI task :** This task is just to check that the application turn well and just to inform about the accuracy with the human machine interface (HMI)

I will have 3 interrupts:

* **Timer interrupt:** This interrupt will provide the basic clock to the WSN node. By counting how many interrupts have been generated and reading the reminder in the Timer reminder, the node is aware the time. The interval between two interrupts can be configured by the timer’s threshold. For performance of comparison, this timer-interrupt clock will also be used to generate a PPS to make a comparator with the PPS that coming from the GPS clock. Like this we can measure the accuracy of the application. The period of this timer’s is the Tthreshold
* **Receive network interrupt:** This interrupt will be call to save the time when a data is received.
* **PPS interrupt:**  This interrupt will correct the master it is needed to synchronize the master with the GPS.

**Peripherals management:**

I have different peripheral to configure and to manage and I have different tools to make this:

* **Mbed :** This application given by Mbed is a very general layer to use peripheral without consideration of hardware this is why it not offer lot of option to control the peripheral, but it is the best for a faster development.
* **Atmel Studio Framework (ASF) :** This application given by Atmel can be used with the entire Atmel microcontroller. It offers a lot of option just in passing argument in some function but the management is not made by you. The difficulty of this layer is average but you need to understand partially how your peripherals work.
* **Register configuration:** This layer is the lowest value just before assembler it offers the full option of the peripheral but it is most difficult to use because you have to understand very well how the peripheral use. But you have a totally control of the execution and the configuration. This layer is very important to control well some peripheral critical of your application.

For all the peripheral it is important to choose the best solution of management, because I have not a lot of time to do my project and I need an accuracy of the time this mean reduce at the maximum the delay due to some library.

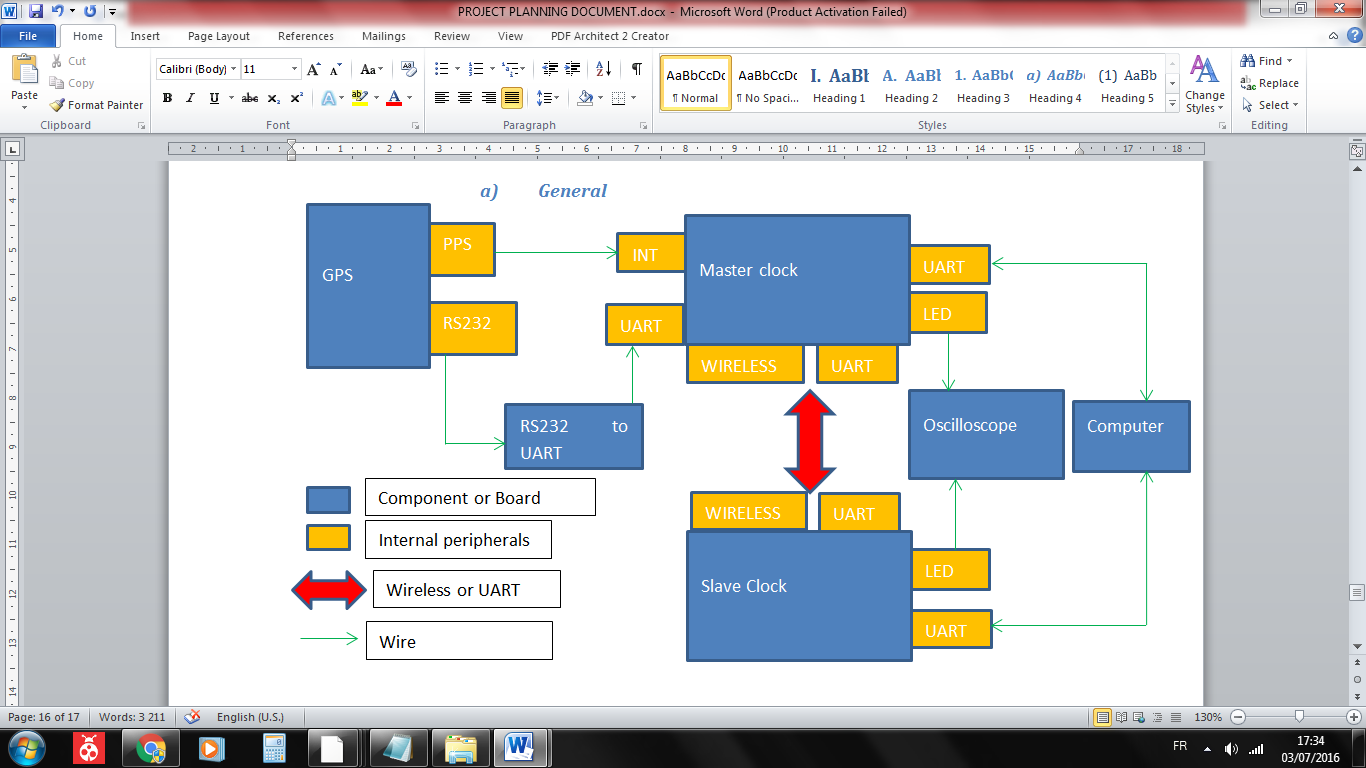
This is the list of the peripheral and the management of this :

* **Internal timer:** This peripheral is very important because the accuracy of my application will depend of him. This is why I need to use the register for the configuration.
* **External interrupt:** This is very important to control well this peripherals because it is use to synchronize the master with the PPS. I have use ASF for the configuration because ASF provide enough option to configure easily this peripheral
* **Serial communication:** This is not important because it is use just for the HMI so I can use mbed to configure the peripheral.
* **Wireless communication:** This controller is very difficult to configure an make about one month to make the configuration and the management this is why I use a mbed library.
* **GPIO:** This is not very important because it is need to generate a signal for a LED so I have use mbed to configure it.

**Hardware:**

I have chosen how my hardware will be built. And what peripheral I use to make the communication. I need a RS232 to UART converter to convert the signal that coming from the GPS to the Master clock to get the UTC time.

I will use a computer for the HMI and oscilloscope to validate the accuracy of my application.



### Correction

As I said before correction is important because we need to correct the frequency to keep accurate without depending of the time. The correction is the same principle for the slave and the master node, the only differences is how to calculate the offset, for the master node it is with the PPS and for the slave node it is with the network protocol.

I have defined 2 ways to correct the clock:

* **Hardware correction :**

We have a threshold period that will actualize the time of a constant time, the principle is too not change the theoretical value add to the time each threshold but to change the real time for corresponding to the time given by the master. Like this we can change the frequency of the local time to correspond to the master time.

We find that **Tthreshold(i+1)= Tthreshold(i)** avec the offset per Tthreshold. Like this we can calculate the new parameter for the timer’s threshold.

* **Software correction :**

Here, the principle is to adjust the clock at a regular interval, for example we can add 1ms every threshold period. Like this we will have a better accuracy. We have 2 parameters to find:

* the value to correct Ccorr
* the number of Tthreshold to make this correction Ncorr

We know the Offset per threshold.

We have to solve the equation: **Ccorr=Ncorr x** with **Ccorr and Ncorrℕ**

### Bench Test

For all the tests the hardware will be the same, the GPS will send a PPS to a slave and the master node. Each pulse the master and the slave will report the offset to the HMI. The network use is a wireless communication with IEEE 802.15.4.

Slave Node

Master Node

GPS



HMI

Network

IEEE 802.15.4

PPS

Serial COM

All the measure of offset will be done with a sample time of 1s command by the PPS.

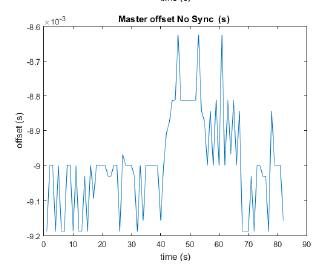
The threshold period is of 0,3125s for all the test.

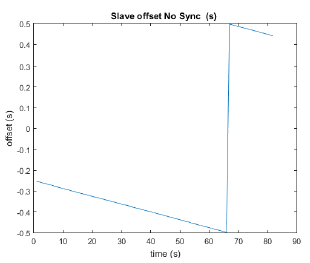
* **Accuracy**

This test is a final result of my project it will measure the accuracy of my project.

For the entire slave test the master will be with synchronization and with all the correction. The slave offset is the difference of time between the slave clock and the GPS clock.

**Basic system:**

In this test the network is not use just the PPS will correct the offset of the master but without correction.

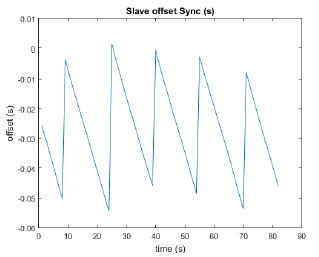


For the slave we can see that the offset is not change and that the difference of frequency is constant.

For the master we have accuracy quite constant.

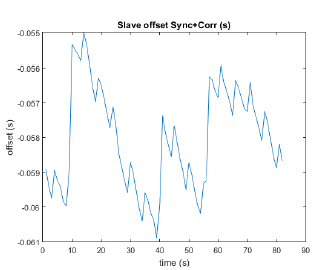
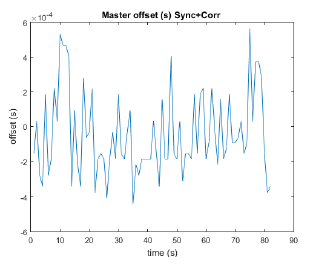
**With time synchronization:**

Now we don’t change the master and the slave will use the network to be synchronized. The synchronization will be done every 15s and we measure every second the offset.

We can see that every 15s the offset will be adjust and between 2 synchronization we have the difference of frequency that is constant.

**Correction:**

Now we can correct the threshold frequency using the time synchronization and the corrections.

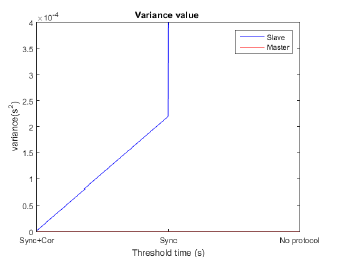
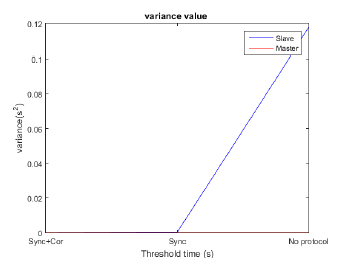


For the master the accuracy is a few hundred of us.

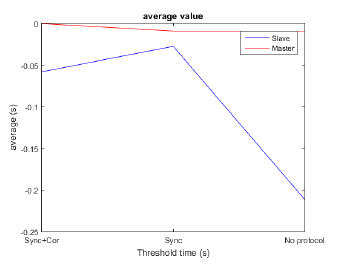
For the slave the accuracy is a few ms.

**Comparison:**

Now all done we can make a comparison between all the update of my time protocol. What is very important to look is the variance because it is him that will influence a lot the accuracy because the average can be compensate if we know it.



We can see that the variance of the master is better than the variance of the slave it is good because it is the master need that to synchronize the slave clock.



# Conclusion

To conclude, my project is to make a time Synchronization for industrial sensor network, I need to make synchronization with the lower cost possible. The system is composed of one master and multiple slave. In the final of my application I arrived to make a synchronization, the master have an accuracy of a few hundred us and the slave have an accuracy of a few millisecond. Of course we can improve the system by having a best control of the peripherals and in adding a Kalman filter to predict the offset. Like this we can know the offset in advance and compensate directly before having an error.

This project will be very interesting for me because I have discovered the board and the wireless peripherals that use IEEE 802.15.4; I have understood how to use it. I have understand too what is time synchronization, how it work and why it is use for. I have discovered how to configure some peripheral from different way.

I have the occasion of practicing English language for general speaking in Newcastle Upon Tyne (UK) but also for technical discussion with different people. Like this I have taken some advice about my project, and I have confirmed that discussions in team are very important to improve the project and to exchange some idea.

I expect that my project will be taken again to improve the algorithm and the microcontroller management, this is why I have written a lot of document to explain in detail how my project and some idea of improvement.

# **References**

**Clock Synchronization for Wireless Sensor Networks: A Survey** Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani Department of Computer Science

**Time Synchronization for Wireless Sensor Networks􀀀** Jeremy Elson and Deborah Estrin

Department of Computer Science, University of California, Los Angeles; and USC/Information Sciences Institute

**Time Synchronization and Calibration in Wireless Sensor Networks** Kay R¨omer, Philipp Blum, Lennart Meier ETH Zurich, Switzerland

**IEEE Standard for Local and metropolitan area networks Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) :**IEEE Computer Society sponsored by the AN/MAN Standards Committee

# **Annexes**

# **French resume**