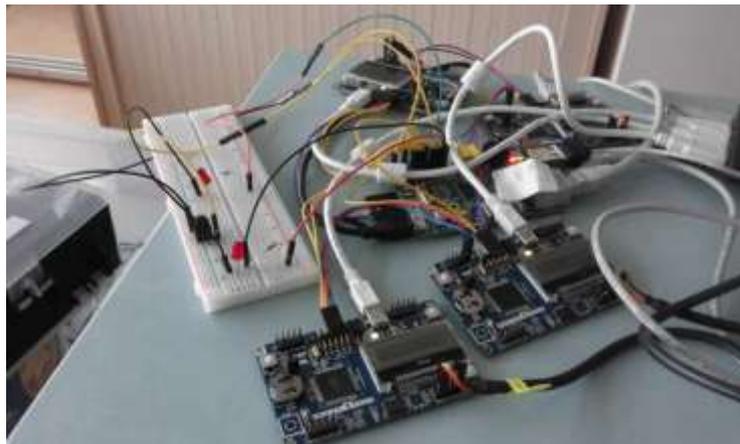


# Plateforme expérimentation IOT

---



ROCHE François

23/02/2016

## Table des matières

I-	Présentation des IOT .....	2
II-	Le cahier des charges .....	3
III-	Présentation de la plateforme .....	5
a)	Sa composition .....	5
b)	Sa structure .....	6
IV-	Sa programmation.....	7
a)	PDI .....	7
b)	Procédure de programmation.....	9
1)	Configuration sur une Raspberry PI .....	9
2)	Configuration sur une Raspberry PI 2.....	10
3)	(Re)Programmation du nœud .....	11
c)	Procédure de remise à zéro .....	14
V-	Les données.....	15
a)	Capteur.....	15
b)	Format de la trame des données .....	16
c)	Remonter des données en mode normal.....	17
d)	Remonter des données en mode dégradé.....	20
VI-	Interface web .....	21
VII-	Pour aller plus loin.....	22
VIII-	Bilan .....	23

## I- Présentation des IOT

Dans un monde qui va devenir connecté, il a été décidé de réaliser un projet sur ces nouvelles technologies. La plateforme pourra être implantée dans la nouvelle bibliothèque universitaire du campus de Lille. Les utilisateurs pourront bénéficier d'informations disponibles tel que les places disponibles, la température, la luminosité ....

Ces informations dépendront des capteurs mis sur le nœud.

Le deuxième intérêt que l'internet des objets nous procure, c'est l'historique et la prévision. En effet on aura une possibilité de stocker les informations :

Exemple :

- Commander le chauffage en fonction des températures extérieures ou d'estimer la consommation par rapport aux années précédentes.
- Effectuer un suivi de la fréquentation de la bibliothèque. Cette fréquentation pourra être analysée afin de mettre plus de personnel lors des périodes de forte affluence comme des jours étudiants.

Le but de ce projet est de développer une plateforme d'expérimentation.

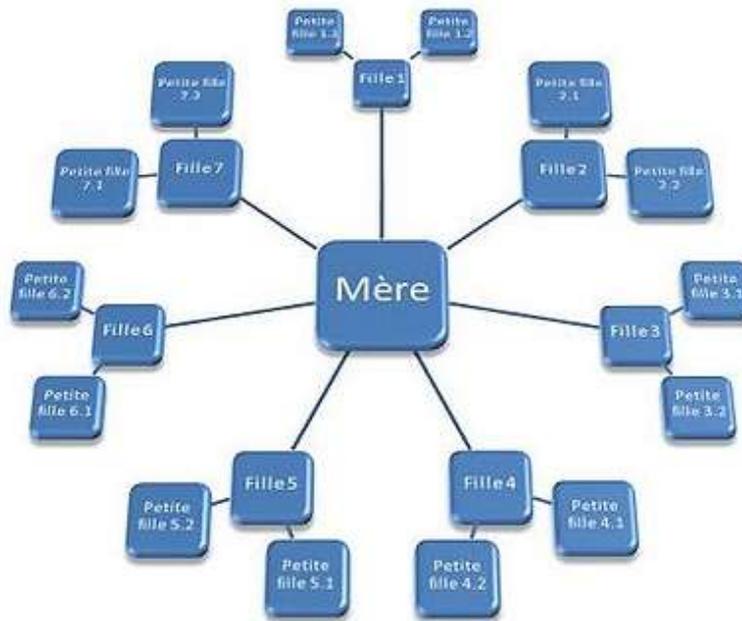


## II- Le cahier des charges

Le cahier des charges qui a été établi est le suivant :

- La structure d'un nœud

Le nœud sera composé d'une carte mère, de 8 cartes filles et 16 capteurs.

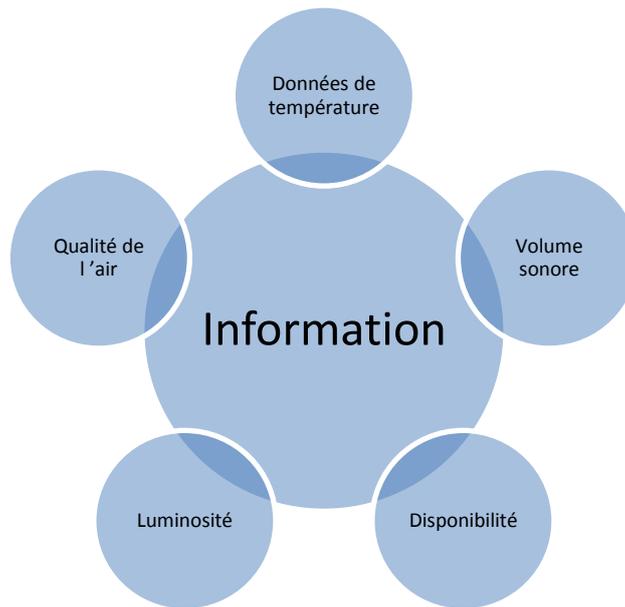


La carte mère comportera un ordinateur embarqué de type Raspberry, d'une unité USB ainsi que de liaison série. Cette carte aura les fonctions suivantes :

- récupération les fichiers de configuration.
- Reprogrammation des cartes filles et petites-filles.
- Mesure de la consommation du nœud.
- Communiquer sur le réseau les informations des capteurs.

La carte fille embarquera deux cartes petites filles. Ces dernières communiqueront à l'aide de la liaison radio.

Les informations que la plateforme d'expérimentation pourra remonter :



Ces informations sont importantes dans un bâtiment. Le nœud devra être reconfigurable par tout le monde (chercheur, étudiant ...). Les personnes qui utiliseront ce nœud pourront tester également des capteurs.

- Mode de communication en fonction du mode de fonctionnement

Le tableau ci-dessous vous indiquera les modes de communications du nœud

	<b>Fonctionnement normal</b>	<b>Fonctionnement dégradé</b>
Transmission des données des capteurs	Radio et USB	USART
Reconfiguration du nœud	PDI	PDI
Transmission des données énergétiques	USART	USART
Communication avec le réseau	Ethernet	Ethernet

### III- Présentation de la plateforme

#### a) Sa composition

La plateforme à la fin du projet est constituée de deux nœuds :

Nœud 1 :

Carte mère

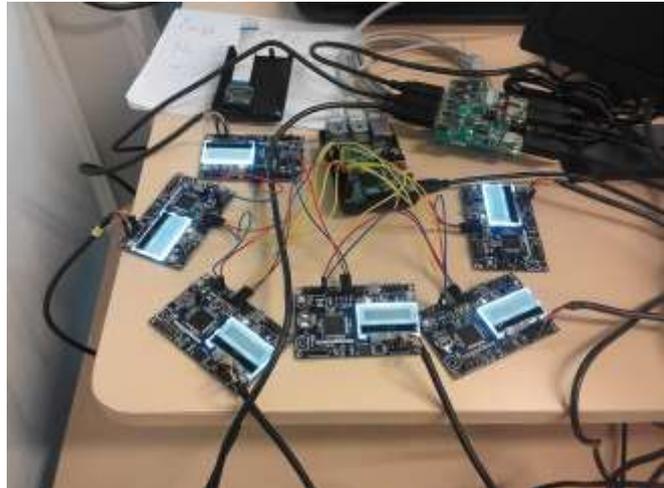
- 1 Raspberry Pi2
- 2 cartes EVB-USB2517

Carte fille

- 6 Cartes d'essais Xplained PRO

Carte petite fille

Ce nœud n'est pas équipé de carte petite fille. Les capteurs sont simulés.



Nœud 2 :

Carte mère

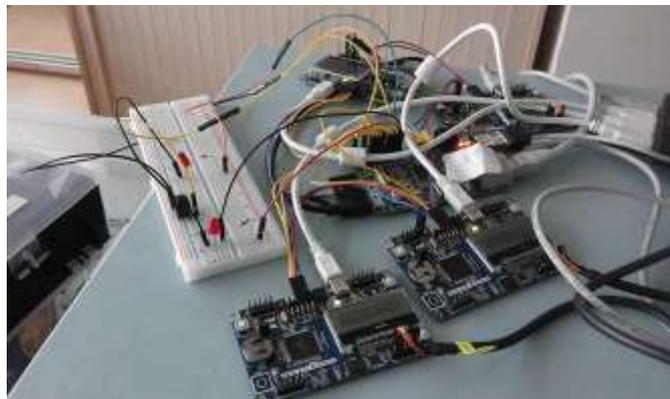
- 1 Raspberry Pi
- 1 carte EVB-USB2517
- 1 hub USB
- 1 bouton de remise à zéro
- 1 voyant d'état

Carte fille

- 4 Cartes d'essais Xplained PRO

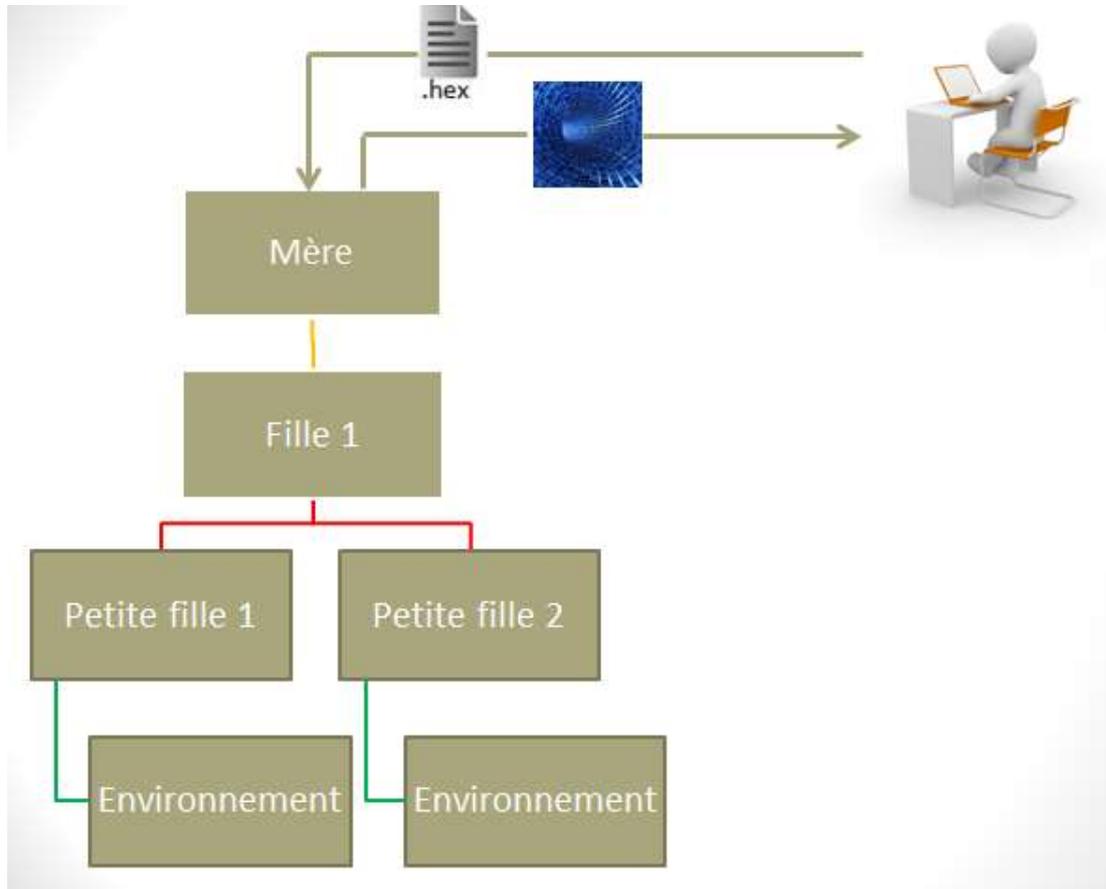
Carte petite fille

Ce nœud n'est pas équipé de carte petite fille. Les capteurs sont simulés.



## b) Sa structure

Vue d'une branche d'un nœud :



Les connexions disponibles sur les nœuds sont :

Programmation	
Carte fille	PDI (jaune)
Carte petite fille	SPI (rouge)
Remontée des données	
Carte petite fille -> Environnement	Radio (Vert)
Carte petite fille -> Carte fille	UART /I2C (rouge)
Carte fille -> Carte mère	UART / USB (jaune)
Carte mère -> Réseaux	Ethernet

L'utilisateur doit déposer un fichier .hex sur la Raspberry pour pouvoir reconfigurer le nœud

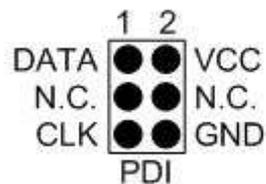
## IV- Sa programmation

### a) PDI

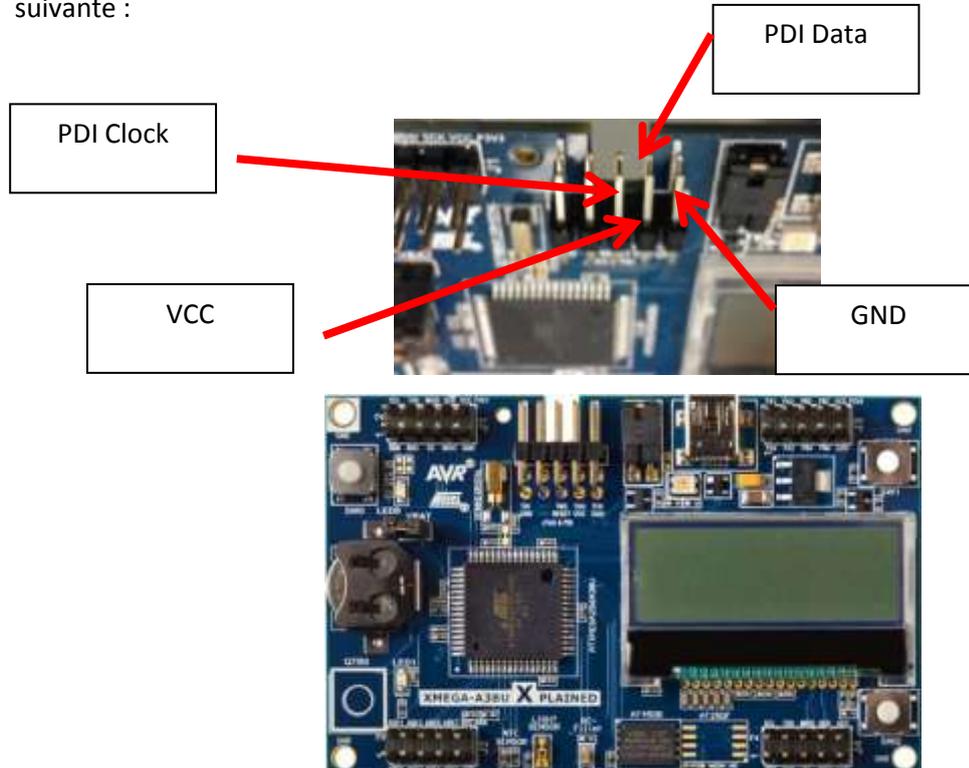
Le PDI est une technologie propriétaire de chez Atmel qui nous permet de reprogrammer les nœuds de capteurs. Le PDI est composé de 4 éléments qui sont :

- PDI Data
- -PDI Clak
- GND
- VCC

Ces quatre éléments peuvent se rencontrer sous forme d'un connecteur comme nous l'indique l'image ci-dessous :

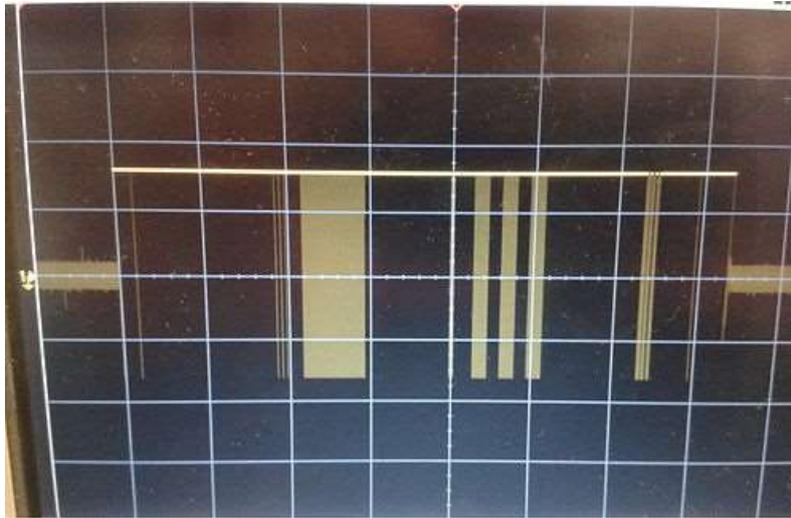


Le PDI sur les cartes d'essais Xplained se présentait différemment comme nous le montre l'image suivante :



A l'aide du programmeur d'ATMEL nous avons observé les signaux du PDI :

Signal issue du PDI Data :



La fréquence de la trame est de 2 KHZ. On remarque également que la première partie est à un niveau logique haut, c'est normal on initialise la broche pour envoyer le programme un fichier hexadécimal (HEX).

Signal issue du PDI clock :



On remarque que l'horloge est parasitée. Ceci est dû à l'alimentation qui n'est pas la même. La fréquence minimale pour pouvoir programmer est de 10Khz.

## b) Procédure de programmation

Afin de pouvoir programmer facilement le nœud nous avons établi des configurations sur les Raspberry. Durant ce chapitre vous allez voir les différentes configurations ainsi que la programmation.

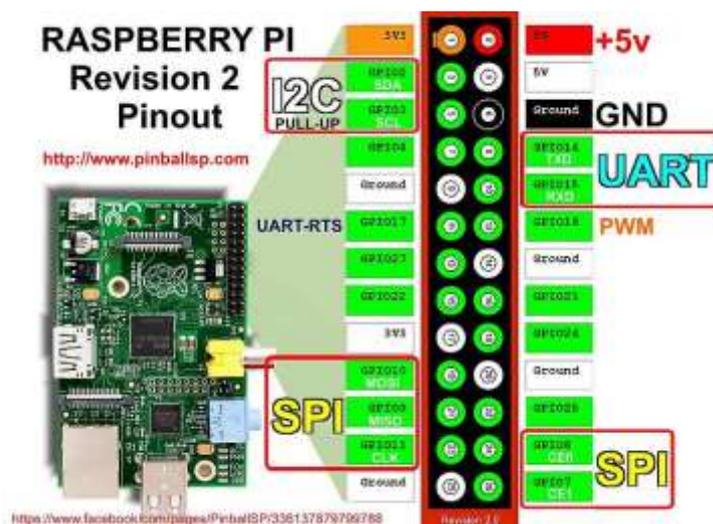
### 1) Configuration sur une Raspberry PI

La configuration pour reprogrammer le nœud avec une Raspberry Pi est la suivante :

	PDI DATA (GPIO)	PDI Clock (GPIO)
Carte A (reprogrammation de la carte UART)	23	24
Carte B	22	27
Carte C	10	9
Carte D	17	18
Carte E	4	25

	GPIO	Etats
Led	8	Eteinte Pas de programmation
		Clignotant Programmation en cours
		Allumé : Erreur de programmation

Schéma des GPIO :



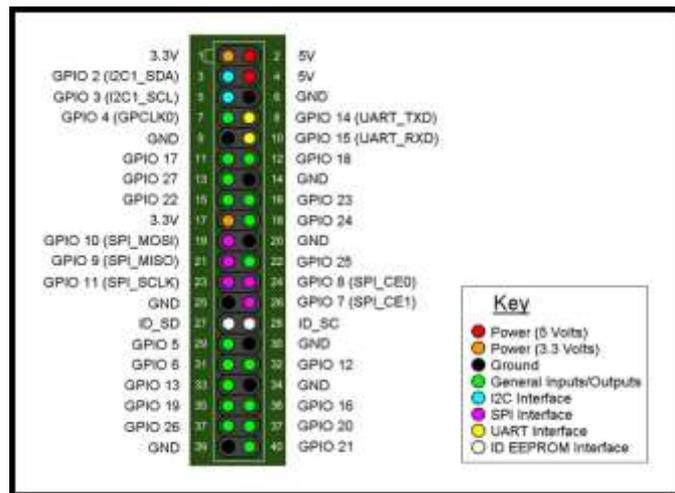
Dans le programme de configuration du nœud il faudra choisir la configuration : 1

## 2) Configuration sur une Raspberry PI 2

La configuration pour reprogrammer le nœud avec une Raspberry Pi2 est la suivante :

	PDI DATA (GPIO )	PDI Clock (GPIO )
Carte A (reprogrammation de la carte UART)	17	27
Carte B	18	25
Carte C	13	19
Carte D	23	24
Carte E	16	20
Carte F	22	26
Carte G	12	21
Carte H	10	9
Carte I	5	6

Schéma des GPIO :



Dans le programme de configuration du nœud il faudra choisir la configuration : 2

### 3) (Re)Programmation du nœud

La programmation des cartes filles s'effectue avec le PDI. La commande pour pouvoir programmer est la suivante : programmer.

Cette commande a besoin d'argument pour pouvoir reprogrammer le nœud. Un argument est un fichier .hex Le tableau suivant vous indique le nombre d'argument que vous avez besoin en fonction de l'action voulue :

Nombre d'argument	Actions
1	<ul style="list-style-type: none"><li>- Reprogrammation de l'ensemble du nœud</li></ul>
2	<ul style="list-style-type: none"><li>- Reprogrammation d'une carte</li></ul>
3	<ul style="list-style-type: none"><li>- Reprogrammation de la carte A</li><li>- Reprogrammation de la carte B</li></ul>
4	<ul style="list-style-type: none"><li>- Reprogrammation de la carte A</li><li>- Reprogrammation de la carte B</li><li>- Reprogrammation de la carte C</li></ul>
5	<ul style="list-style-type: none"><li>- Reprogrammation de la carte A</li><li>- Reprogrammation de la carte B</li><li>- Reprogrammation de la carte C</li><li>- Reprogrammation de la carte D</li></ul>
6	<ul style="list-style-type: none"><li>- Reprogrammation de la carte A</li><li>- Reprogrammation de la carte B</li><li>- Reprogrammation de la carte C</li><li>- Reprogrammation de la carte D</li><li>- Reprogrammation de la carte E</li></ul>
7	<ul style="list-style-type: none"><li>- Reprogrammation de la carte A</li><li>- Reprogrammation de la carte B</li><li>- Reprogrammation de la carte C</li><li>- Reprogrammation de la carte D</li><li>- Reprogrammation de la carte E</li><li>- Reprogrammation de la carte F</li></ul>
8	<ul style="list-style-type: none"><li>- Reprogrammation de la carte A</li><li>- Reprogrammation de la carte B</li><li>- Reprogrammation de la carte C</li><li>- Reprogrammation de la carte D</li><li>- Reprogrammation de la carte E</li><li>- Reprogrammation de la carte F</li><li>- Reprogrammation de la carte G</li><li>- Reprogrammation de la carte H</li></ul>

Nombre d'argument	Actions
9 ou plus	<ul style="list-style-type: none"> <li>- Reprogrammation de la carte A</li> <li>- Reprogrammation de la carte B</li> <li>- Reprogrammation de la carte C</li> <li>- Reprogrammation de la carte D</li> <li>- Reprogrammation de la carte E</li> <li>- Reprogrammation de la carte F</li> <li>- Reprogrammation de la carte G</li> <li>- Reprogrammation de la carte H</li> <li>- Reprogrammation de la carte I</li> </ul>

Le programme s'adapte aux deux configurations de raspberry.

La capture ci-dessous nous montre une reprogrammation :

```

root@rpi2:/home/pi/project/hex_backup/soutenance_demo# programmer ADC_EXAMPLE1_GFX1.hex
Configuration :(1 pour Raspberry Pi ou 2 pour Raspberry PI2):1
Voulez vous reprogrammer l'ensemble des cartes ? (Y or N):N
Quelle carte voulez vous programmer ? (De A a E):A

*****Debut programmation*****

Using: clk=gpio24, data=gpio23, delay: 0us, baseaddr: 0x00800000 (app-flash)
Actions: program:ADC_EXAMPLE1_GFX1.hex
ok

*****Fin de programmation *****

Erreur

Reussite

La carte A a subi 0 echec avant d'etre reprogramme

Liste peripherique UART :
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3

Liste peripherique USB :
ls: cannot access /dev/ttyAC*: No such file or directory

```

Ce programme s'appuie sur un code de programmation disponible sur GITHUB. Nous avons rencontré des problèmes de fiabilité avec ce programme.

Le programme développé nous permet de résoudre l'ensemble des problèmes.

Exemple de problème rencontré :

L'erreur suivante fut la plus courantes : Fail to rewrite at the adress xxx . Nous avons réalisé des essais pour connaître le nombre d'échec avant réussite. Le tableau ci-dessous vous indique le résultat sur 20 essais différents :

Essai	Carte A	Carte B	Carte C	Carte D
1	0	2	1	1
2	0	2	1	2
3	1	1	1	1
4	1	2	0	1
5	0	3	7	2
6	1	1	0	2
7	1	2	0	0
8	1	0	0	1
9	0	1	2	1
10	0	0	0	0
11	0	2	1	0
12	1	0	1	2
13	1	3	1	0
14	2	1	1	0
15	0	0	1	0
16	0	1	1	1
17	0	1	0	1
18	1	5	1	1
19	1	2	1	1
20	1	1	1	1
21	2	1	8	1

Nous remarquons que le nombre de tentative maximum est de 8 pour cette série d'essai. Cet essai a été réalisé en local.

Lorsque nous sommes sur le réseau nous avons des résultats similaires.

Le programme qui a été développé et qui s'appuie sur l'utilitaire à un taux de réussite de 100 %.

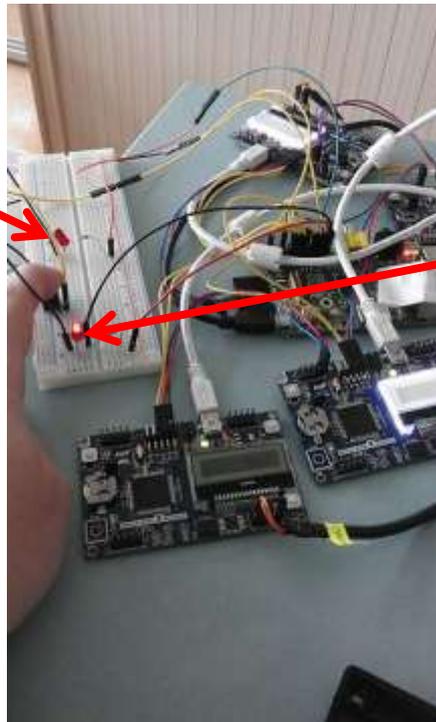
### c) Procédure de remise à zéro

La procédure de remise à zéro doit être utilisée en cas d'erreur de programmation ou d'une détection d'erreur. Ces erreurs sont matérialisées par une led qui reste fixe. Afin de remettre à zéro le nœud il est important de suivre la procédure suivante :

- Appuie sur le bouton poussoir
- Quand la demande sera prise en compte la led s'allumera, vous pouvez relâcher le bouton
- La led sera allumée pendant tout la durée de la procédure
- La Raspberry effectue la remise à zéro
- La led de programmation et la led de remise zéro s'éteint
- Fin de la procédure

A la fin de la procédure vous devez reprogrammer le nœud.

LED de  
programmation



LED de remise  
à zéro

## V- Les données

### a) Capteur

Des capteurs seront implantés sur les cartes petites filles. Voici un panel de capteur que nous pourrions implanter sur le nœud :

- Mesure de température

Fournisseur : Conrad

Référence : Capteur de température numérique

Plage de mesure : -10 C - 60C

Tension d'alimentation : 3-5.5V

Précision : +/- 0.1 C

Consommation : 30 - 60  $\mu$ A



- Mesure du volume sonore

Fournisseur : Zartronic.fr

Référence : Capteur Sonore Analogique

Tension d'alimentation : 5V



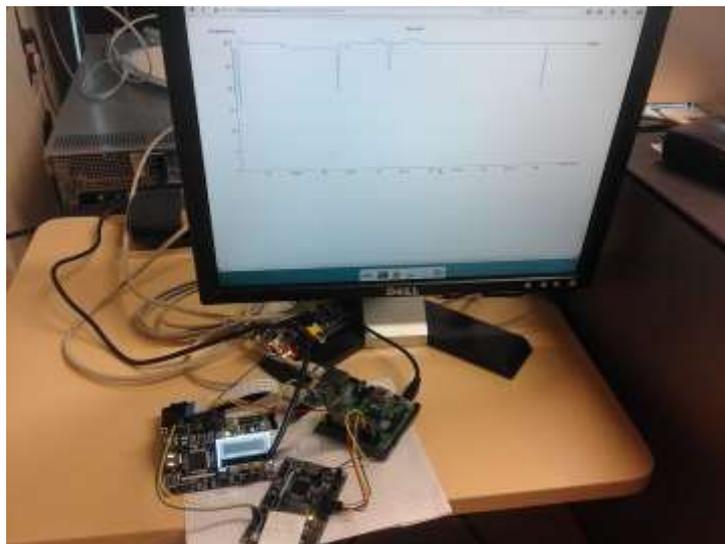
- Mesure de la luminosité

Fournisseur : Conrad

Référence : Photorésistance FW200



Pour l'ensemble des tests réalisés sur la plateforme nous avons simulé les capteurs. Seul le capteur de température a été déployé comme nous le montre l'image suivante.



## b) Format de la trame des données

La mise en forme des données est importante car elle nous permet de connaître la provenance et le type de données que nous manipulons.

Lors de nos essais nous avons utilisé une trame unique. Cette trame a le format suivant :

Mode de transmission	ID	Numéro de la carte fille	Numéro de la carte petite fille	Data
----------------------	----	--------------------------	---------------------------------	------

Descriptif des différents champs

- Mode de transmission :

Ce champ est codé sur 8 bits. Il va nous permettre de connaître la provenance des données. Les valeurs que prendra ce champ sont :

- S pour la liaison Série
- U pour la liaison USB
- R pour la liaison Radio

- ID

Ce champ est codé sur 8 bits. Il nous permet d'identifier la Raspberry qui nous remonte les informations.

- Numéro de la carte fille

Ce champ est codé sur 3 bits. Il nous permet d'identifier la carte fille. Ce champ prendra des valeurs comprises entre 1 et 8.

- Numéro de la carte petite fille

Ce champ est codé sur 1 bit. Il nous permet d'identifier la carte petite fille

- Data

Ce champ est codé sur 8 bits. Il nous permet de connaître les valeurs des capteurs. Les valeurs remontées seront comprises entre 0 et 255

### c) Remonter des données en mode normal

La plateforme est équipée de deux technologies pour que la Raspberry puisse récupérer les data. Nous privilégierons la technologie USB pour le mode normal.

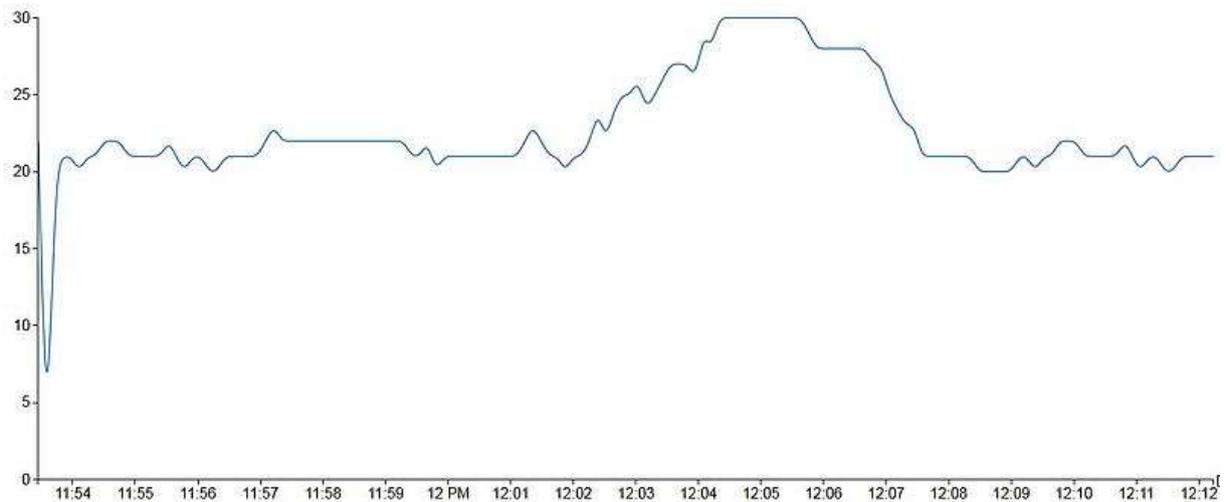
Ce choix nous permet de remonter des données à grande vitesse. En théorie nous devions remonter les données à la vitesse de 12 Mbits. Pour les essais nous avons baissé cette vitesse à 3.5 Mbits.

L'information issue de la simulation des capteurs est contenue dans la trame suivante :

Mode de transmission	ID	Numéro de la carte fille	Numéro de la carte petite fille	Data
----------------------	----	--------------------------	---------------------------------	------

Le détail de la trame a été traité précédemment.

Exemple de données remontées via l'USB



Cette trame ne correspond pas totalement aux besoins de l'utilisateur. Nous privilégierons la trame suivante qui nous permet d'envoyer plus de données dans un temps plus courts :

En tête :

Mode	Mode de transmission	ID
------	----------------------	----

- Mode

Ce champ est codé sur 1 bit. Les valeurs que prendra ce champ sont :

- 0 : Ce mode va nous permettre de remontée une trame à la fois
- 1 : Ce mode va nous permettre de remontée l'ensemble des données d'un nœud en un seul envoie

- Mode de transmission

Ce champ est codé sur 2 bits. Il va nous permettre de connaitre la provenance des données. Les valeurs que prendra ce champ sont :

- 00 pour la liaison Série
- 01 pour la liaison USB
- 10 pour la liaison Radio

- ID

Ce champ est codé sur 8 bits. Il nous permet d'identifier la Raspberry qui nous remonte les informations.

Constitution d'un paquet de donnée

Numéro de la carte fille	Donnée de la carte petite fille 1	Donnée de la carte petite fille 2
--------------------------	-----------------------------------	-----------------------------------

- Numéro de la carte fille

Ce champ est codé sur 3 bits. Les valeurs de ce champ sont :

- 000 : Code pour la carte B (fille 1)
- 001 : Code pour la carte C (fille 2)
- 010 : Code pour la carte D (fille 3)
- 011 : Code pour la carte E (fille 4)
- 100 : Code pour la carte F (fille 5)
- 101 : Code pour la carte G (fille 6)
- 110 : Code pour la carte H (fille 7)
- 111 : Code pour la carte I (fille 8)

- Donnée de la carte petite fille 1

Ce champ est codé sur 8 bits. Il nous permet de connaître les valeurs des capteurs. Les valeurs remontées seront comprises entre 0 et 255

- Donnée de la carte petite fille 2

Ce champ est codé sur 8 bits. Il nous permet de connaître les valeurs des capteurs. Les valeurs remontées seront comprises entre 0 et 255

Exemple :

Trame simple

0	01	001	000 11111111 00110011
Mode	Mode de transmission	ID	Carte B

Trame complète

0	01	001	000 11111111 00110011
Mode	Mode de transmission	ID	Carte B

Suite de la trame

001 11111111 00110011	010 11111111 00110011	011 11111111 00110011
Carte C	Carte D	Carte E

Suite de la trame

100 11111111 00110011	101 11111111 00110011	110 11111111 00110011
Carte F	Carte G	Carte H

Suite de la trame

111 11111111 00110011
Carte I

Nous remontons l'ensemble des informations du nœud avec la trame complète

## d) Remonter des données en mode dégradé

La plateforme est équipée de deux technologies pour que la Raspberry puisse récupérer les data. Nous privilégierons la technologie UART pour le mode dégradé.

Ce choix nous permet de remonter des données à petite vitesse. Les informations sont remontées à la vitesse de 115 200 bauds.

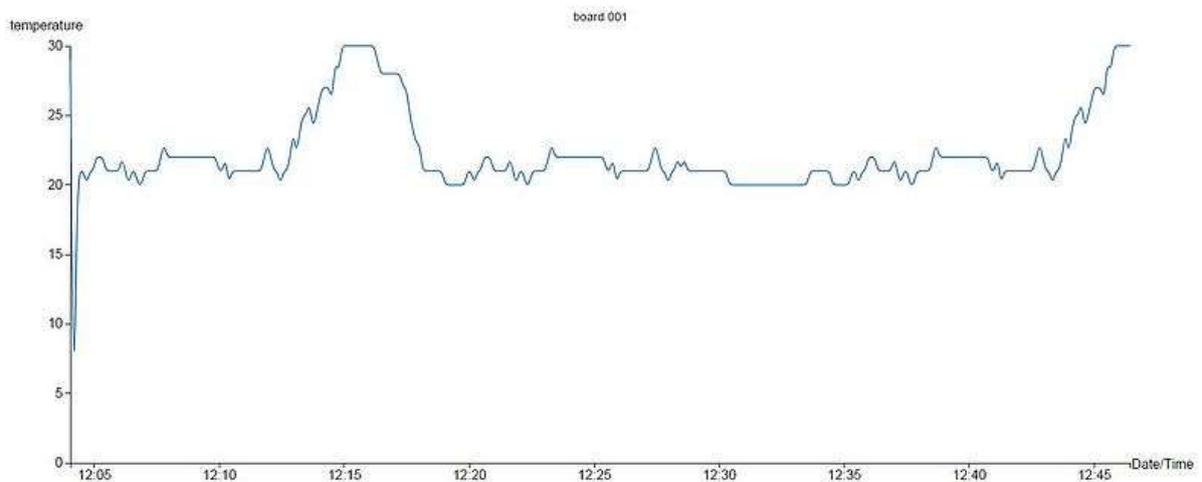
L'information issue de la simulation des capteurs est contenue dans la trame suivante :

Mode de transmission	ID	Numéro de la carte fille	Numéro de la carte petite fille	Data
----------------------	----	--------------------------	---------------------------------	------

Le détail de la trame a été traité précédemment.

Cette trame correspond aux besoins de l'utilisateur.

Exemple de données remontées via l'USART



## VI- Interface web

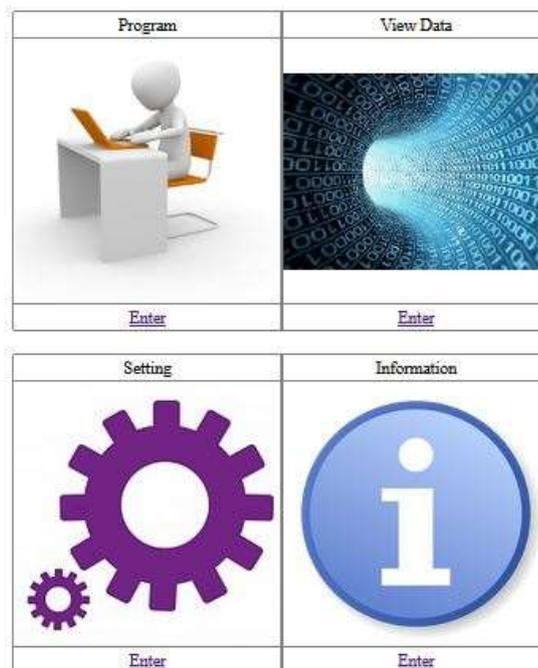
Une fois les informations remontées à la carte mère nous devons les mettre à disposition de l'utilisateur.

L'utilisateur via une interface web pourra consulter les données et reprogrammer les nœuds de capteurs.

L'interface est la suivante :



### Welcome on the platform of demonstration



Elle a été développée en collaboration avec la partie réseaux.

## VII- Pour aller plus loin

La plateforme qui a été réalisée est un prototype. On ne peut pas l'implanter dans un lieu public.

Le prototypage est la première étape avant la réalisation des cartes électroniques.

Nous ne conserverons pas les cartes d'essai qui nous ont permis de réaliser cette plateforme. Elles seront remplacées par des microcontrôleurs de chez Atmel.

Sur la carte mère les modifications seront les suivantes :

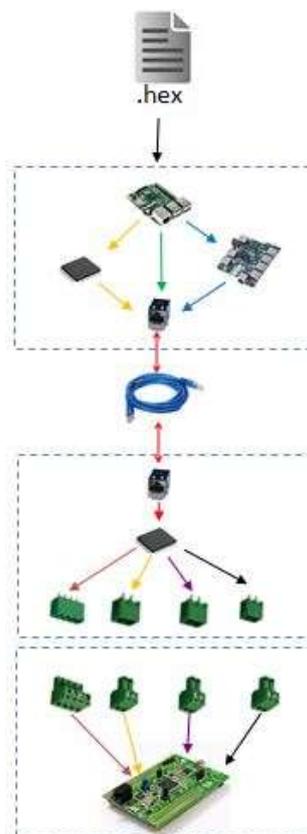
- Remplacement de l'UART par deux microcontrôleurs
- Ajout de la mesure énergétique

Sur la carte fille les modifications seront les suivantes :

- Remplacement des cartes d'essais par des microcontrôleurs
- Ajout de l'interfaçage pour les cartes petites filles

L'interfaçage pour les cartes petites filles seront composé d'un ensemble de connecteur. Les technologies pour les cartes petites filles sont : I2C, SPI et UART

Schéma de la nouvelle plateforme :



## VIII- Bilan

Ce projet de fin d'étude a été consacré aux développements des technologies de demain. En effet de plus en plus de bâtiment accueillant recevant du public seront connectés.

Ce projet avait pour but de réaliser un prototypage d'une plateforme d'expérimentation. Cet objectif a été réalisé avec succès grâce au suivi de mes encadrants. Je les en remercie.

Avant l'implantation dans un lieu public comme la nouvelle bibliothèque universitaire il est important de reprendre l'ensemble du travail réalisé pour la conception et la certification des cartes électroniques.

Sur un point de vue personnel ce projet m'a permis acquérir de nouvelles compétences telles que l'internet des objets, la programmation avec le pdi ...