

Revue finale

**Projet : Surveillance de niveau de cuves cryogéniques
(entreprise 2SP)**



Etudiants :

CROTTE Johann

DEFRANCESCO Anton

STIN Vincent

CELIE Cédric



Académie d'Aix-Marseille – Session 2017

Lycée A. BENOIT



Table des matières :

Table des matières :	2
Partie Commune :	5
1) Présentation de l'entreprise 2SP :	5
2) Présentation du projet :	6
3) Présentation du matériel :	7
4) Présentation du software :	8
a) Le site web :	8
b) La base de donnée :	9
c) La fiche de traçabilité :	10
d) Application d'aide au placement des boitiers :	10
Parties individuelles :	11
Etudiant EC (CROTTE Johann) :	11
I) Présentation du travail à réaliser :	11
1) Présentation du travail réalisé en 2016 :	11
2) Améliorations à amener au projet :	12
3) Diagramme prévisionnel :	13
II) Travail réalisé :	14
1) Améliorations réalisées sur la carte d'extension :	14
2) L'horloge temps réel DS3231MZ+ :	17
3) Le convertisseur analogique numérique MCP3427 :	18
4) Le module Bluetooth Low Energy RN4020 :	23
III) Conclusion sur le travail réalisé :	24
1) Diagramme réel :	24
2) Conclusion et connaissance acquise :	25
IV) Annexes Johann Crotte :	26
1) Nomenclature des composants :	26
2) Mise en œuvre DS3231MZ+ sur RaspBerry Pi2 (<i>jessie</i> et <i>jessielight</i>) en I2C :	27
3) Fiches de lecture croisée sur la mise en œuvre de l'horloge temps réel DS3231MZ :	32
4).....	32
Programme convertisseur analogique numérique :	33
5) Organigramme correspondant au programme :	34
6) Journal de bord :	35
7) Compte rendu de réunion pour la revue 2 :	40
8) Informations concernant les cuves cryogéniques :	41
Etudiant IR1 (DEFRANCESCO Anton) :	42
I) Présentation de mes tâches :	42



1) Objectifs :	42
a) Rappel sur l'architecture globale et les problématiques :	42
b) Planification prévisionnelle :	47
2) Partie Physique : Bluetooth module RN4020 :	49
Comment fonctionne le Bluetooth :	49
3) Travail accompli dans les itérations précédentes :	51
a) Découverte des langages utilisés sur le projet :	51
b) Mise en place de l'application web existante :	52
c) Simulateur de données et transfert Bluetooth :	53
d) Le programme Simulateur :	53
e) Le programme Moniteur :	54
f) Prise en main et amélioration de l'application web :	54
II) Travail accompli dans la troisième itération :	60
1) Création d'une page de vue d'ensemble des cuves :	60
a) Défaut de l'application préexistante :	60
b) Mise en œuvre :	61
2) Conclusion :	63
a) Bilan :	63
b) Pistes d'amélioration :	63
III) Annexes :	64
1) Code de la page infoscuve.html.twig :	64
2) Journal de bord :	71
Etudiant IR2 (STIN Vincent) :	73
I) Rappel du projet :	73
II) Présentation de mes tâches :	75
III) Les diagrammes :	75
IV) Planification prévisionnelle :	77
V) Travail effectué pour les revues 1 et 3:	78
Itération 1 : Prise en main de PHP sur mon PC.	78
Itération 2 : Accès aux données de la BDD depuis PDO.	80
Itération 3 : Génération d'un PDF statique.	81
Itération 4 : Génération d'un PDF avec des données issues d'une BDD.	82
Itération 5 : Portage sur Raspberry Pi.	84
VI) Travail effectué pour la revue finale :	85
Itération 6 : Intégration au site existant.	85
VII) Partage des informations dans le groupe :	92
VIII) Conclusion :	94
IX) Annexes :	95
1) Journal de bord :	95
2) Documentation des fonctions FPDF utilisées :	102



Etudiant IR3 (CELIE Cédric) :	115
I) Présentation des mes tâches :	115
1) Objectifs :	115
2) Rappel sur l'architecture globale et les problématiques :	116
3) Planification prévisionnelle :	117
II) Travail accompli dans les itérations précédentes :	117
1) Découverte de l'écosystème Node.js :	117
a) Node.js + Express.js :	117
b) Module liaison série :	119
c) Module accès BDD :	120
2) Application d'optimisation de placement du module central :	121
a) Synoptique de l'architecture de l'application :	121
b) Fonctionnement global :	121
c) Mise en Oeuvre :	123
c.1) Détails de l'implémentation :	123
c.2) Principe de la détection du RSSI :	124
d) Problèmes rencontrés :	125
III) Travail accompli dans l'itération 3 :	127
1) Evolution application Node.js :	127
a) Défauts de l'application et ses solutions :	127
2) Mise en oeuvre :	127
3) Accès sécurisé à l'appli web de supervision des cuves :	128
a) Présentation du VPN :	128
b) Principe de fonctionnement global :	129
c) Présentation OpenVPN :	130
d) Mise en oeuvre :	130
e) Installation PiVPN :	131
f) Paramétrage au niveau de l'installation :	131
g) Test depuis le client "OpenVPN Connect" :	132
IV) Conclusion :	133
1) Bilan :	133
2) Pistes d'améliorations :	134
V) Annexes :	135
Journal de bord :	135



Partie Commune :

1) Présentation de l'entreprise 2SP :

Riche d'une expérience de plus de 20 ans en électronique, Dominique Morin est le fondateur de 2SP Electronic pour accompagner les entreprises vers des solutions électroniques pratiques, efficaces qui leur permettent d'optimiser leurs outils de production, de test, de contrôle et/ou de maintenance, en toute confiance.

L'entreprise est dans le secteur d'activité de l'informatique et de l'électronique à Moulton en Normandie.



Le bureau d'études 2SP Electronic conçoit le système électronique qui répond aux problématiques d'entreprises :

- Analyse de cahier des charges
- Etudes, conception et développement électronique
- Industrialisation et production
- Service après-vente et maintenance sur site

2SP Electronic conçoit et développe des systèmes électroniques sur mesure. De l'analyse aux besoins, jusqu'à la réalisation de projets, l'entreprise livre un projet clé en main.

L'entreprise nous a donc proposé de participer à un projet déjà existant afin de l'améliorer.



2) Présentation du projet :

Notre projet doit être la poursuite et le développement d'un système développé l'année passée par les étudiants de seconde année en partenariat avec l'entreprise 2SP Electronic.

Ce projet a pour but de gérer le système de surveillance du niveau d'un ensemble de cuves cryogéniques qui servent à l'entreprise à stocker des produits pour ses clients.



Il est important pour l'entreprise d'avoir une surveillance fiable du niveau de ses cuves, en effet, il se peut que les différents contenants puissent s'évaporer dû à un changement de température ou même d'une défaillance du système hermétique. Si c'est le cas, il faut que le propriétaire ou le superviseur puisse en être informé et regarder si le niveau reste stable au fil du temps. Cela assure au client que son produit est bien entreposé et qu'il peut avoir confiance car il peut lui-même regarder si tout va bien. Quant à l'entreprise, cela lui permet de garantir une sécurité et une supervision optimum.

La tâche principale que nous devons effectuer consiste à surveiller les niveaux d'un parc de cuves qui contiennent de l'azote liquide. La mesure du niveau ne fait pas partie du projet. Le projet ne concerne que l'exploitation des différentes mesures.

Chaque cuve est équipée d'un système de mesure de niveau, cela représente le coffret « Périphérique » (composé d'une carte RaspBerry PI 2 et d'une carte d'extension avec module Bluetooth RN4020, d'une sonde de niveau, d'une horloge en temps réel et d'un convertisseur Analogique/Numérique). Chaque cuve doit transmettre à intervalle de temps régulier une mesure de niveau par liaison sans fil BLE (Bluetooth Low Energy) au Coffret « Maître » (composé d'une carte RaspBerry PI 3, d'une carte d'extension avec module Bluetooth Low Energy RN4020 et d'une horloge en temps réel). Celui-ci les recueille et les stockent. Les cartes auront 2 modes, le premier (mode normal) servira à surveiller les cuves et le second (mode optimisation) servira à vérifier l'optimisation du placement des cuves. Les modes pourront être changés grâce à un cavalier.

Une surveillance à distance doit être possible (site embarqué, alerte par mail) pour que le client ou les superviseurs puissent surveiller le niveau à tout moment tout en étant sécurisé par un cryptage des données qui permettra l'accès aux informations aux seules personnes concernées par celles-ci. Il faut toutefois que le système ait une consommation la plus faible possible.

L'objectif du projet est qu'il puisse être transposable dans un autre contexte que la mesure de niveau des cuves et qu'il devienne un service de supervision générique proposé par 2SP à ses clients.



3) Présentation du matériel :

L'horloge temps réel DS3231MZ

L'utilité de l'horloge temps réel (**DS3231MZ+**) est de lors du redémarrage du coffret périphérique pour l'envoi des données qui a lieu fréquemment la Raspberry puisse envoyer les informations de la cuve ainsi que l'heure à laquelle les données ont été relevées. Le coffret périphérique n'étant pas connecté à internet il fallait donc trouver une solution. Une horloge temps réel reliée à une pile faisant office de montre est sur laquelle la Raspberry devait s'appuyer pour avoir l'heure lors du redémarrage.

Le Convertisseur analogique numérique (CAN) MCP3427

Le convertisseur MCP3427 permettra de récupérer le niveau de la cuve que l'on recevra par un signal analogique, de le convertir en numérique afin de pouvoir le transmettre via le module Bluetooth au coffret maître.

Nous ne nous occupons pas de l'acquisition de la valeur du niveau qui sera superviser par l'entreprise. Cette donnée sera récupérée par un bornier à 3 entrées pour être traité par la suite.

Le composant est imposé par le client, il était trop petit pour faire les mesures directement dessus, il a donc fallu fabriquer une carte pour écarter les broches et ainsi pouvoir commencer les tests.

Le module Bluetooth RN4020 BLE (Bluetooth low energy)

Le module Bluetooth sera sur le coffret maître et périphérique et permettra donc une transmission sans fil des informations du niveau des cuves converties en format numérique ainsi que l'heure à laquelle la mesure a été effectuée des cartes périphériques à la carte maître pour ainsi pouvoir réunir les informations dans une base de données.

L'avantage de ce composant est sa taille ainsi que comme son nom l'indique sa faible consommation.



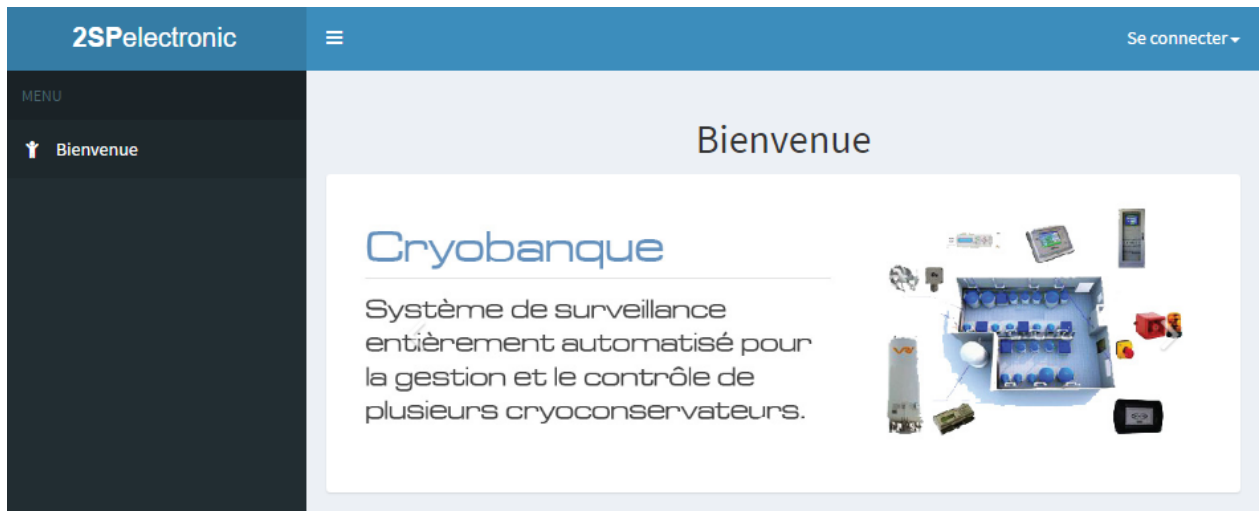
Mémoire HAT

La mémoire HAT a été demandée par le client pour garder les données en mémoire si la transmission n'est pas instantanée.



4) Présentation du software :

a) Le site web :



Le site web est l'application permettant de gérer et de monitorer les cuves. Celui-ci est embarqué sur la RaspBerry du coffret maître et est accessible à l'intérieur comme à l'extérieur du site de stockage. Son utilisation limitée par des comptes utilisateur. Il faut en effet être connecté en tant qu'administrateur pour avoir accès à ses fonctions.

Une fois connecté on peut accéder à des pages web.

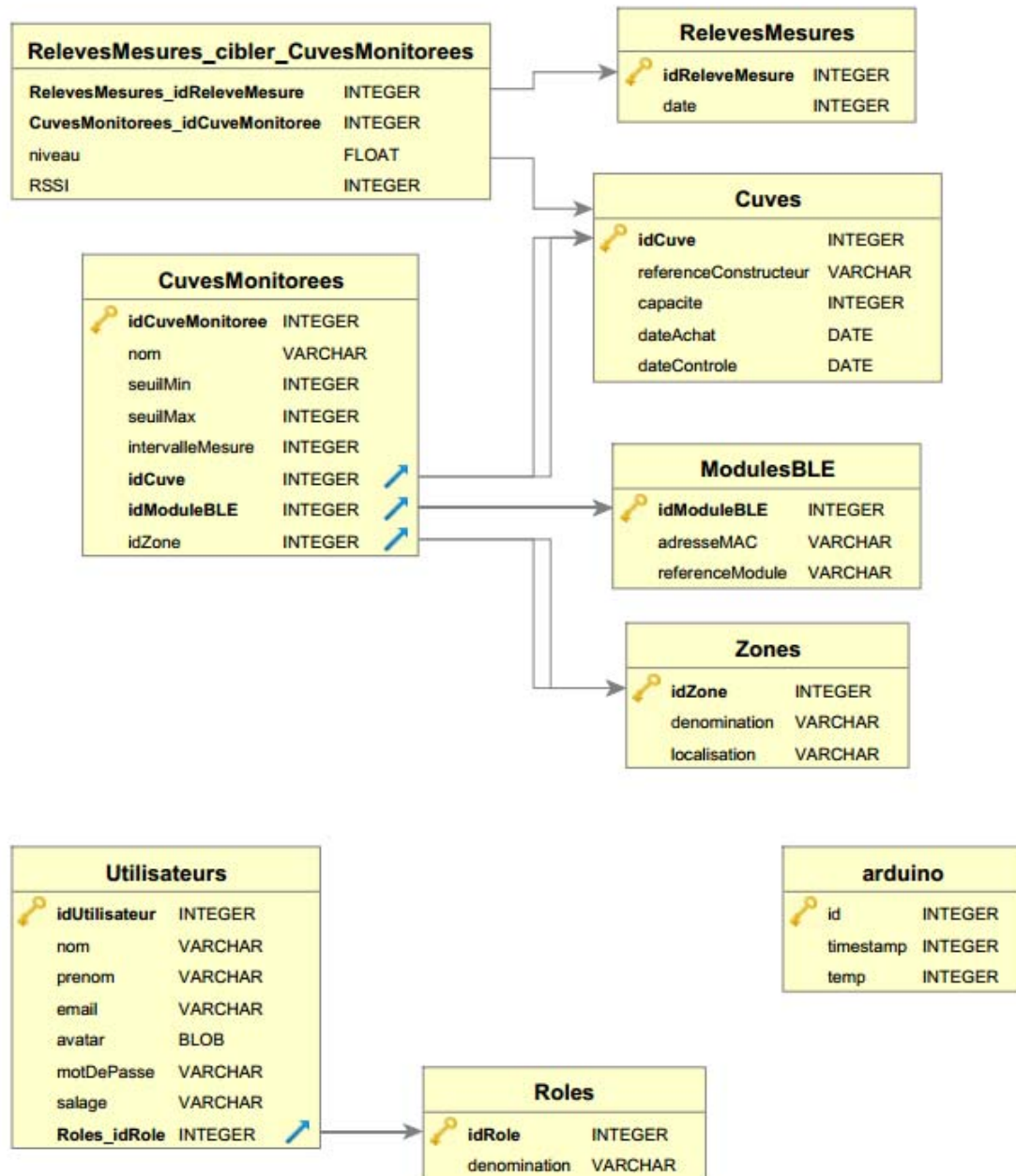


Ainsi un onglet d'administration permet d'ajouter des cuves, des modules Bluetooth ou des utilisateurs.

Un onglet graphique permet de visualiser les derniers changements de niveau des cuves surveillée.

Un dernier onglet permet de créer des fichiers de traçabilité qui servent de fiche d'historique des niveaux des cuves.

b) La base de donnée :



c) La fiche de traçabilité :

Le fichier de traçabilité permet de faire un listing des derniers relevés de niveaux des cuves avec un exemple ci-dessous.



Fichier de traçabilité du 02/05/2017

Date/Heure	Cuve	Zone	Niveau
02/03/2017 16:07:01	B 2003 M	0	15.5
02/03/2017 16:07:01	Cuve n°1	0	36.76
02/03/2017 16:07:01	CuveTest	0	21.88
02/03/2017 16:07:01	Cuve n°2	0	67.61
02/03/2017 16:08:01	B 2003 M	0	20.27
02/03/2017 16:09:01	B 2003 M	0	15.61
02/03/2017 16:09:01	Cuve n°1	0	22.07
02/03/2017 16:10:01	B 2003 M	0	17.32
02/03/2017 16:10:01	CuveTest	0	28.56
02/03/2017 16:11:01	B 2003 M	0	11.9
02/03/2017 16:11:01	Cuve n°1	0	49.08
02/03/2017 16:12:01	B 2003 M	0	15.02
02/03/2017 16:12:01	Cuve n°2	0	97.66

d) Application d'aide au placement des boîtiers :

D'après le cahier des charges, l'entreprise 2SP nous a demandé le développement d'une application d'optimisation de placement des boîtiers Maître et Périphériques par analyse du niveau de réception Bluetooth RSSI.

Cette application nous permettra d'avoir la meilleure réception possible entre l'ensemble des cuves et le boîtier Maître. Le principe de fonctionnement est le suivant :

En déplaçant le boîtier Maître nous pourrons voir la réception des cuves sur un smartphone, une tablette ou un ordinateur. Nous pourrons donc le placer au mieux en fonction des résultats de réceptions.

L'intensité de réception sera affiché grâce à des barres de progression et en chiffres numériques.



Parties individuelles :

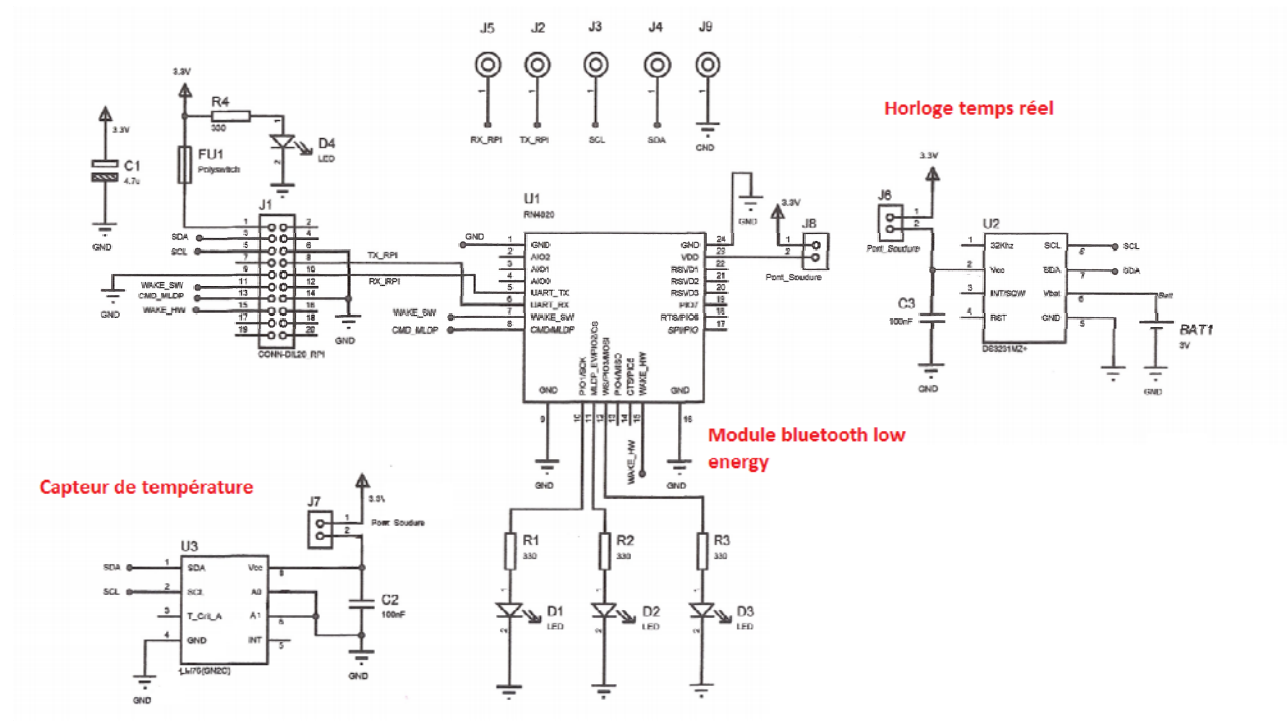
Etudiant EC (CROTTE Johann) :

I) Présentation du travail à réaliser :

1) Présentation du travail réalisé en 2016 :

Le Projet 2SP réalisé en 2016 avait pour but de transférer des données par Bluetooth entre 2 RaspBerry pi avec l'heure de ce transfert ,les données étaient générées par un capteur de température ,Ce projet était destiné à la société 2sp Electronic.

Schéma Réalisé en 2016 :



Ils avaient ainsi réalisé une carte d'extension ainsi qu'une partie logicielle permettant le bon fonctionnement du projet .

Le projet avait plus pour but un travail sur les technologies utilisées et n'était pas mis en conditions réelles mais en s'en approchant.

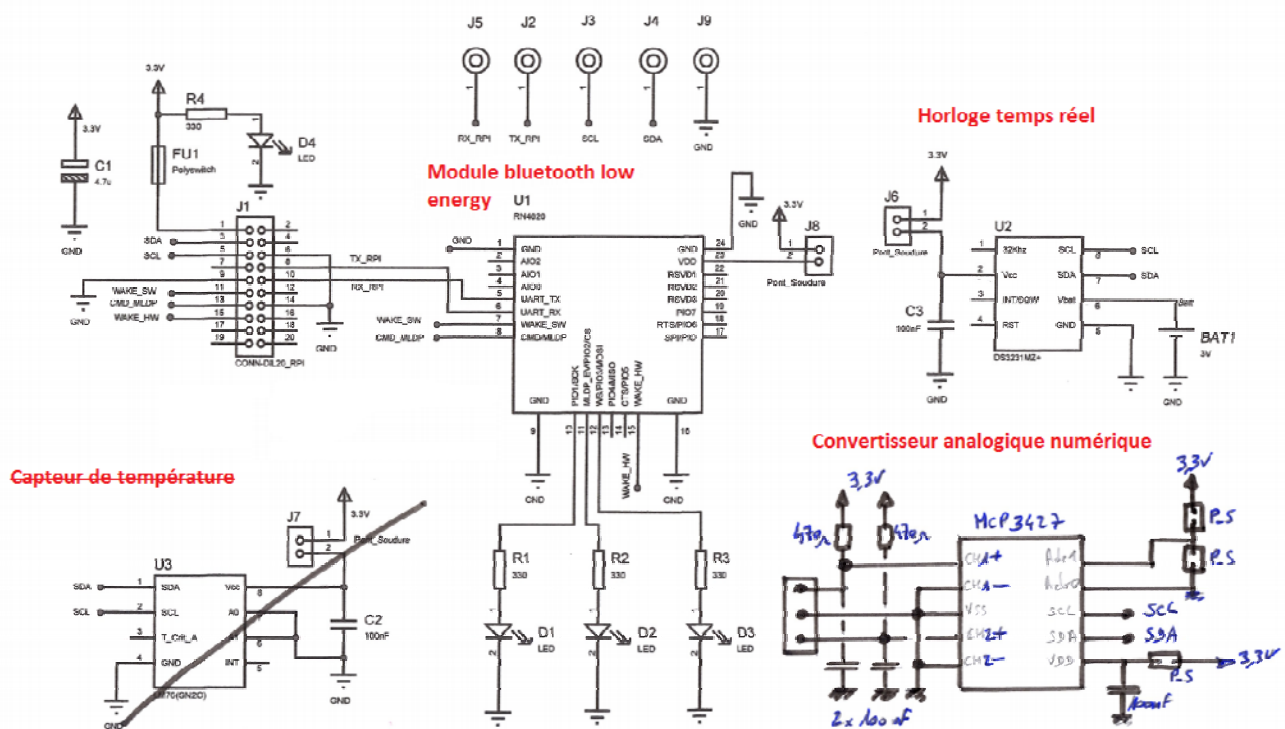


2) Améliorations à amener au projet :

Le projet de cette année à plus pour but une application en conditions réelles

Le client a donc proposé un nouveau schéma avec un convertisseur analogique numérique et un bornier qui seront là pour récupérer et convertir les informations du niveau des cuves donc l'acquisition de ces valeurs sera réalisée par le client ce système remplacera le capteur de température qui était là pour simuler une valeur .

Schéma proposé par le client :



Le projet comprend également une option optimisation de positionnement des cuves , un cavalier sera raccordé à une broche du GPIO pour une programmation de ce changement de mode.

Une librairie pour le Module Bluetooth devra être créé pour réaliser le transfert entre les RaspBerry pi

Pour le convertisseur il sera nécessaire de réaliser une librairie pouvant réaliser plusieurs conversions et prendre en compte le transfert des données

Pour l'horloge temps réel en plus de l'envoi de la données 'heure' il faut réaliser une synchronisation entre l'heure de la RaspBerry pi Centrale qui sera à l'heure juste d'internet et avec la RaspBerry pi Périphérique sur les cuves en cas de décalage de l'heure des deux RaspBerry pi malgré la présence de la pile qui alimente l'horloge .

Une mémoire HAT EEPROM sera également présente pour garder la configuration des liaisons entre la Raspberry pi et les composants

C'est une sonde PT1000 à l'intérieur de la cuve qui relèvera le niveau
Sa résistance change en fonction de la température ce qui permet d'obtenir une tension en
sortie dépendant du niveau de la cuve (Annexe information cuves P26)



3) Diagramme prévisionnel :

1		→	▲ Projet Finale	189 h	Lun 09/01/17	Lun 19/06/17
2		→	▲	57 h	Lun 09/01/17	Jeu 02/03/17
3	✓	→	Planification du projet (création du Gant+drive)	8 h	Lun 09/01/17	Mar 10/01/17
4		→	Recherches sur le DS3231	7 h	Jeu 12/01/17	Mar 17/01/17
5		→	essai de la mise en œuvre sur l'horloge temps réel 2016	8 h	Mar 17/01/17	Lun 23/01/17
6		→	Création nouvelle mise en œuvre sur l'horloge temps réel pour la bonne version de raspbian	12 h	Lun 23/01/17	Mar 31/01/17
7		→	Recherche sur le convertisseur Analogique numérique + test	8 h	Mar 31/01/17	Lun 06/02/17
8		→	Rédaction librairie pour le CAN	12 h	Mar 07/02/17	Jeu 02/03/17
9		→	▲ Pour revue 2	65 h	Jeu 09/03/17	Lun 08/05/17
10		→	Evolution schéma de la carte réalisée en 2016 puis routage	40 h	Jeu 09/03/17	Mar 04/04/17
11		→	Recherche + librairie sur le module bluetooth RN4020	25 h	Jeu 06/04/17	Lun 08/05/17
12		→	▲ Pour revue final	60 h	Lun 08/05/17	Lun 19/06/17
13		→	Rassembler les parties	30 h	Lun 08/05/17	Lun 29/05/17
14		→	rédaction documents a rédigé	10 h	Lun 29/05/17	Lun 05/06/17
15		→	Test projet	20 h	Lun 05/06/17	Lun 19/06/17



II) Travail réalisé :

1) Améliorations réalisées sur la carte d'extension :

IBD et Schéma proteus ISIS :

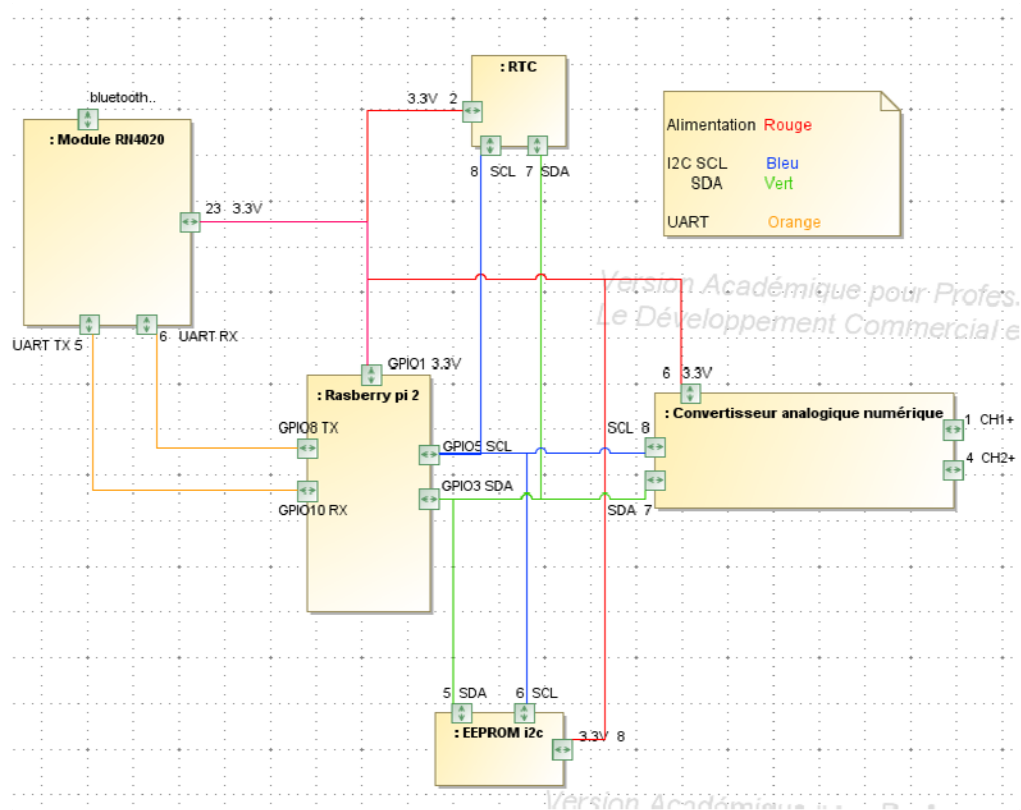
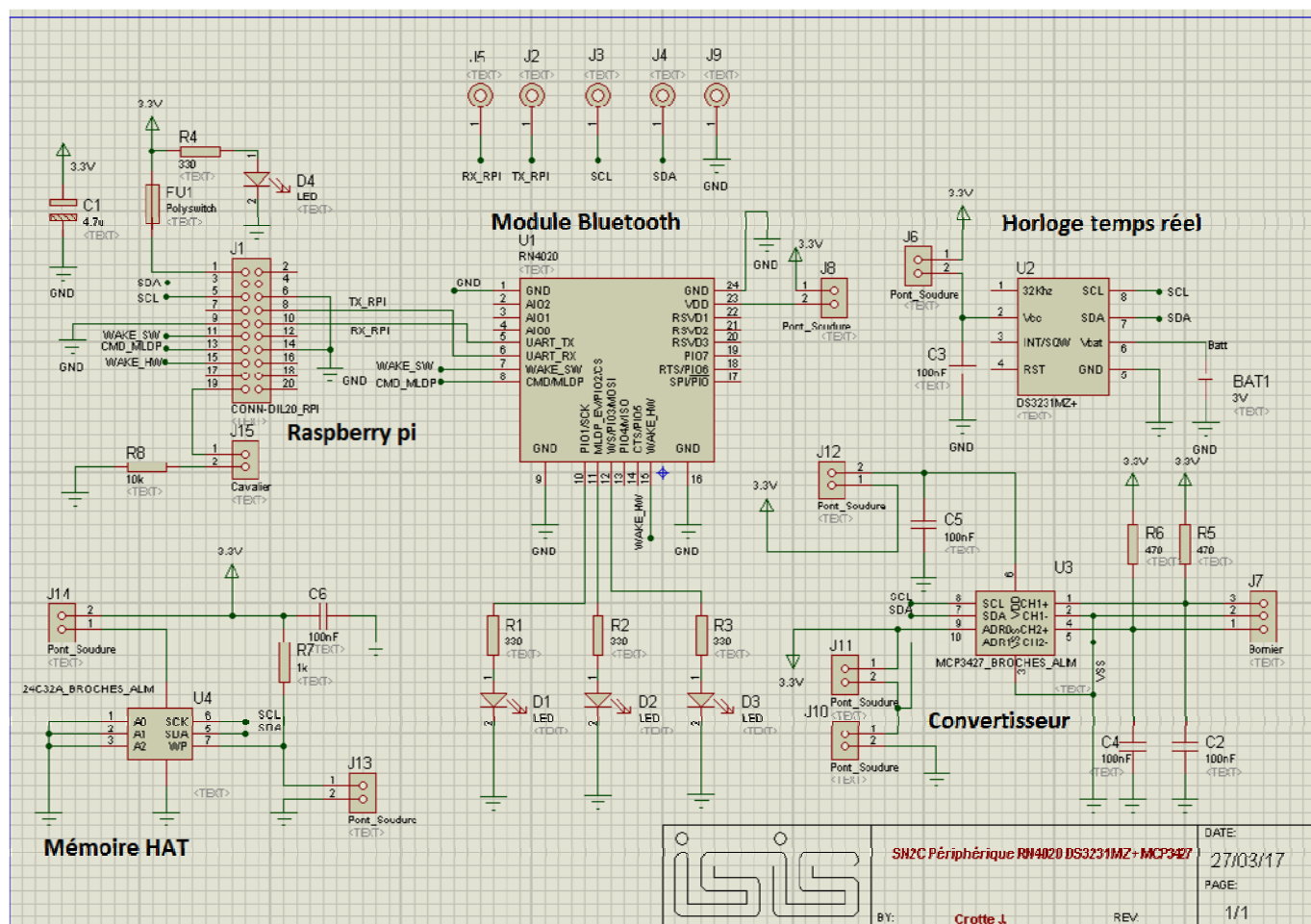
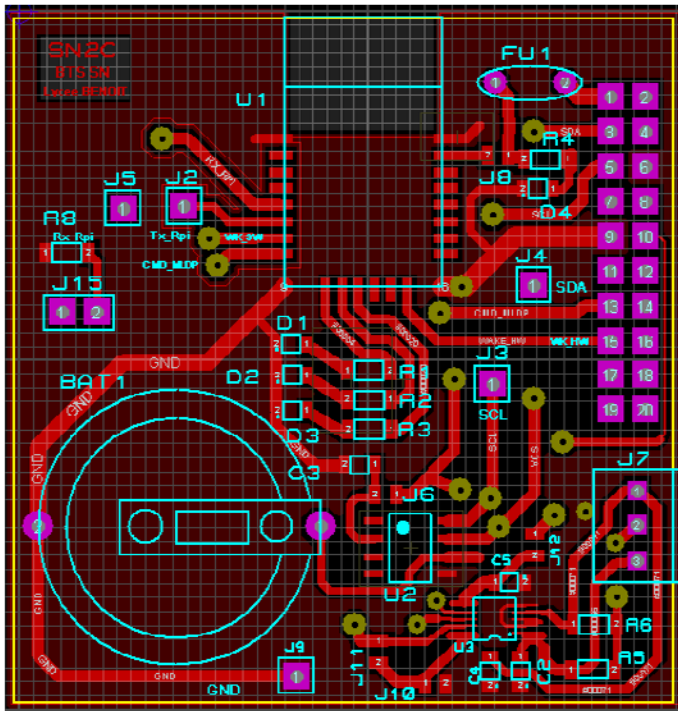


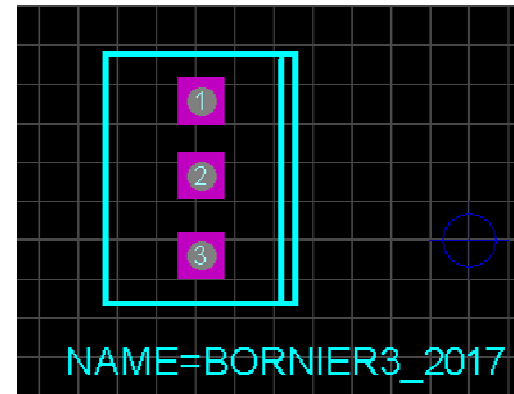
Schéma ARES :

Coté TOP :

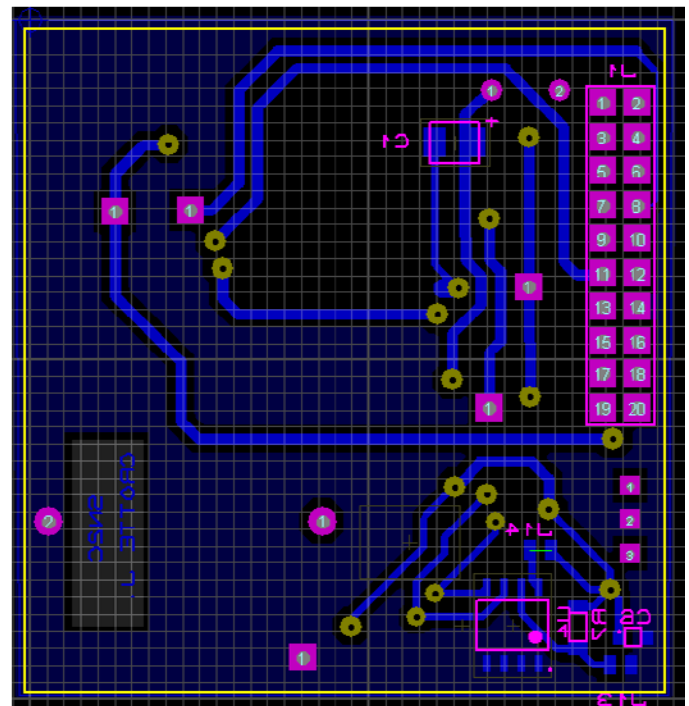
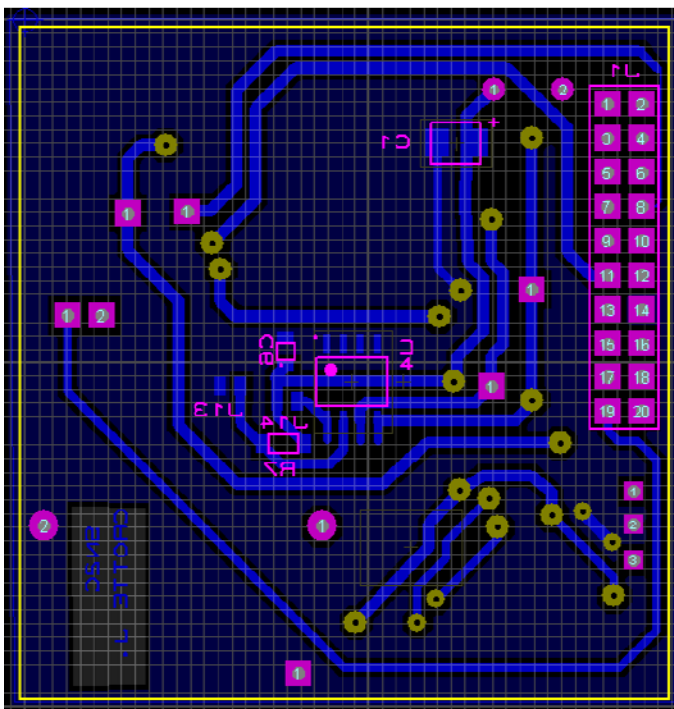




Pour le schéma ARES j'ai du réaliser l'empreinte du bornier avec les dimensions de la documentation et les informations des broches pour réaliser une empreintes conforme et des pastilles et le perçage correspondant



Coté *BOTTOM*



J'avais proposé deux versions concernant la position de la mémoire HAT .La première a été choisie car la mémoire HAT en bas en droite aurait crée un surplus de composant est un déséquilibre lors de la soudure .

J'ai donc réalisé le schéma proposé par le client avec l'ajout du cavalier de changement d option

Ainsi que le routage pour cela je suis reparti du schéma de l'année 2016 au même format et en gardant les composants similaires aux mêmes emplacements .

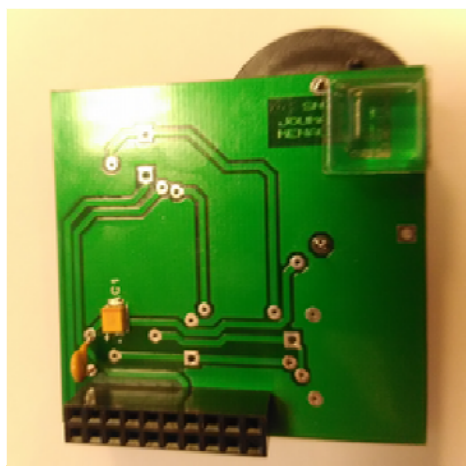
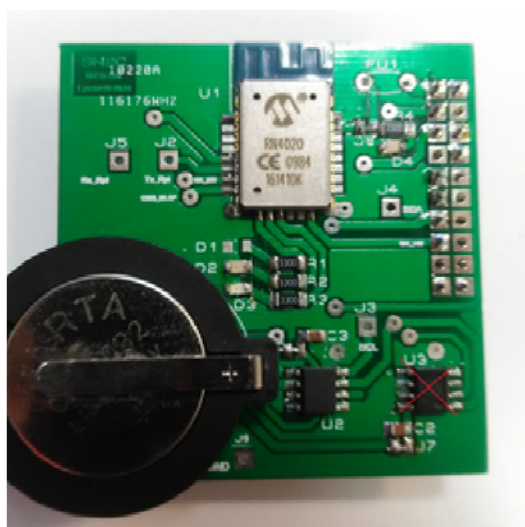
La carte a été commandé sur sitestudio .Il suffit d'envoyer le fichier Gerber crée a partir des fichiers Proteus .Les paramètres proposés par défaut nous convenait .La carte serait donc sur les 2 couche TOP et BOTTOM et sera fabriquée et livrée dans les deux semaines suivantes

La plupart des composants ont été commandés sur RS (voir Annexe Nomenclature p26)

Développement de la carte

Carte réalisé en 2016

- Module Bluetooth
- Capteur de température
- RTC

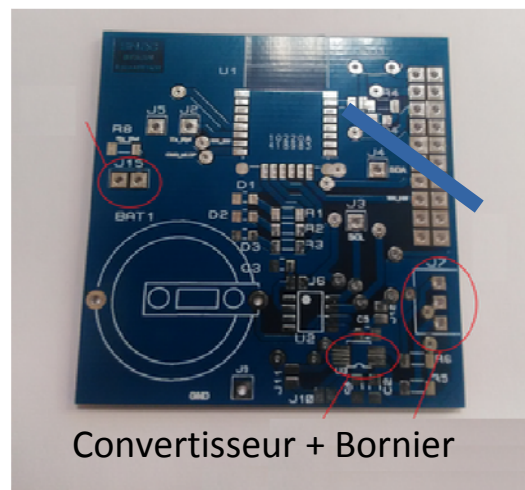


Carte réalisé en 2017

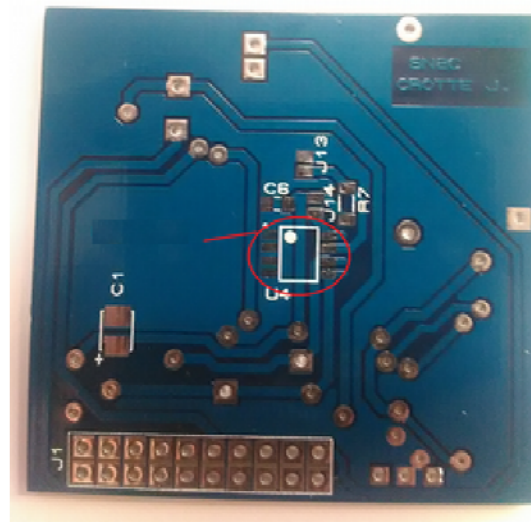
- Module Bluetooth
- CAN + BORNIER
- RTC
- Cavalier pour le mode optimisation
- Mémoire HAT

----->

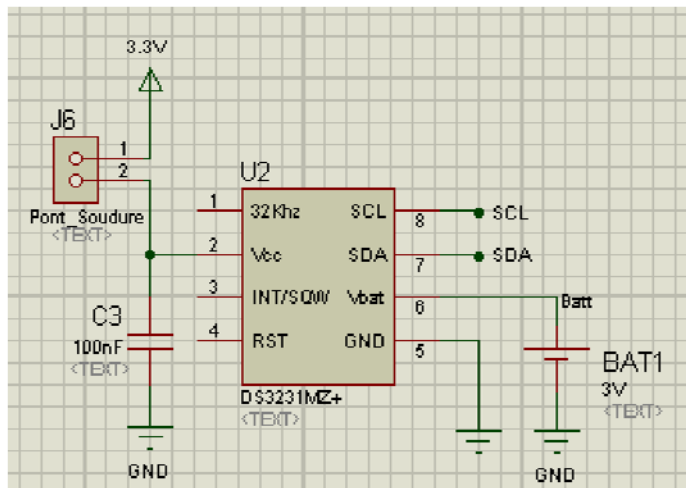
Cavalier



Convertisseur + Bornier



2) L'horloge temps réel DS3231MZ+ :

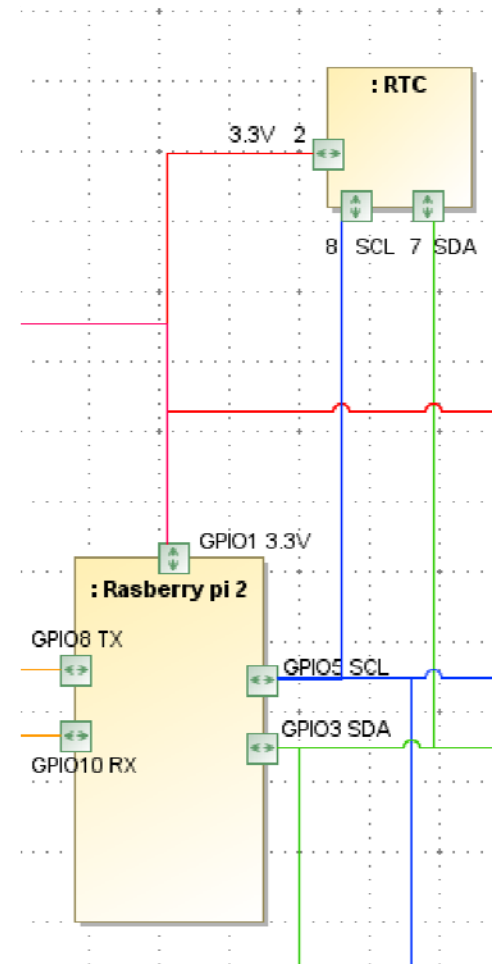


L'Horloge temps réel DS3231MZ+ est la même que celle utilisée l'année précédente

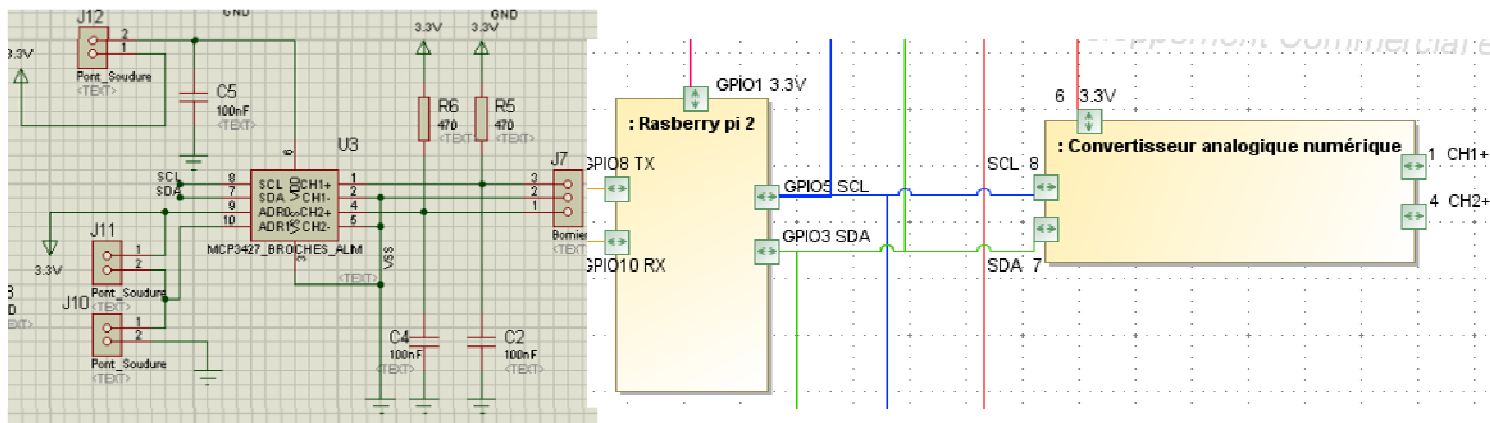
Pour que la Raspberry pi prenne l'heure sur l'Horloge et non sur ses registres d'origine il faut y faire quelques changements, j'ai donc réalisé un tutoriel basé sur ces changements inspirés de la mise en œuvre de 2016 pour une Raspberry pi sous Jessie (voir annexes p29) je dispose aussi de la carte de l'année précédente sur laquelle j'ai effectué les tests que j'ai par la suite effectué sur ma propre carte et elle fut bien reconnue par la Raspberry pi à l'adresse i2c 0x68z

`i2cget -y 1 0x68 0x06 b` permet d'obtenir l'année
`i2cget -y 1 0x68 0x05 b` permet d'obtenir le mois
`i2cget -y 1 0x68 0x04 b` permet d'obtenir le jour
`i2cget -y 1 0x68 0x03 b` permet d'obtenir l'heure
`i2cget -y 1 0x68 0x02 b` permet d'obtenir les minutes
`i2cget -y 1 0x68 0x01 b` permet d'obtenir les secondes

Cette mise en œuvre a été donnée à 2 élèves pour un test (voir Annexes p32)



3) Le convertisseur analogique numérique MCP3427 :



Le convertisseur MCP3427 permettra de récupérer le niveau de la cuve et de le convertir afin de pouvoir le transmettre via le module Bluetooth au coffret maître
Ce composant a pour particularité de ne pas être un convertisseur à approximation successive mais un convertisseur sigma-delta qui permet une meilleure résolution

Le composant est imposé par le client, Il était trop petit pour faire les mesures directement dessus il a donc fallu fabriquer une carte pour accéder aux broches les broches et ainsi pouvoir commencer les tests. Cette carte a été réalisée dans l'enceinte du lycée

Schéma Proteus isis

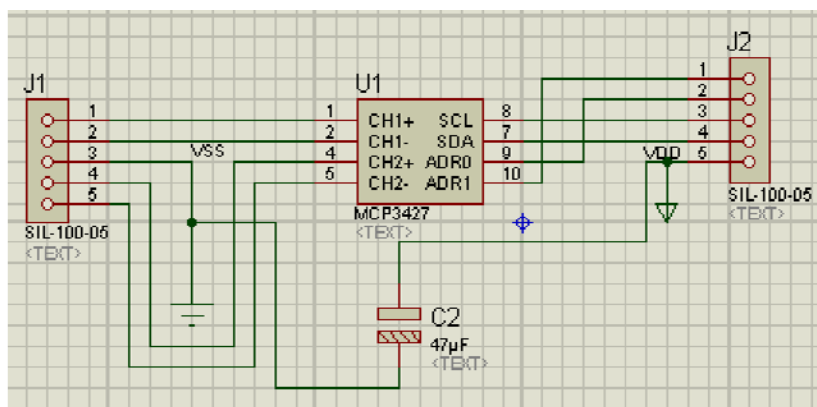
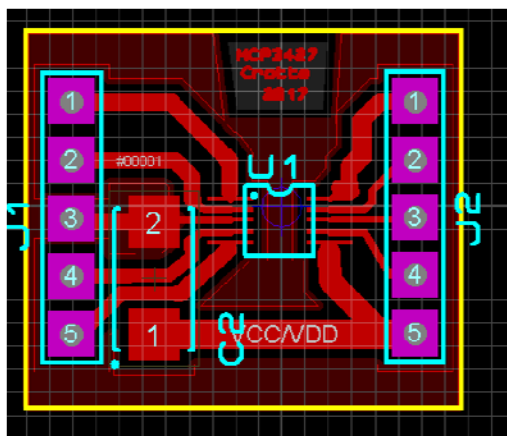
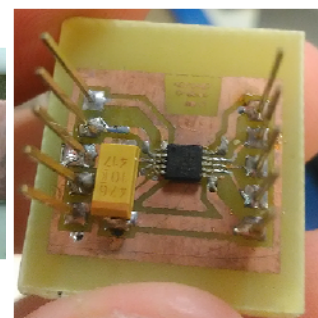
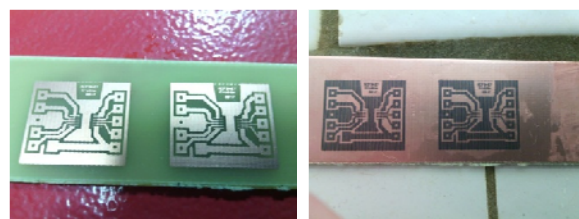


schéma Proteus Ares



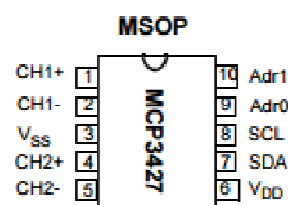
Gravure ,perçage et soudure au lycée



Début des test sur la petite carte

Les deux broches d'adresse seront reliées au 3,3 v pour avoir un niveau haut et donc ainsi mettre l'adresse i2c du CAN à 6E
Ce choix de cette adresse a été réalisé car l'adresse par défaut était 0x68 qui était

Equipe projet 2SP



identique à l'adresse de la RTC

Le niveau des cuves sera simulé par un potentiomètre câblé sur l'alimentation

un petit programme a été fait pour les premiers tests affichant seulement la valeur de CH1+

La librairie bmc2835 est nécessaire pour compiler les programmes qui suivent

```
void init_BCM2835()
{
  unsigned int slave_address=0x6e;
  if (!bcm2835_init())
  {
    printf("bcm2835_init failed. Are you running as root??\n");
  }
  if (!bcm2835_i2c_begin())
  {
    printf("bcm2835_i2c_begin failed. Are you running as root??\n");
  }
  bcm2835_i2c_setSlaveAddress(slave_address);
  bcm2835_i2c_setClockDivider(clk_div);
}
```

I ² C Device Address Bits			Logic Status of Address Selection Pins	
A2	A1	A0	Adr0 Pin	Adr1 Pin
0	0	0	0 (Addr_Low)	0 (Addr_Low)
0	0	1	0 (Addr_Low)	Float
0	1	0	0 (Addr_Low)	1 (Addr_High)
1	0	0	1 (Addr_High)	0 (Addr_Low)
1	0	1	1 (Addr_High)	Float
1	1	0	1 (Addr_High)	1 (Addr_High)
0	1	1	Float	0 (Addr_Low)
1	1	1	Float	1 (Addr_High)
0	0	0	Float	Float

Les PIN Adr0 et Adr1 sont à l'état haut ce qui crée l'adresse i2c « 6e »

```
double getMesure()
{
  int temp=0;
  double mesure = 0;

  char i2cOut[2];
  char i2cIn[3];

  i2cOut[0]=0x88;
  bcm2835_i2c_write(i2cOut, 1); // Configuration en 16bits et mesure sur CH1+
  bcm2835_i2c_read(i2cIn, 3);
  mesure = (256*i2cIn[0] + i2cIn[1])*3.3/32767;

  return mesure;
}
```

L'octet de configuration est en mode mode one shot sur la channel 1 en mode 16 bit sans gain « 10001000 » « 0x88 »

Le quantum pour une alimentation de 3.3v de la RaspBerry pi pour une résolution de 16 bit

bit 7	RDY: Ready Bit This bit is the data ready flag. In read mode, this bit indicates if the output register has been updated with a latest conversion result. In One-Shot Conversion mode, writing this bit to "1" initiates a new conversion. Reading RDY bit with the read command: 1 = Output register has not been updated. 0 = Output register has been updated with the latest conversion result. Writing RDY bit with the write command: Continuous Conversion mode: No effect One-Shot Conversion mode: 1 = Initiate a new conversion. 0 = No effect.
bit 6-5	C1-C0: Channel Selection Bits 00 = Select Channel 1 (Default) 01 = Select Channel 2 10 = Select Channel 3 (MCP3428 only, treated as "00" by the MCP3426/MCP3427) 11 = Select Channel 4 (MCP3428 only, treated as "01" by the MCP3426/MCP3427)
bit 4	0/C: Conversion Mode Bit 1 = Continuous Conversion Mode (Default). The device performs data conversions continuously. 0 = One-Shot Conversion Mode. The device performs a single conversion and enters a low power standby mode until it receives another write or read command.
bit 3-2	S1-S0: Sample Rate Selection Bit 00 = 240 SPS (12 bits) (Default) 01 = 60 SPS (14 bits) 10 = 15 SPS (16 bits)
bit 1-0	G1-G0: PGA Gain Selection Bits 00 = x1 (Default) 01 = x2 10 = x4 11 = x8

Tests

En faisant varier le potentiomètre le chiffre varie de 0 à +3,3 V

```
Valeur de CH1+ = 1.53715 V
Valeur de CH1+ = 1.72367 V
Valeur de CH1+ = 2.03406 V
Valeur de CH1+ = 2.11765 V
Valeur de CH1+ = 2.11846 V
Valeur de CH1+ = 2.11856 V
Valeur de CH1+ = 2.11896 V
Valeur de CH1+ = 2.11906 V
```



Par la suite ma carte réalisée cette année est arrivée et j'ai continué les tests sur celle-ci

J'ai donc reproduit le même test sur ma carte mais plusieurs problèmes sont apparues

Le premier problème était un problème d'alimentation et de détection de l'i2c par la RaspBerry pi qui fut résolu après avoir dessoudé et resoudé quelques composants

```
root@raspberrypi:/home/pi# i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  68  --  --  --  --  --  6e  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Après cela j'ai pu commencer les tests avec un potentiomètre câblé sur le bornier mais contrairement au premier test a cause de la présence de résistance de tirage , il ne fallait donc pas connecter le potentiomètre à l'alimentation

Mais un problème de changement brusque de valeur(0,5v à 3,3v) étant apparu par rapport au test sur la petite carte j'ai donc enlevé les résistances de tirage et rebranché le potentiomètre a l'alimentation pour se rapprocher au mieux du test réalisé sur la petite carte et essayer de voir la source du problème.

Le problème étant toujours présent il pouvait venir du programme et après des recherches dans la documentation du convertisseur que la valeur de la résolution ne dépendait pas de l'alimentation d'entrée mais d'une valeur de référence 2,048v

EQUATION 4-3:

$$LSB = \frac{2 \times V_{REF}}{2^N} = \frac{2 \times 2.048V}{2^N}$$

Where:

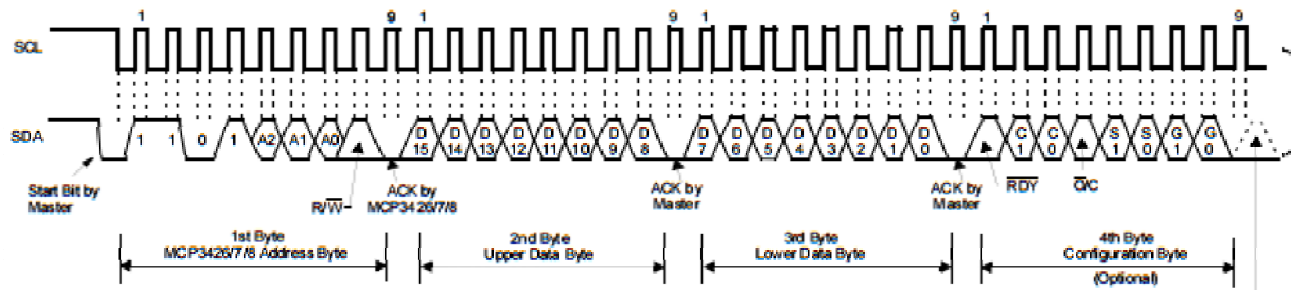
N = Resolution, which is programmed in the Configuration Register: 12, 14, or 16.

```
mesure = (256*i2cIn[0] + i2cIn[1])*2.048/32767;
```



Après ce changement sur le programme l'évolution était plus linéaire mais une fois arrivé à une valeur intermédiaire ,elle n'était pas fixe cela se voyait par le résultat du programme mais également par le multimètre ce qui permet de voir que le problème n'était pas logiciel.

J' ai donc voulu regarder de plus près grâce à un analyseur logique selon la documentation une trame est compose de :



- Octets d'adressage
- Les deux octets de la valeur convertie
- L'octet de configuration qui normalement en mode continue est sensé changer la valeur du bit 7 et donc changer de valeur si la conversion est finie et prête à en commencer une nouvelle.

bit 7

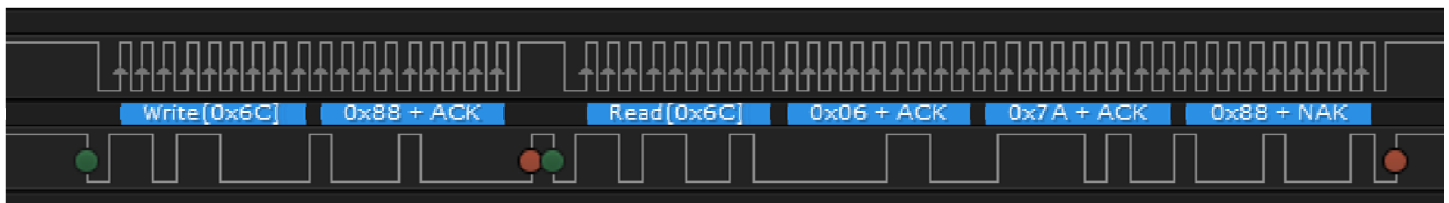
RDY: Ready Bit

This bit is the data ready flag. In read mode, this bit indicates if the output register has been updated with a latest conversion result. In One-Shot Conversion mode, writing this bit to "1" initiates a new conversion.

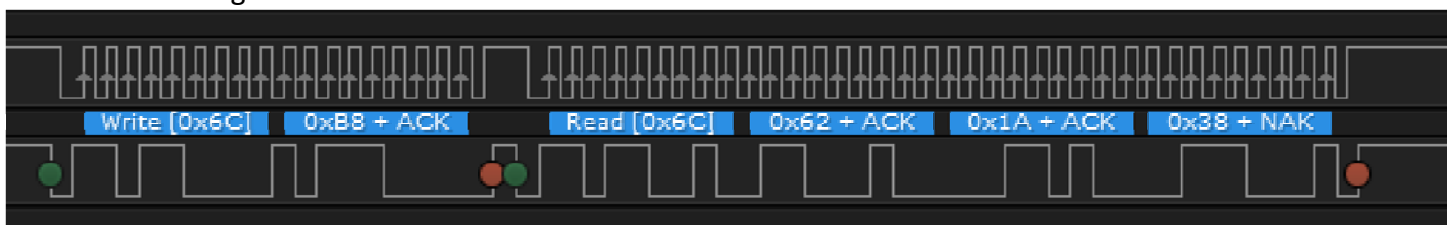
Reading RDY bit with the read command:

1 = Output register has not been updated.

0 = Output register has been updated with the latest conversion result.



Sur cette trame il y a d'abord l'envoi de la configuration à l'adresse du convertisseur et la lecture a cette adresse la valeur ainsi que le bit de configuration inchangé ce qui veut dire que la conversion n'était pas finie et donc le convertisseur n'était pas prêt à recommencer une conversion j'ai donc rajouté dans le programme un léger délai pour laisse le temps au convertisseur de finir sa conversion et donc ne pas mélanger les trames ce qui pouvait créer ce changement de valeur



Après ce changement de délai le dernier bit de configuration est bien changé comme écrit sur la documentation 0x88 devient 0x38

Le problème étant toujours présent mais le changement étant faible j'ai donc réalisé sur le programme une moyenne faite sur 20 résultats et ne prenant que les conversions finies avec ce bit changé



```

    i2cOut[0]=0x98;
    int var1=0;
    for (int i=0; i<20;i++)
    {
        bcm2835_i2c_write(i2cOut, 1); // Configuration en 16bits et mesure sur CH1+
        bcm2835_i2c_read(i2cIn, 3);
        if (i2cIn[2]<128)
        {
            mesure = (256*i2cIn[0] + i2cIn[1])*2.048/32767;
            resultat+=mesure;
            var1++;
        }
        delay(50);
    }
    cout<<"var1= "<< var1<<endl;
    resultat/=var1;
    cout<<"Valeur de CH1+ = "<< resultat<<" V"<<endl;

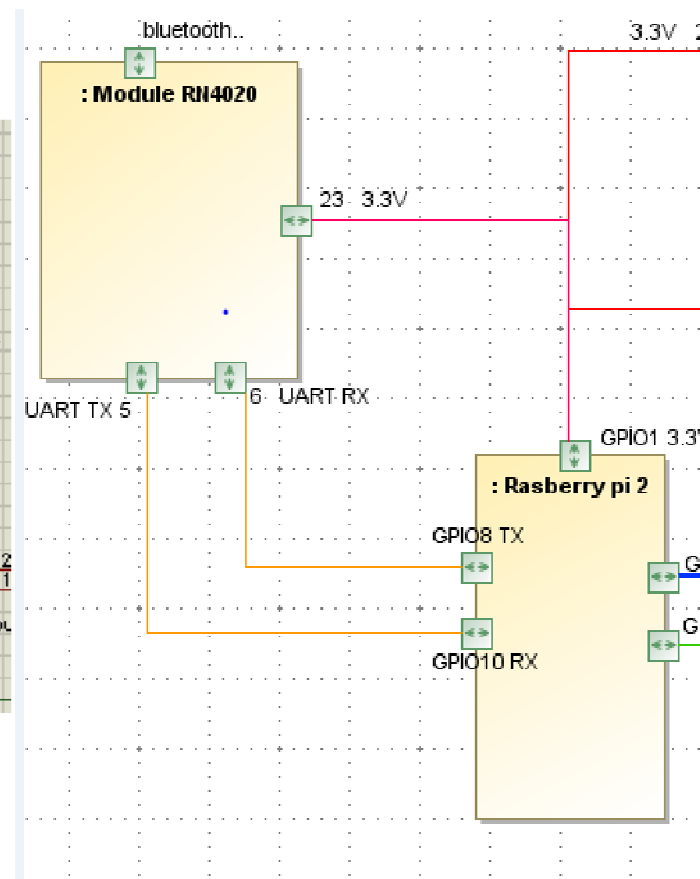
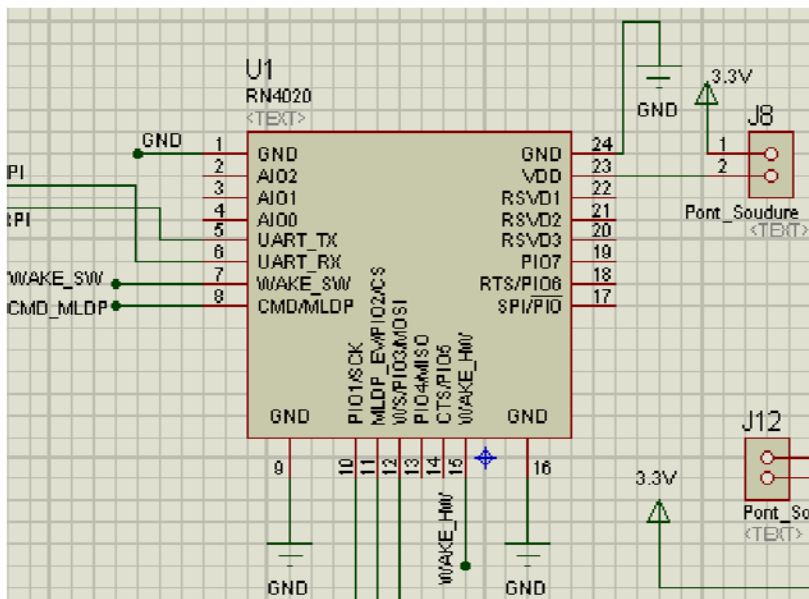
```

La Version complète comprenant les deux entrées , la moyenne sur 20 valeur ainsi que la validation de la transmission est disponible en annexes avec l'organigramme correspondant (p33)

Le problème persistant cela aurait pu venir de parasites dû au matériel utilisé fils non blindés mais après la mise d'un condensateur sur le potentiomètre voyant que rien n'avait changé cela pourrait venir de l'alimentation la RaspBerry pi alimentant déjà l'horloge temps réel et le composant lui même en plus du potentiomètre la RaspBerry pi ne permet peut être donc pas de réaliser ce test de cette façon .

4) Le module Bluetooth Low Energy

RN4020 :



Le module Bluetooth sera sur le coffret maitre et périphérique et permettra donc une transmission sans fil des informations du niveau des cuves converti en format numérique ainsi que l'heure à laquelle la mesure a été effectuée des cartes périphérique à la carte maitre pour ainsi pouvoir réunir les informations dans une base de données.


















Les avantages de ce composant sont sa taille ainsi que comme son nom l'indique sa faible consommation

Le RN4020 aurait fonctionné en UART entre lui et la RaspBerry pi mais je n'ai malheureusement pas pu mener a bien le début des essais sur ce composant .



III) Conclusion sur le travail réalisé :

1) Diagramme réel :

1			Projet Finale	180 h	Lun 09/01/17	Lun 12/06/17
2				71 h	Lun 09/01/17	Mar 14/03/17
3			Planification du projet (création du Gant+drive)	11 h	Lun 09/01/17	Jeu 12/01/17
4			Recherches sur le DS3231	7 h	Lun 16/01/17	Mar 17/01/17
5			essai de la mise en œuvre sur l'horloge temps réel 2016	10 h	Jeu 19/01/17	Mar 24/01/17
6			Création nouvelle mise en œuvre sur l'horloge temps réel pour la bonne version de raspbian	15 h	Jeu 26/01/17	Lun 06/02/17
7			Recherche sur le convertisseur Analogique numérique	11 h	Lun 06/02/17	Mar 28/02/17
8			création petite carte pour test CAN +test	18 h	Mar 28/02/17	Mar 14/03/17
9			Pour revue 2	63 h	Mar 14/03/17	Mar 09/05/17
10			Evolution schéma de la carte réalisée en 2016 puis routage et soudure	63 h	Mar 14/03/17	Mar 09/05/17
11			Pour revue final	46 h	Jeu 11/05/17	Lun 12/06/17
12			Test fonctionnement sur la carte soudée RTC+CAN	6 h	Jeu 11/05/17	Lun 15/05/17
13			Test + résolution problème du Convertisseur	24 h	Mar 16/05/17	Jeu 01/06/17
14						
15			Rédaction des dossiers	15 h	Jeu 01/06/17	Lun 12/06/17



2) Conclusion et connaissance acquise :

Le projet était très intéressant mais a cause de nombreux problèmes et un manque d'organisation il n'a pas pu être mené à bien. Mais j'ai tout de même appris beaucoup de choses en soudure ainsi qu'en C++ lors de la réalisation du programme du convertisseur.

J'ai également appris ce qu'était une horloge temps réel ainsi que le fonctionnement d'un convertisseur analogique numérique.

Le projet était la suite de celui de l'année 2016 mais malgré cela les problèmes sur le convertisseur avec la création de la carte pour accéder aux broches et les problèmes survenus pendant les tests nous ont empêchés de mener à bien ce projet.

Mais j'ai tout de même réussi à prendre l'heure de l'horloge depuis la Raspberry Pi, réalisé une nouvelle carte d'extension pour le projet comprenant tous les composants désirés et également compris le fonctionnement du convertisseur analogique numérique.



IV) Annexes Johann Crotte :

1) Nomenclature des composants :

Nomenclature

Carte 2sP							Client :		
Repère	Désignation du matériel	Valeur (Référence)	Tol.±%	Fabricant @ Fournisseur	Boîtier câblage	Qté	ref rapide "RS" (FA)	Prix U HT	Prix T HT
R1-R4	Résistance <u>cms</u>	330 Ohms	1%	<u>Vishay</u>	1206	4	<u>RS:679-2049</u>	0,014	0,056
R5,R6	Résistance <u>cms</u>	470 Ohms	1%	<u>Vishay</u>	603	2	<u>RS:679-0419</u>	0,019	0,038
R7,R8	Résistance <u>cms</u>	1K Ohms	1%	<u>Vishay</u>	603	2	<u>RS:678-9875</u>	0,020	0,040
C1	Condensateur Tantale	4,7µF	10%	<u>AVX</u>	B	1	<u>RS:699-2911</u>	0,137	0,137
C2-C6	Condensateur céramique multicouche	100nF	20%	<u>Phycomp</u>	805	5	<u>RS:378-542</u>		0,000 0,000
U1	Module BLE(blueetooth low energy)	<u>RN4020</u>		<u>Microship</u>		1	<u>RS:828-2856</u>	9,400	9,400
U2	Horloge temps réel (RTC)	<u>DS3231MZ+</u>		<u>Maxim</u>	SOIC8	1	<u>RS:778-1484</u>	6,120	6,120
U3	Convertisseur analogique numérique	<u>MCP3427</u>		<u>Microship</u>	MSOP10	1	<u>RS:703-7949</u>	2,730	2,730
U4	Mémoire <u>HAT</u>	<u>AT24C32A</u>		<u>ON Semiconductor</u>	SOIC8	1	<u>RS:781-4828</u>	0,335	0,335
D1,D3	LED VERTE			<u>Lite-on</u>	805	2	<u>RS:692-0922</u>	0,367	0,734
D2	LED ROUGE			<u>OSRAM</u>	805	1	<u>RS:497-4804</u>	0,218	0,218
D4	LED JAUNE			<u>BROADCAM</u>	805	1	<u>RS:486-0519</u>	0,227	0,227
							TOTAL HT FEUILLE	20,035	
							TOTAL HT GLOBAL	20,035	
							COEFFICIENT	0,000	

	Nom	Date	Folio
Crée :	2sp	27/03/2017	1/3
Mis à jour :	???	??/??/2012	

Carte 2sP							Client :		
Repère	Désignation du matériel	Valeur (Référence)	Tol.±%	Fabricant @ Fournisseur	Boîtier câblage	Qté	ref rapide "RS" (FA)	Prix U HT	Prix T HT
BAT1	PILE	3v		<u>PANASONIC</u>		1	<u>RS:513-2871</u>	2,010	2,010
FU1	<u>Polyswitch fusible</u>	0.5A - 0.25A		<u>TE connectivity</u>		1	<u>RS:517-8607</u>	0,401	0,401
J1	Connecteur raspberry pi			<u>ASSMANN WSW</u>		1	<u>RS:674-2365</u>	0,918	0,918
J7	Bornier	63V 6A		<u>Phoenix Contact</u>		1	<u>RS:220-4276</u> <u>FA: 3041360</u>	1,104 1,10	1,104
J15	Cavalier					1	<u>RS:674-2397</u>	0,052	0,052



2) Mise en œuvre DS3231MZ+ sur RaspBerry Pi2 (jessie et jessielight) en I2C

:

Réalise dans le cadre du projet 2SP par Crotte Johann

source :

<http://hardware-libre.fr/2013/08/RaspBerry-pi-ajouter-une-horloge-rtc-en-i%C2%B2c/>

<https://www.RaspBerrypi.org/forums/viewtopic.php?f=66&t=125003>

<http://innovelectronique.fr/2013/03/02/utilisation-du-bus-i2c-sur-RaspBerrypi/>

Documentation technique DS3231M

inspiré de la mise en œuvre du DS3231MZ+ de 2016

1)Activation de l'i2c sur la carte RaspBerry pi

-Utilisez votre RPi en mode Super User :

`pi@RaspBerrypi:~$ sudo su`

-Editez le fichier `/etc/module` avec l'éditeur nano

`root@RaspBerrypi:/home/pi# nano /etc/module`

-Ajoutez les lignes :

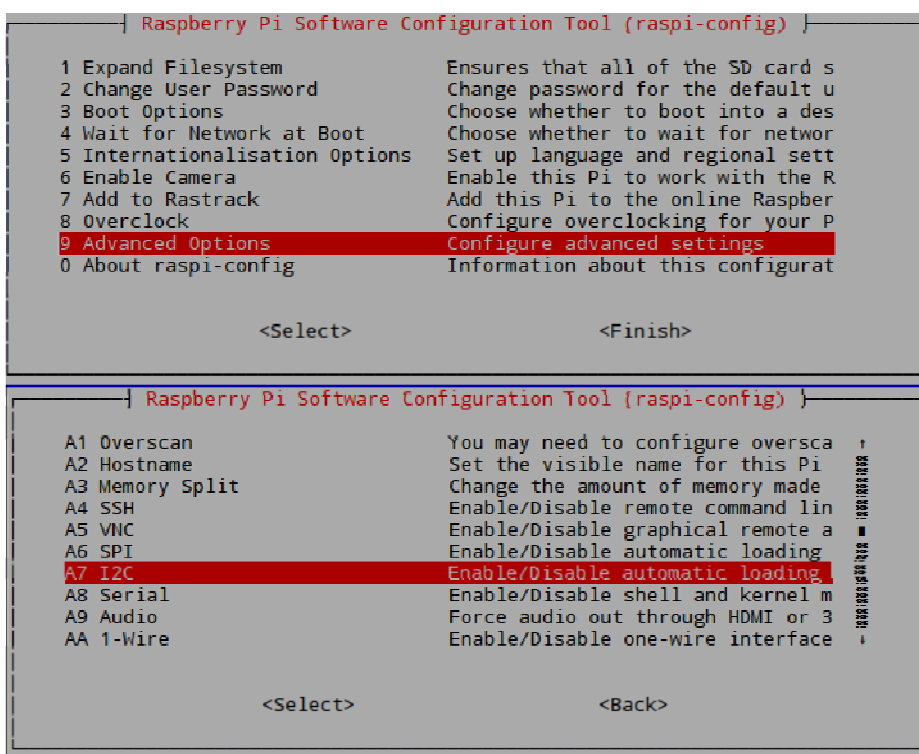
`i2c-bcm2708`

`i2c-dev`

-Sauvegardez puis lancer l'interface de configuration

`root@RaspBerrypi:/home/pi# raspi-config`

-Entrez dans « Advanced options » et validez l'i2c



-Quitter puis
redemarrez la
carte

`root@RaspBerrypi:`



/home/pi# reboot

-Pour Vérifier le bon fonctionnement de l'i2c sur votre carte :

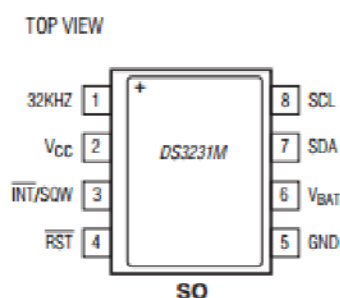
```
root@RaspBerrypi:/home/pi# dmesg|grep i2c
root@raspberrypi:/home/pi# dmesg|grep i2c
[ 3.158266] i2c /dev entries driver
```

```
root@RaspBerrypi:/home/pi# ls /dev/i2c*
root@raspberrypi:/home/pi# ls /dev/i2c*
/dev/i2c-1
```

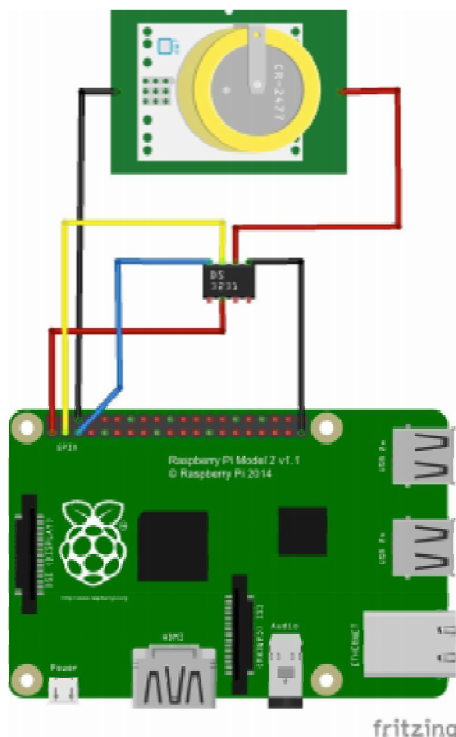
Le Bus i²c fonctionne correctement comme le montrent ces commandes

II)Montage

Disposition des broches de la RTC : DS3231M



Pour avoir une idée de ce à quoi doit ressembler le montage utilisant la RTC, voici le schéma Fritzing qui vous aidera à câbler :



Il est important de noter que sur le logiciel Fritzing la pile seule n'existe pas et est indépendante de la carte sur laquelle elle se trouve en haut du schéma. Sur votre montage, pluggez donc simplement la pile sur une breadboard.

En ce qui concerne le DS3231, ne disposant de ce composant qu'en version CMS, un typon de test a été créé afin de l'exploiter. À vous de voir si vous faites de même ou si vous utilisez le composant dans sa version non-CMS.

Broches utilisées:1,3,5,6,25 du GPIO

III)Mise en oeuvre du DS3231MZ+ :

une fois le montage réalisé vérifier que le circuit est bien détecté

```
root@RaspBerrypi:/home/pi# apt-get install i2c-tools
```

Le résultat obtenu est les installations faites sur votre Rpi.




```
root@RaspBerrypi:/home/pi# i2cdetect -y 1
```

Permet de detecter le DS3231MZ+ et que Le Bus i2c est exploitable

```
pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  48  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

-Le caractère ASCII 68 correspond à l'adresse du composant, c'est donc bien l'horloge temps réel DS3231MZ+ qui est détectée.

On trouve dans la documentation de la RTC l'adresse 0b1101000 ce qui correspond en hexadécimal au caractère 0x68 que nous obtenons.

-Afin de vérifier maintenant le bon fonctionnement du DS3231MZ+, suivez les instructions suivantes qui vont interroger le capteur de température interne de la RTC qui permet de retourner sa propre température car le DS3231MZ+ est aussi un capteur de temperature. Le résultat de la conversion est accessible avec l'adresse 0x11 sur deux octets, on a donc besoin de renseigner dans la commande le numéro du BUS (ici 1), le numéro et l'esclave (ici 0x68) et le registre à lire ou écrire (ici w pour écrire) :

```
root@RaspBerrypi:/home/pi# i2cget -y 1 0x68 0x11 w ,
```

Le résultat de cette commande est donc une valeur hexadécimale composée de deux octets. Vous pouvez essayer de faire évoluer la température du composant et relancer la commande pour vérifier que le résultat diffère .

Initialisation de la RTC

Pour cela, il faut écrire la valeur 0x04 dans le registre « control » qui est le 0x0E ainsi que la valeur 0x08 dans le registre « status » qui est le 0x0F :

```
root@RaspBerrypi:/home/pi# i2cset -y 1 0x68 0x0E 0x04 b
```

```
root@RaspBerrypi:/home/pi# i2cget -y 1 0x68 0x0F b
```

0x88 (resultat de la commande)

```
root@RaspBerrypi:/home/pi# i2cset -y 1 0x68 0x0F 0x08 b
```

```
root@RaspBerrypi:/home/pi# i2cget -y 1 0x68 0x0F b
```

0x08

À présent vous pouvez lire les secondes, minutes, heures, jour, mois et année en questionnant le bon registre, registres allant de 0x01 à 0x06 et correspondant à chacune des valeurs ci-dessus dans l'ordre. Par exemple la commande...

```
root@RaspBerrypi:/home/pi# i2cget -y 1 0x68 0x06 b
```

va permettre d'obtenir la valeur hexadécimale correspondant à l'année.

Vous allez maintenant pouvoir installer l'horloge temps réel à la RaspBerry pour que la RaspBerry démarre avec l'heure de celle ci et non celle d'origine et sans internet pour cela brancher la RaspBerry a internet pour qu'elle soit a l'heure puis taper

```
root@RaspBerrypi:/home/pi# echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```



(déclaration nouvelle esclave i2c)

(Notez que dans la commande figure la référence du composant DS1307 qui est une autre horloge temps réel mais qui est compatible avec notre DS3231MZ+)

```
root@RaspBerrypi:/home/pi# hwclock -r (lecture de l'heure sur la RTC)
root@RaspBerrypi:/home/pi# date (lecture de l'heure sur la RaspBerry)
root@RaspBerrypi:/home/pi# hwclock -w (écriture de l'heure de la RaspBerry sur la RTC)
root@RaspBerrypi:/home/pi# hwclock -r (lecture de l'heure sur la RTC pour vérifier le bon
changement )
```

```
root@raspberrypi:/home/pi# hwclock -r
mar. 24 janv. 2017 15:06:56 CET -0.308749 seconds
```

La RTC est désormais à l'heure et les restera même si elle est déconnectée de la RaspBerry pi ou si la RaspBerry pi n'est plus alimentée grâce à la pile

Maintenant nous allons affecter cette RTC au démarrage de la RaspBerry pour que celle-ci dispose de l'heure y compris déconnectée du réseau et d'internet
 Pour cela il faut changer certains fichiers de la RaspBerry

-Éditez le fichier **/etc/modules** avec l'éditeur nano

```
root@RaspBerrypi:/home/pi# nano /etc/modules
```

ajoutez la ligne

```
rtc-ds1307
```

Sauvegardez et quittez

-Éditez le fichier **/etc/rc.local** avec l'éditeur nano

```
root@RaspBerrypi:/home/pi# nano /etc/rc.local
```

avant la ligne

```
exit 0
```

ajoutez les lignes :

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

```
sudo hwclock -s
```

Sauvegardez et quittez

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s
```

Les commandes faites jusqu'à présent suffisent pour la version jessie light de RaspBerry

Pour la version jessie basique il faut en plus rajouter ces quelques commandes

Donc pour finir dans le fichier **/lib/udev/hwclock-set**

```
root@RaspBerrypi:/home/pi# nano /lib/udev/hwclock-set
```

```
if [ -e /run/systemd/system ] ; then
```

```
exit 0
```

```
fi
```

mettre en commentaire

```
#if [ -e /run/systemd/system ] ; then
```

```
#exit 0
```

```
#fi
```

Sauvegardez et quittez

```
#if [ -e /run/systemd/system ] ; then
#   exit 0
#fi
```

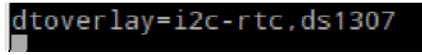


et écrire dans le fichier **/boot/config.txt**

root@RaspBerrypi:/home/pi# nano /boot/config.txt

rajouter la ligne

dtoverlay=i2c-rtc,ds-1307



Sauvegardez et quittez

Vous pouvez maintenant essayer :

Éteignez votre RaspBerry attendez un moment et redémarrez la et toujours connecté a la RTC mais déconnecter du réseau

et la RaspBerry sera a la bonne heure et non à l'heure d'où elle s'ai éteinte comme auparavant.

Fin



3) Fiches de lecture croisée sur la mise en œuvre de l'horloge temps réel DS3231MZ :

1. Fiche de rapport de lecture croisée.		Réf. <u>FRLC-XX</u>
<i>Projet</i>	<i>Auteur</i> Johann crotte	<i>Date</i> XX/XX/2017
<i>Produit</i> Horloge en temps réelle	<i>Durée de la réunion</i> X	
<i>Lecteur(s)</i> Doux Christopher	<i>Contrôleur des corrections</i> X	
<i>Liste des erreurs détectées</i>		
<p>Une mise a jour a était effectuer donc présence de quelques variation dans les interfaces graphiques.</p> <p>Mais aucune présence d'erreur majeure qui pourrait entraver le fonctionnement des composants.</p> <p>Si-joint le document mis a jour pour une version plus récente de la <u>raspberry pi</u>.</p>		

1. Configuration <u>RTC</u>.		Réf. <u>FRLC-XX</u>
<i>Projet</i> AIES	<i>Auteur</i> Crotte j,	<i>Date</i> 30/01/2017
<i>Produit</i> <u>RTC</u>	<i>Durée de la réunion</i> X	
<i>Lecteur(s)</i> <u>Lavigne N.</u>	<i>Contrôleur des corrections</i> X	
<i>Liste des erreurs détectées</i>		
<p>Aucune erreurs détectées, tous fonctionne parfaitement, l'horloge est à l'heure même en l'enlevant du réseau et en redémarrant la <u>Raspberry PI</u></p>		

4)



Programme convertisseur analogique numérique :

(pour 2 entrées avec moyenne sur 20 valeurs)

```
void init_BCM2835()
{
    unsigned int slave_address=0x6e;
    if (!bcm2835_init())
    {
        printf("bcm2835_init failed. Are you running as root??\n");
    }
    if (!bcm2835_i2c_begin())
    {
        printf("bcm2835_i2c_begin failed. Are you running as root??\n");
    }
    bcm2835_i2c_setSlaveAddress(slave_address);
    bcm2835_i2c_setClockDivider(clk_div);
}

double getMesure()
{
    int temp=0;
    double mesure = 0;

    char i2cOut[2];
    char i2cIn[3];

    i2cOut[0]=0x88;
    bcm2835_i2c_write(i2cOut, 1); // Configuration en 16bits et mesure sur CH1+
    bcm2835_i2c_read(i2cIn, 3);
    mesure = (256*i2cIn[0] + i2cIn[1])*3.3/32767;

    return mesure;

    i2cOut[0]=0x98;
    int var1=0;
    for (int i=0; i<20;i++)
    {
        bcm2835_i2c_write(i2cOut, 1); // Configuration en 16bits et mesure sur CH1+
        bcm2835_i2c_read(i2cIn, 3);
        if (i2cIn[2]<128)
        {
            mesure = (256*i2cIn[0] + i2cIn[1])*2.048/32767;
            resultat+=mesure;
            var1++;
        }
        delay(50);
    }
    cout<<"var1= "<< var1<<endl;
    resultat/=var1;
    cout<<"Valeur de CH1+ = "<< resultat<<" V"<<endl;

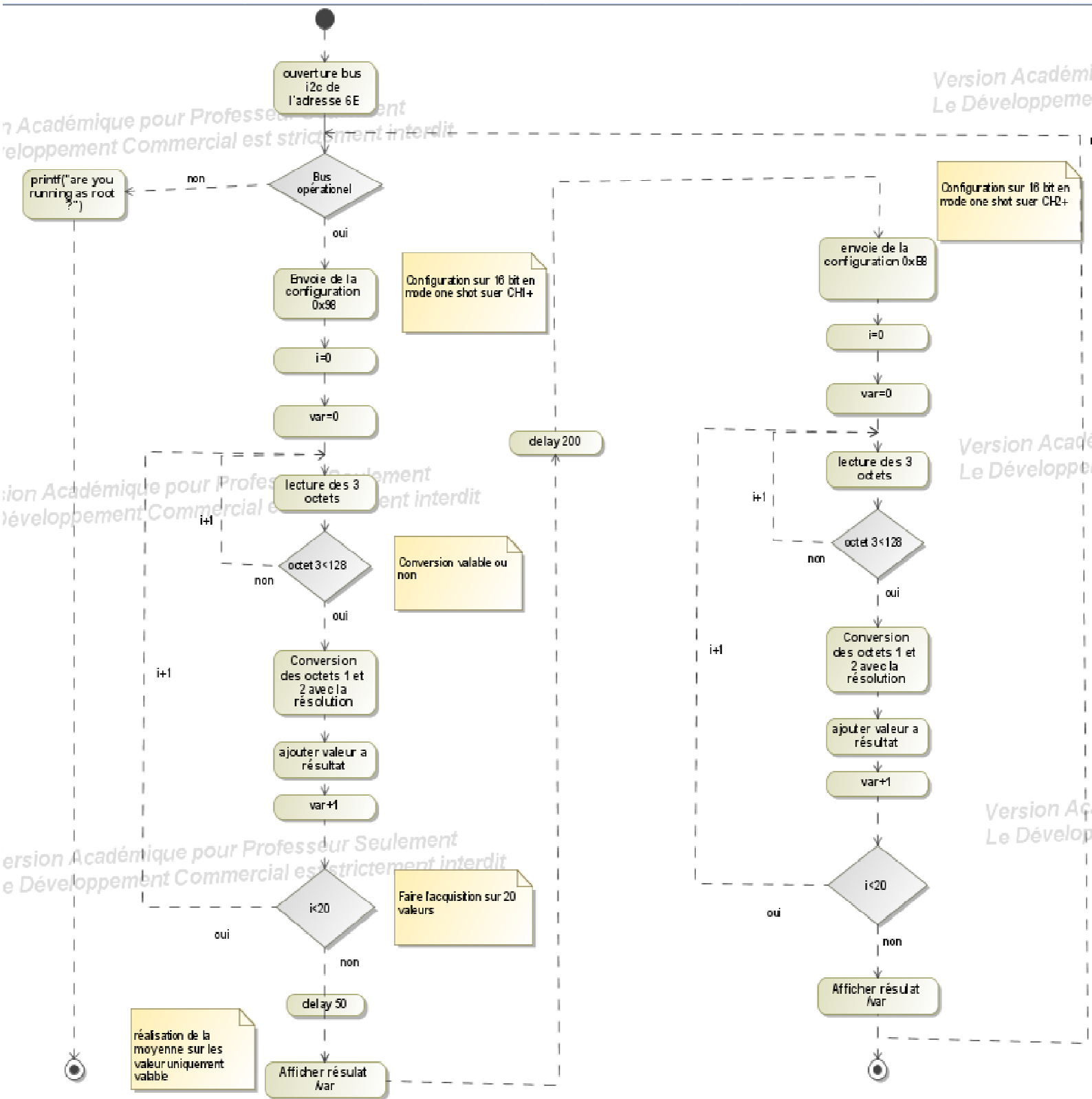
    delay(200);
    resultat=0;
    i2cOut[0]=0xB8;
    int var2=0;
    for (int i=0; i<20; i++)
    {
        bcm2835_i2c_write(i2cOut, 1); // Configuration en 16bits et mesure sur CH2+
        bcm2835_i2c_read(i2cIn, 3);
        if (i2cIn[2]<128)
        {
            mesure = (256*i2cIn[0] + i2cIn[1])*2.048/32767;
            resultat+=mesure;
            var2++;
        }
        delay(50);
    }
    cout<<"var2 = "<< var2<<endl;
    resultat/=var2;
    cout<<"Valeur de CH2+ = "<< resultat<<" V"<<endl;

    return 0;
}

int main(void)
{
    init_BCM2835();
    while(1)
    {
        getMesure();
    }
    bcm2835_i2c_end();
    return 0;
}
```



5) Organigramme correspondant au programme :



6) Journal de bord :

Lundi 9 janvier 2017 (3h)

- prise de connaissance des docs
- création drive en équipe

mardi 10 janvier 2017 (4h)

- prise de connaissance projet 2sp 2017 (2h)
- lien RaspBerry composant (2h)

jeudi 12 janvier 2017 (3h)

- recherche doc rtc (1h)
- mise en œuvre de l'horloge temps réel depuis le fichier des élève (install i2c)2016(1,5h)
erreur trouvé `ls /dev/i2c-` correction `ls /dev/i2c*` (trouvé sur inwelectrique.fr)
- recherche Can mcp3427 raspberry (fichier drive)(0,5h)

lundi 16 janvier 2017 (3h)

- suite mise en œuvre avec la carte année 2016(2h)
- `pb(echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device)` (un pas l)
- ne récupère pas l'heure car manque
`echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device`
- `sudo hwclock -s`
- pour l'affecter au démarrage (1h)

mardi 17 janvier 2017 (4h)

- les deux lignes ne fonctionnent que pour le prochain redémarrage +correction (1h)
- prise en main trello (planification en groupe)(le reste)
- recherche (le reste du reste)

jeudi 19 janvier 2017 (3h)

- <http://hardware-libre.fr/2013/08/RaspBerry-pi-ajouter-une-horloge-rtc-en-i%C2%B2c/site> trouvé ligne manquante
- dans `/etc/modules` `rtc-ds1307` (1h),
- problème suivant « module non reconnue ») « UU et pas 68 » i2c plus en fonctionnement(1h)
- bilan -rtc a l'heure -i2c en fonction mais RaspBerry ne prend pas au démarrage(1h)

lundi 23 janvier 2017(3h)

- problème trouvé --->version raspbian**
- rajouter ligne manquante pour le redémarrage (1h)
- `«#if [-e /run/systemd/system] ; then`
- `# exit 0`
- `#fi » dans /lib/udev/hwclock-set (mise en commentaire)`
- et « `dtoverlay=i2c-rtc,ds-1307` » dans `/boot/config.txt`
- source=<https://www.RaspBerrypi.org/forums/viewtopic.php?f=66&t=125003>
- rédaction feuille de mise en œuvre pour lecture croise sur le ds3231 (2h)

mardi 24 janvier 2017(4h)



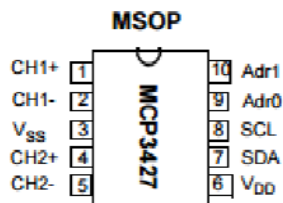
- redaction feuille de mise en œuvre pour lecture croise sur le ds3231MZ (3h)
- briefing avec mr defrance (1h)

jeudi 26 janvier 2017(3h)

Absent

lundi 30 janvier 2017(3h)

- finissions fiche de mise en œuvre sur le ds3231MZ(1h)
- recherche pour mcp3427-E/UN petit circuit imprimé
<https://upverter.com/upn/6f62033b2a535dce/>
https://easyeda.com/component/MCP3427_EUN-8TbCJXNzC
<https://www.snapeda.com/parts/MCP3427-E/MF/Microchip/view-part/> (2h)



mardi 31 janvier 2017(4h)

proteus petite carte pour mesure sur CAN mcp3427

jeudi 02 février 2017(3h)

proteus petite carte pour mesure sur CAN mcp3427
prise d'un condensateur plus grand 47µF pour faire un pont entre VDD et VSS

lundi 06 février 2017(3h)

finissions+gravure carte

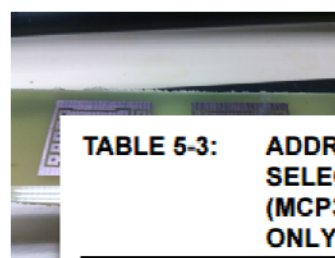
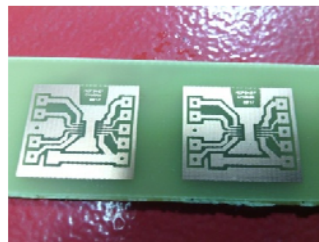
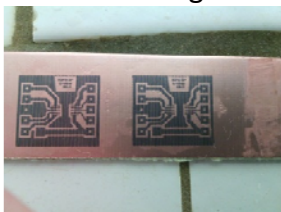


TABLE 5-3: ADDRESS BITS VS. ADDRESS SELECTION PINS FOR (MCP3427 AND MCP3428 ONLY) (NOTE 1, 2, 3)

I ² C Device Address Bits			Logic Status of Address Selection Pins	
A2	A1	A0	Adr0 Pin	Adr1 Pin
0	0	0	0 (Addr_Low)	0 (Addr_Low)
0	0	1	0 (Addr_Low)	Float
0	1	0	0 (Addr_Low)	1 (Addr_High)
1	0	0	1 (Addr_High)	0 (Addr_Low)
1	0	1	1 (Addr_High)	Float
1	1	0	1 (Addr_High)	1 (Addr_High)
0	1	1	Float	0 (Addr_Low)
1	1	1	Float	1 (Addr_High)
0	0	0	Float	Float

Note 1: Float: (a) Leave pin without connecting to anything (left floating), or (b) apply Addr_Float voltage.

2: The user can tie the pins to V_{SS} or V_{DD}:
- Tie to V_{SS} for Addr_Low
- Tie to V_{DD} for Addr_High

3: See Addr_Low, Addr_High, and Addr_Float parameters in **Electrical Characteristics Table**.

mardi 07 février 2017(4h)

perçage +soudure sur la carte



lundi 27 février 2017 (3h)

mcp 3427

premier test du mcp 3427 non concluant

adressage mis a 6E mais détecté a 6D

mardi 28 février 2017(4h)

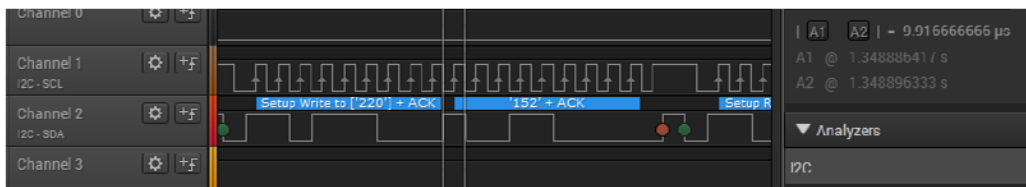
test convertisseur (1h)

varie en fonction du potentiomètre

```
Valeur de CH1+ = 1.53715 V
Valeur de CH1+ = 1.72367 V
Valeur de CH1+ = 2.03406 V
Valeur de CH1+ = 2.11765 V
Valeur de CH1+ = 2.11846 V
Valeur de CH1+ = 2.11856 V
Valeur de CH1+ = 2.11896 V
Valeur de CH1+ = 2.11906 V
```

```
#define clk_div BCM2835_I2C_CLOCK_DIVIDER_2500
```

vitesse défini a 100Kb/s



$1/10\mu s = 100K$
bits/s

-rédaction dossier revue 1 (3h)

jeudi 2 mars 2017(3h+3h physique)

-rédaction dossier

début convertisseur en 2 conversions

lundi 6 mars 2017 (4h)

diapo revue 1

mardi 7 mars 2017(4h)

revue 1 +verif remarque pt 1000,,

jeudi 9 mars 2017 (4h)

schéma carte

rtc--->Microelectromechanical system (MEMS) oscillators

lundi 13 mars 2017(3h)

Equipe projet 2SP



J3 monte ainsi que les traversé pour faire de la place en bas

mardi 14 mars 2017(4h)

mise du can et du bornier(+création empreinte)

jeudi 16 mars (3h)

HAT deux emplacements

lundi 20 mars (3h)

ajustements + mise en place cavalier et résistance de pull down

mardi 21 mars (4h)

finissions ares + code commande isis

jeudi 23 mars 2017 (3h)

perçage bornier(erreur dut a la création des broches)

commande de la carte délai 2 semaines

lundi 27 mars 2017 (3h)

fiche des composants

mardi 28 mars 2017 (4h)

fin fiche des composants

jeudi 30 mars 2017 (3h)

Dossier

lundi 3 avril 2017 (3h)

Dossier

Mardi 4 avril (4h)

Dossier

Lundi 24 avril(3h)

Réunion pour revue 2

arrivé de la carte

Mardi 25 avril (4h)

ordre soudure +soudure

jeudi 4 mai (4h)

http://www.framboise314.fr/un-module-rtc-a-base-de-ds1302-pour-le-RaspBerry-pi/#Modification_pour_assurer_lrsquoécriture_dans_le_DS1302

semaine 8 au 12 mai

épreuve bts

lundi 15 mai (4h)

test de la carte soudée

Reconfiguration de la RTC sur la carte

mardi 16 mai(4h)

Equipe projet 2SP



test mcp 3427 sur la carte
jeudi 18 mai(4h)
suite test du convertisseur



7) Compte rendu de réunion pour la revue 2 :

5. Compte-rendu d'activité.		
Date	Participants	Activité
24/04/2017	STIN, De Francesco, Crotte, Célié	<p>Rappel des objectifs de chacun pour la revue 2.</p> <p>Réunion pour la mise au point de la partie commune du dossier Revue 2. Suite à la demande du client (professeur).</p>
<p>Rappel des objectifs Revue 2 :</p> <p>Anton : Présenter un site web avec des graphiques par zone et un résumé des cuves par zone</p> <p>Vincent : Améliorer le fichier de traçabilité PDF.</p> <p>Cédric : Dynamiser le site Web avec PUG.</p> <p>Johann : Présenter la carte finie.</p> <p>Mise au point Partie commune revue 2 :</p> <ol style="list-style-type: none"> 1) Présentation de l'entreprise 2SP 2) Le Projet (but, cible <u>clientelle</u>, <u>interet</u> par rapport à l'existant) 3) Présentation du matériel (cuves, cartes, raspberry, Convertisseur, RTC, RN4020 ...) 4) Présentation des Software <ul style="list-style-type: none"> • Le site web (Onglet, utilisateurs) • La base de données • La fiche de traçabilité • Application web d'aide au placement 5) Les codes utilisés (leurs intérêts...) <p>Nous nous sommes mis d'accord pour ce plan de la partie commune du dossier pour la seconde revue.</p> <p>Anton</p> <p>Vincent</p> <p>Cédric</p> <p>Johann</p>		



8) Informations concernant les cuves cryogéniques :

Tableau de lecture des sondes

Sondes de niveau

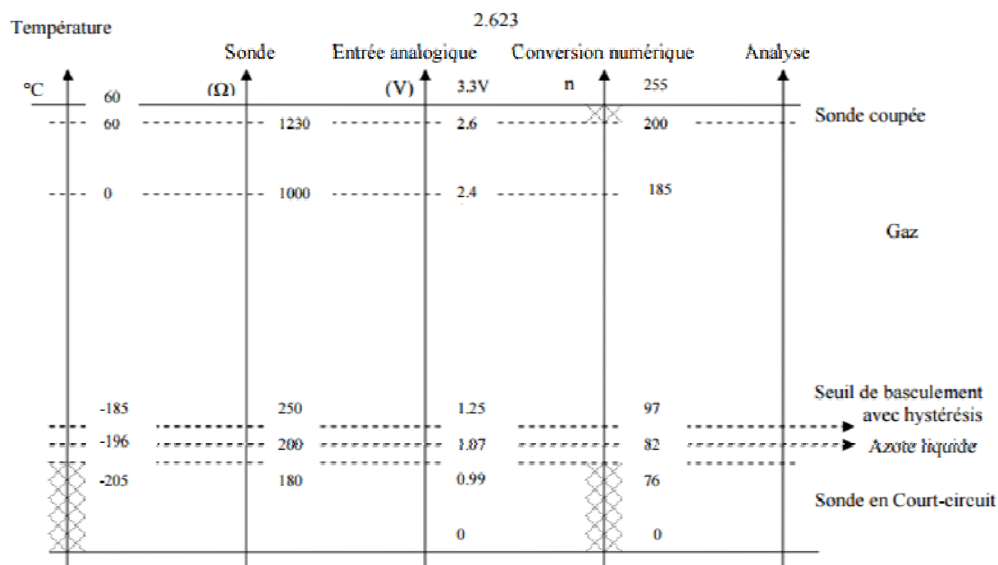


Table de Conversion sonde Platine PT1000

* (C)	R (Ω)	* (C)	R (Ω)	* (C)	R (Ω)	* (C)	R (Ω)	* (C)	R (Ω)	* (C)	R (Ω)	* (C)	R (Ω)
100	1388.8	50	1194	0	1000	-50	803.1	-100	602.5	-150	397.1		
99	1381.2	49	1190.1	-1	996.1	-51	799.1	-101	598.5	-151	393		
98	1377.4	48	1186.2	-2	992.2	-52	795.1	-102	594.4	-152	388.8		
97	1373.6	47	1182.4	-3	988.3	-53	791.1	-103	590.4	-153	384.6		
96	1369.8	46	1178.5	-4	984.4	-54	787.2	-104	586.3	-154	380.4		
95	1366	45	1174.7	-5	980.4	-55	783.2	-105	582.2	-155	376.3		
94	1362.2	44	1170.8	-6	976.5	-56	779.2	-106	578.2	-156	372.1		
93	1358.4	43	1167	-7	972.6	-57	775.2	-107	574.1	-157	367.9		
92	1354.6	42	1163.1	-8	968.7	-58	771.3	-108	570	-158	363.7		
91	1350.8	41	1159.3	-9	964.8	-59	767.3	-109	566	-159	359.5		
90	1347	40	1155.4	-10	960.9	-60	763.3	-110	561.9	-160	355.3		
89	1343.2	39	1151.5	-11	956.9	-61	759.3	-111	557.8	-161	351.1		
88	1339.4	38	1147.7	-12	953	-62	755.3	-112	553.8	-162	346.9		
87	1335.6	37	1143.8	-13	949.1	-63	751.3	-113	549.7	-163	342.7		
86	1331.8	36	1139.9	-14	945.2	-64	747.3	-114	545.6	-164	338.5		
85	1328	35	1136.1	-15	941.2	-65	743.3	-115	541.5	-165	334.3		
84	1324.2	34	1132.2	-16	937.3	-66	739.3	-116	537.4	-166	330.1		
83	1320.4	33	1128.3	-17	933.4	-67	735.3	-117	533.3	-167	325.9		
82	1316.6	32	1124.5	-18	929.5	-68	731.3	-118	529.2	-168	321.6		
81	1312.7	31	1120.6	-19	925.5	-69	727.3	-119	525.2	-169	317.4		
80	1308.9	30	1116.7	-20	921.6	-70	723.3	-120	521.1	-170	313.2		
79	1305.1	29	1112.8	-21	917.7	-71	719.3	-121	517	-171	309		
78	1301.3	28	1109	-22	913.7	-72	715.3	-122	512.9	-172	304.7		
77	1297.5	27	1105.1	-23	909.8	-73	711.3	-123	508.8	-173	300.5		
76	1293.7	26	1101.2	-24	905.9	-74	707.3	-124	504.7	-174	296.3		
75	1289.8	25	1097.3	-25	901.9	-75	703.3	-125	500.6	-175	292		
74	1286	24	1093.5	-26	898	-76	699.3	-126	496.4	-176	287.8		
73	1282.2	23	1089.6	-27	894	-77	695.3	-127	492.3	-177	283.5		
72	1278.4	22	1085.7	-28	890.1	-78	691.3	-128	488.2	-178	279.3		
71	1274.5	21	1081.8	-29	886.2	-79	687.3	-129	484.1	-179	275		
70	1270.7	20	1077.9	-30	882.2	-80	683.3	-130	480	-180	270.8		
69	1266.9	19	1074	-31	878.3	-81	679.2	-131	475.9	-181	266.5		
68	1263.1	18	1070.2	-32	874.3	-82	675.2	-132	471.8	-182	262.3		
67	1259.2	17	1066.3	-33	870.4	-83	671.2	-133	467.6	-183	258		
66	1255.4	16	1062.4	-34	866.4	-84	667.2	-134	463.5	-184	253.7		
65	1251.6	15	1058.5	-35	862.5	-85	663.1	-135	459.4	-185	249.4		
64	1247.7	14	1054.6	-36	858.5	-86	659.1	-136	455.2	-186	245.2		
63	1243.9	13	1050.7	-37	854.6	-87	655.1	-137	451.1	-187	240.9		
62	1240.1	12	1046.8	-38	850.6	-88	651.1	-138	447	-188	236.6		
61	1236.2	11	1042.9	-39	846.7	-89	647	-139	442.8	-189	232.3		
60	1232.4	10	1039	-40	842	-90	643	-140	438.7	-190	228		
59	1228.6	9	1035.1	-41	838.8	-91	639	-141	434.5	-191	223.7		
58	1224.7	8	1031.2	-42	834.8	-92	634.9	-142	430.4	-192	219.4		
57	1220.9	7	1027.3	-43	830.8	-93	630.9	-143	426.3	-193	215.1		
56	1217	6	1023.4	-44	826.9	-94	626.8	-144	422.1	-194	210.8		
55	1213.2	5	1019.5	-45	822.9	-95	622.8	-145	417.9	-195	206.5		
54	1209.3	4	1015.6	-46	818.9	-96	618.7	-146	413.8	-196	202.2		
53	1205.5	3	1011.7	-47	815	-97	614.7	-147	409.6	-197	197.9		
52	1201.6	2	1007.8	-48	811	-98	610.6	-148	405.5	-198	193.6		
51	1197.8	1	1003.9	-49	807	-99	606.6	-149	401.3	-199	189.3		
										-200	184.9		



Etudiant IR1 (DEFRANCESCO Anton) :

I) Présentation de mes tâches :

1) Objectifs :

a) Rappel sur l'architecture globale et les problématiques :

L'objectif du système est de superviser le niveau de cuves cryogéniques. Pour ce faire des microcontrôleurs **esclaves** associés à des sondes de niveau envoient le niveau des cuves en les convertissant en données numériques à un microcontrôleur **maître** via une liaison Bluetooth. Ce dernier transfère les données à une base de données. Le microcontrôleur maître possède un service web qui assure l'interface homme-machine permettant la supervision des cuves. Cette application web utilise la base de données pour la gestion et l'affichage des cuves entre autres.

La majeure partie de mon travail se base sur le projet 2sp de l'année 2015-2016 qui possédait déjà toutes les caractéristiques mentionnées ci-dessus, à la différence que ce projet ne pouvait pas superviser plusieurs cuves, mais une seule.

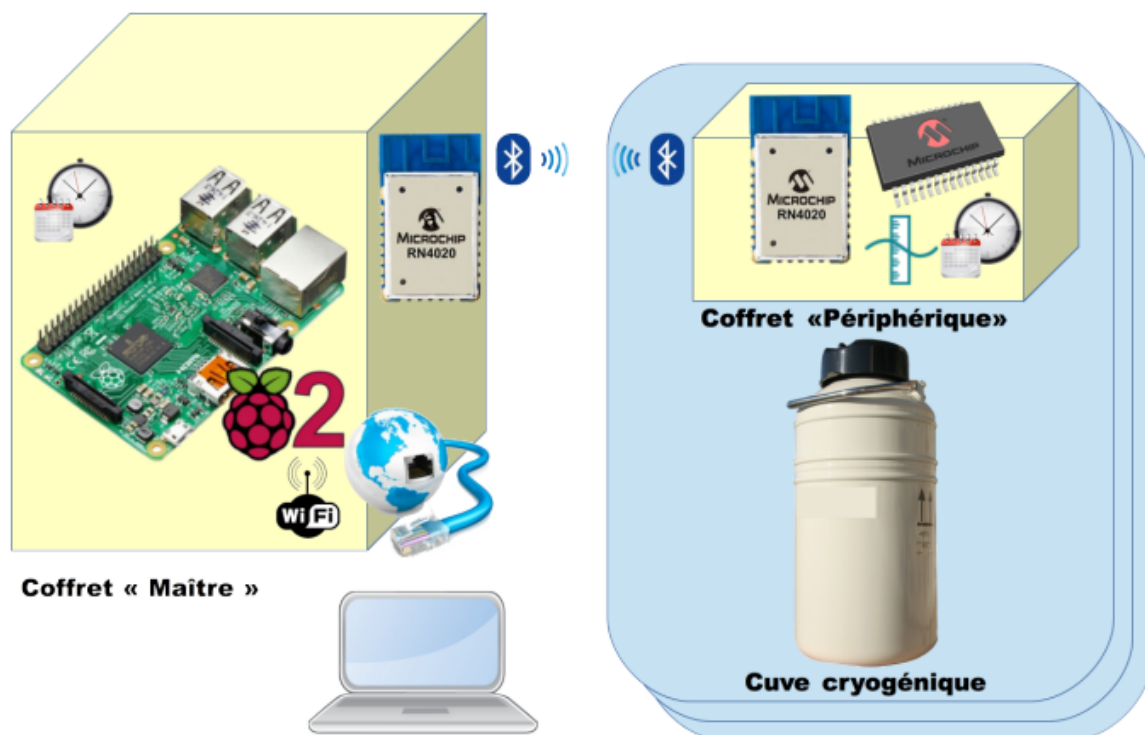


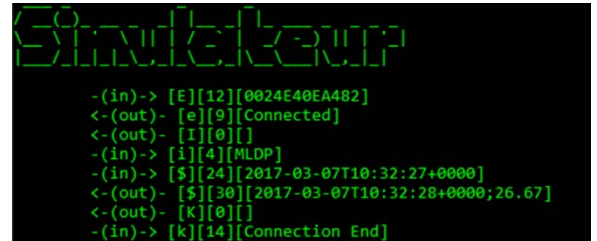
Figure 1 : Vue d'ensemble du système



La partie dont j'ai la charge se concentre sur la récupération des données venant du module Bluetooth (RN4020) ainsi que de leur intégration à la base de données ainsi que sur l'application web.

L'intégration des données à la base de données et la gestion du module RN4020 se réalise via un simulateur qui simule ces deux tâches. Ce simulateur est subdivisé en deux programmes C++, le **simulateur** et le **moniteur** qui communiquent via une file de messages Linux.

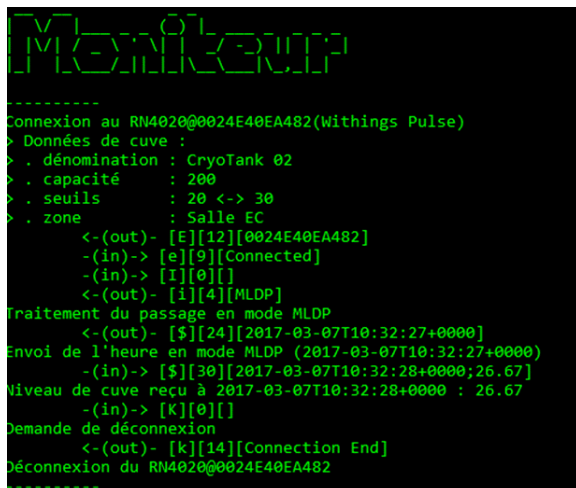
- Le **simulateur** a pour rôle de générer des niveaux de cuves, de créer la file de message ainsi que de créer des modules Bluetooth virtuels qui simulent l'utilisation d'un module réel.



```

Simulateur
-(in)-> [E][12][0024E40EA482]
<-(out)- [e][9][Connected]
<-(out)- [I][0][ ]
-(in)-> [i][4][MLDP]
-(in)-> [$][24][2017-03-07T10:32:27+0000]
<-(out)- [$][30][2017-03-07T10:32:28+0000;26.67]
<-(out)- [K][0][ ]
-(in)-> [k][14][Connection End]
  
```

Figure 3: Le simulateur de niveau



```

Moniteur
-----
Connexion au RN4020@0024E40EA482(Withings Pulse)
> Données de cuve :
> . dénomination : CryoTank 02
> . capacité : 200
> . seuils : 20 <-> 30
> . zone : Salle EC
<-(out)- [E][12][0024E40EA482]
-(in)-> [e][9][Connected]
-(in)-> [I][0][ ]
<-(out)- [i][4][MLDP]
Traitement du passage en mode MLDP
<-(out)- [$][24][2017-03-07T10:32:27+0000]
Envoi de l'heure en mode MLDP (2017-03-07T10:32:27+0000)
-(in)-> [$][30][2017-03-07T10:32:28+0000;26.67]
Niveau de cuve reçu à 2017-03-07T10:32:28+0000 : 26.67
-(in)-> [K][0][ ]
Demande de déconnexion
<-(out)- [k][14][Connection End]
Déconnexion du RN4020@0024E40EA482
-----
  
```

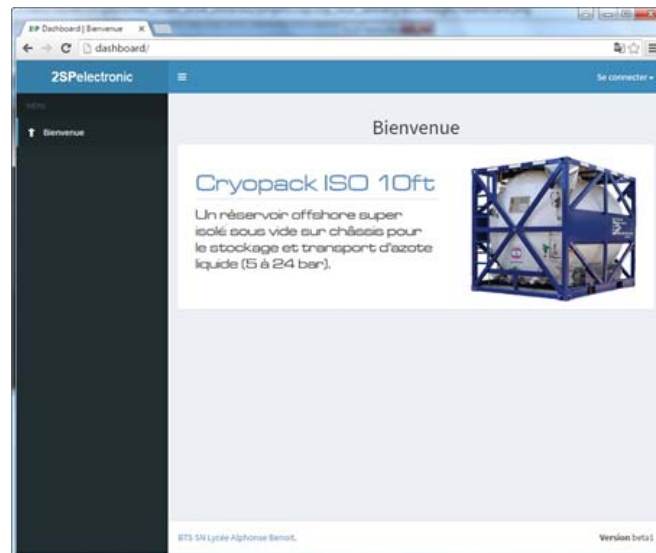
Figure 2: Le moniteur de niveau

- Le **moniteur** a lui pour rôle d'envoyer une trame de demande de niveau de cuve, ainsi que de recevoir le niveau d'azote liquide généré par le simulateur avant d'insérer cette donnée dans la base de données.

L'application web a plusieurs rôles :

- La gestion d'utilisateurs et d'administrateurs du système.
- L'ajout, la modification et la suppression de cuves cryogéniques.
- L'ajout, la modification et la suppression de zones de cuves cryogéniques.
- L'affichage des niveaux de cuves par zones en graphique et en jauge.





L'application web fonctionne sous une architecture PHP, reprenant le fonctionnement de modèle **Modèle-Vue-Contrôleur**. Le rendu de l'application, la **vue**, est généré par un moteur de Template, *Twig*, invoqué par PHP. Des **contrôleurs** sont responsables des traitements des requêtes de l'application web demandées par l'utilisateur et de l'invocation du rendu des pages via *Twig*. La partie **Modèle** est composée des DAO (Data Access Object), qui représentent les différentes tables de la base de données et sont responsables de l'interaction avec celle-ci.

Deux grandes **problématiques** se dégagent dans le cadre de ma mission sur le système.

- La première est d'adapter l'architecture de l'application web et de la base de données pour la gestion de plusieurs cuves par zone.
- La deuxième est d'injecter les données relatives au niveau des cuves dans la base de données via le module Bluetooth (RN4020).

Diagrammes UML/SYSML au niveau système

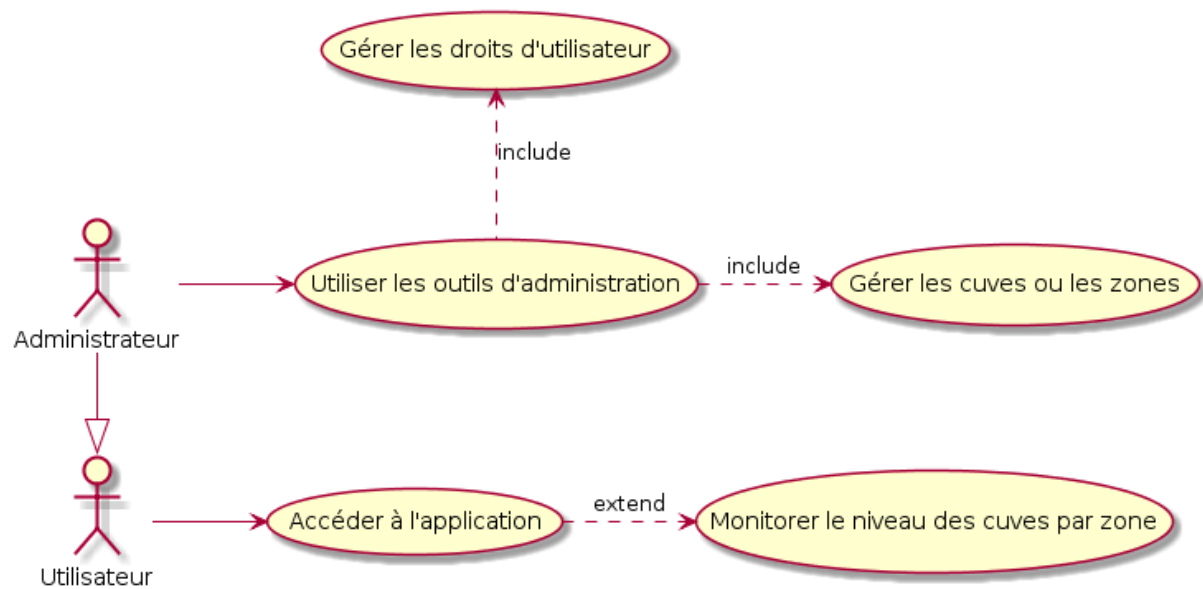
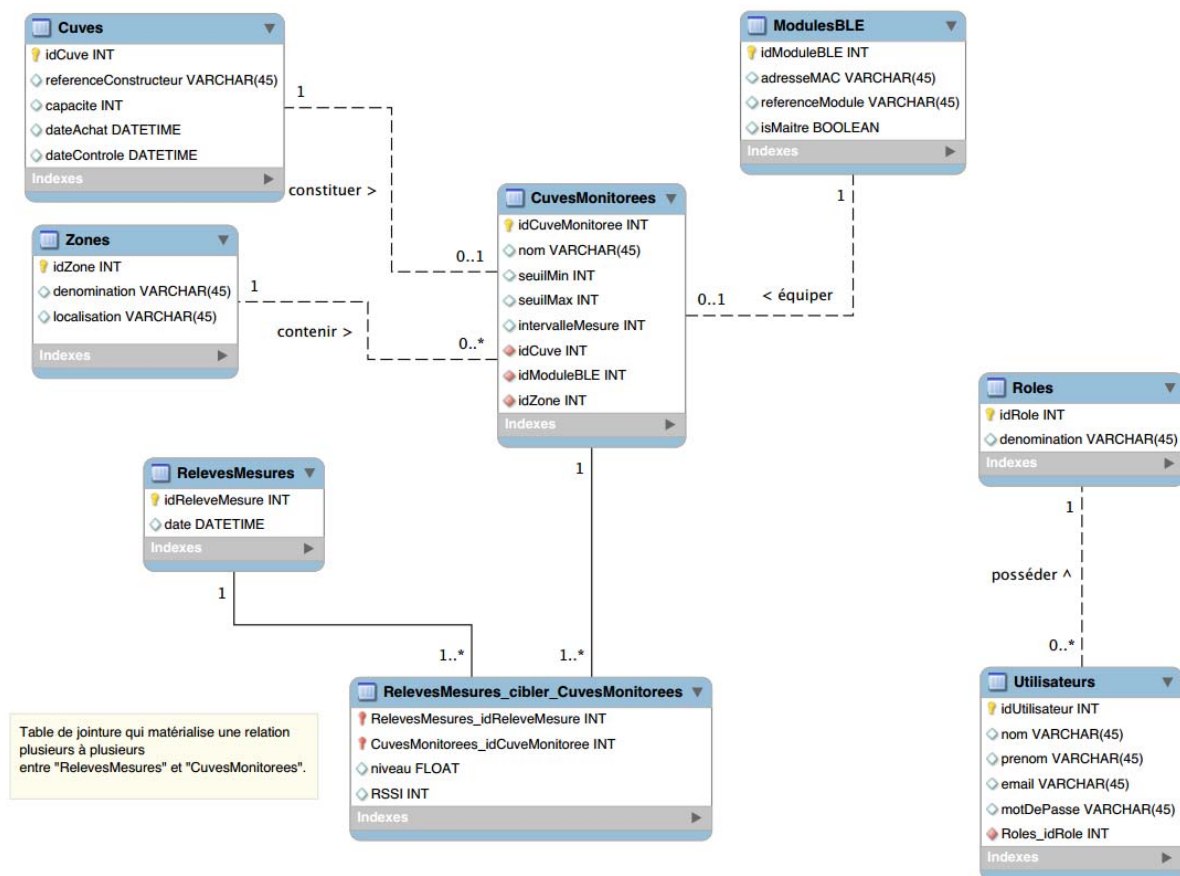


Figure 5: Diagramme des cas d'utilisation de l'application web

Figure 6: Modèle de données de la base de données



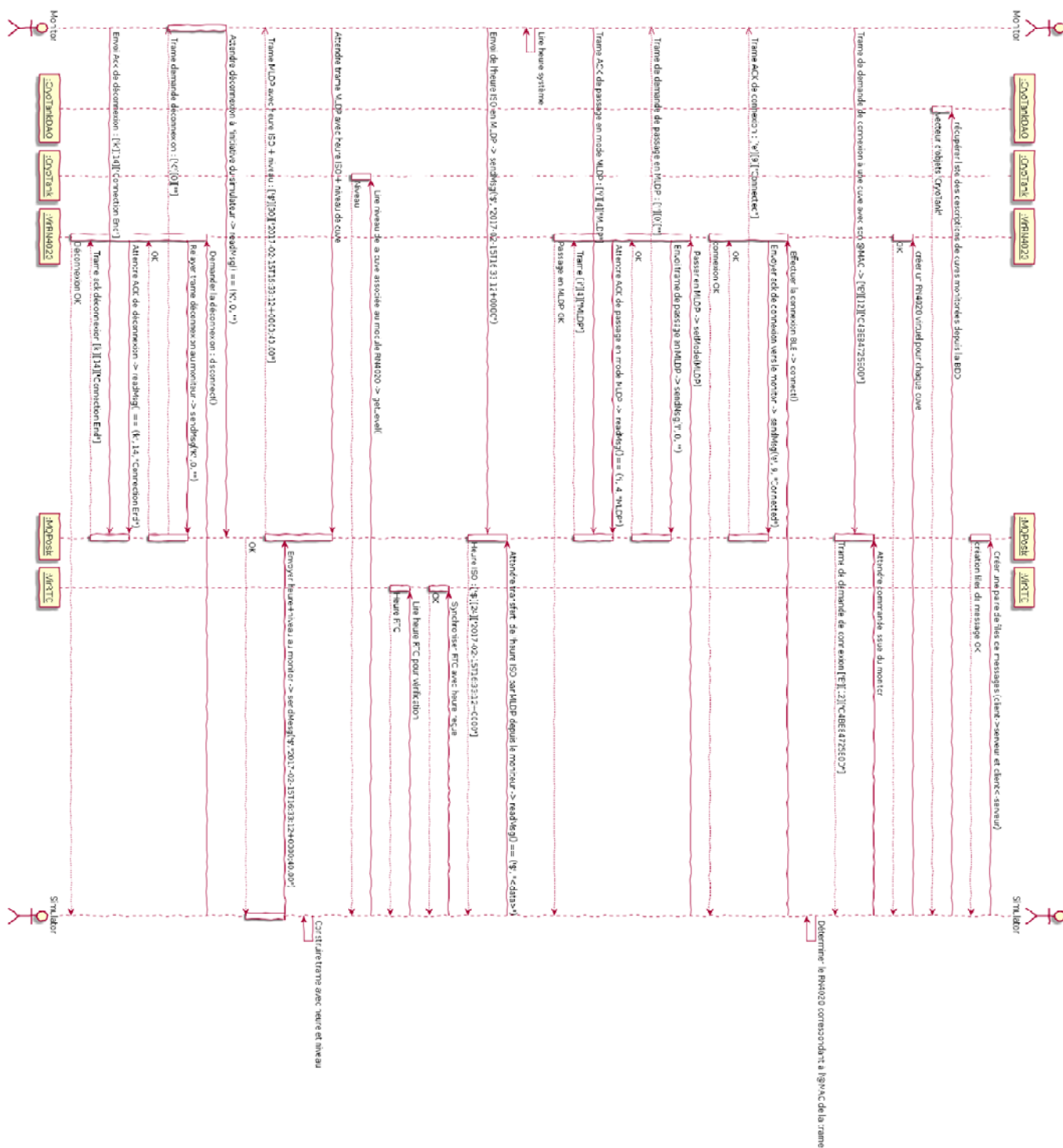


Figure 7: Diagramme de séquence du Simulateur



b) Planification prévisionnelle :

	Mode Tâche	Nom de la tâche	Durée	Début	Fin	Préc	Noms ressources
1		Debut du projet	0 jour	Lun 09/01/17	Lun 09/01/17		
2		Revue 1	0 jour	Mar 07/03/17	Mar 07/03/17		Anton De France
3		Revue 2	0 jour	Mar 02/05/17	Mar 02/05/17		Anton De France
4		Remise du projet	0 jour	Ven 26/05/17	Ven 26/05/17		Anton De France
5		Soutenance finale	0 jour	Lun 12/06/17	Lun 12/06/17		Anton De France
6		Analyse prévisionnelle	3 jours	Lun 02/01/17	Mar 10/01/17	1	Anton De Francesco
7		Tâches de mise en oeuvre	8 jours	Mar 10/01/17	Jeu 02/02/17	6	Anton De Francesco
8		▲ Faire évoluer le script d'injection des données en C++	26 hr	Ven 03/02/17	Jeu 02/03/17	7	Anton De Francesco
9		Acquérir les données d'une cuves au travers d'un simulateur	16 hr	Ven 03/02/17	Lun 27/02/17		
10		Acquérir les données de plusieurs cuves	10 hr	Mar 28/02/17	Jeu 02/03/17	9	
11		Intégrer la librairie d'acquisition de mesures réalisée par l'étudiant EC	8 hr	Ven 03/03/17	Mar 07/03/17	8	Anton De Francesco
12		▲ Concevoir/Coder/Tester l'évolution du site	70 hr	Mar 07/03/17	Jeu 06/04/17	11	Anton De Francesco
13		Le monitoring de plusieurs cuves	5 jours	Mar 07/03/17	Jeu 23/03/17		
14		Une gestion de cuves par zone	5 jours	Jeu 23/03/17	Jeu 06/04/17	13	
15		Alerter par mail en cas de dépassement d'un seuil programmé	20 hr	Jeu 06/04/17	Ven 28/04/17	12	Anton De Francesco

Figure 8: Diagramme de Gant prévisionnel

Le diagramme ci-dessus comporte trois itérations correspondant aux trois revues du projet.

La première itération consiste à **Faire évoluer le script d'injection des données en C++**, c'est-à-dire la création du simulateur de données. Cette itération s'est déroulée selon le planning prévu, mais l'intégration de la librairie d'acquisition de mesures n'a pas pu être effectuée faute d'un manque de matériel à cette époque.

La seconde itération consiste à **faire évoluer l'application web existante**. La prévision du temps que demandait cette itération a été mal jugée et elle a pris plus de temps que prévu, elle a donc été poursuivie lors de la troisième itération.

La troisième itération était cependant vide pour prévoir d'éventuels retards ainsi que pour les différents tests.



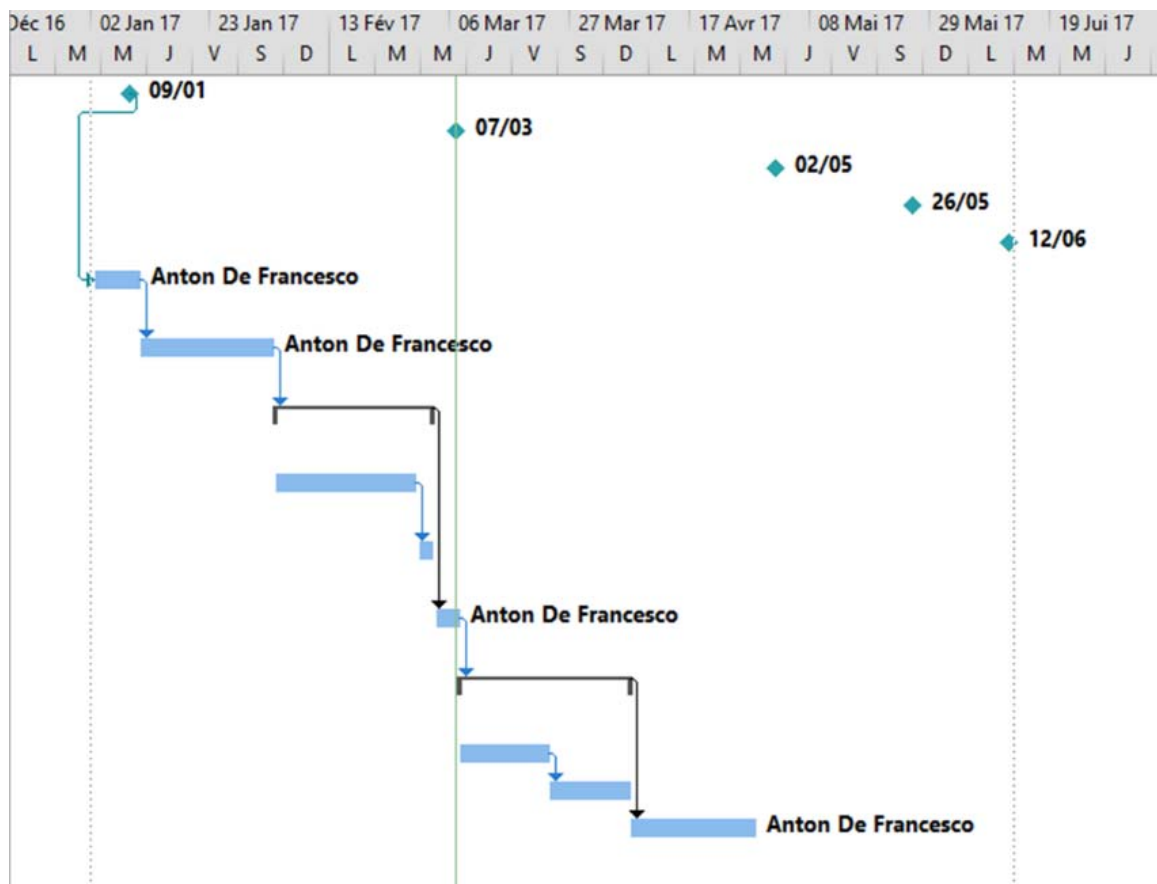


Figure 9: Vue du calendrier des itérations



2) Partie Physique : Bluetooth module RN4020 :

Comment fonctionne le Bluetooth :

Bluetooth est une technologie réseau personnel de communication sans fil mise au point par Ericsson en 1994.

Le Bluetooth utilise l'une des bandes de fréquences ISM (Industrial, Scientific & Medical) réservée pour l'industrie, la science et la médecine. La bande de fréquences utilisée en France s'étend sur 79 canaux de 1 MHz allant de 2400 à 2483,5 MHz.

Son utilisation est libre à l'intérieur des bâtiments pour des puissances d'émissions inférieures à 10mW et inférieures à 4mW à l'extérieur.

Il existe 3 classes d'émission permettant un contrôle de la puissance pour réduire le taux d'erreurs à un niveau acceptable, avec une consommation minimale.

- Classe 1 : puissance maximale émise 100 mW soit 20dBm, contrôle de la puissance par pas de 2 dB min / 8dB max dans la gamme 4 dBm - 20 dBm, soit une portée d'environ 100 mètres
- Classe 2 : puissance maximale 2,5 mW soit 4 dBm, contrôle de puissance facultatif, soit une portée d'environ 10-20 mètres
- Classe 3 : puissance maximale 1 mW soit 0 dBm, contrôle de puissance facultatif, soit une portée d'environ 1-10 mètres

La porteuse du Bluetooth est modulée en fréquence par le signal binaire filtrée par un filtre passe-bas gaussien (modulation GFSK). De plus la fréquence de la porteuse change régulièrement en fonction d'une séquence pseudo aléatoire.

Plus, spécifiquement le Bluetooth utilise 0.5 BT Gaussian-filtered two-frequency shift keying (2FSK), aussi appelé binary frequency shift keying (BPSK).

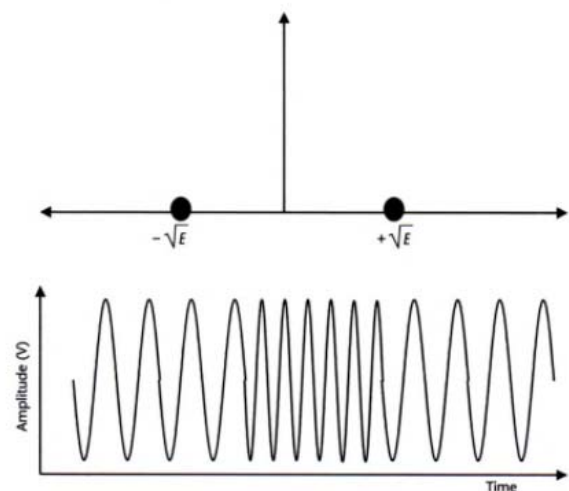


Figure 10: Constellation et amplitude sur le temps du 2FSK

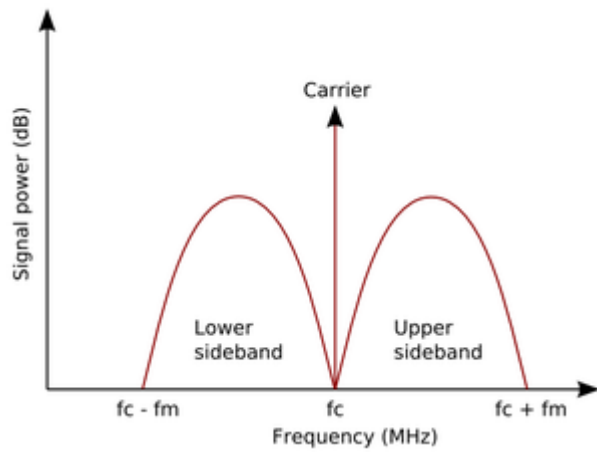


Figure 11: Bandes latérales d'une porteuse

L'intérêt de la modulation GFSK est son filtre de Gauss qui permet de réaliser des transitions plus douces entre chaque donnée. Ce filtre à l'avantage de réduire la puissance des bandes latérales, ce qui réduit les interférences avec les canaux voisins.



3) Travail accompli dans les itérations précédentes :

a) Découverte des langages utilisés sur le projet :

Mes tâches de mise en œuvre ont pris une part importante de ma première itération. En effet, le développement du projet nécessite des connaissances dans plusieurs langages de programmation. Certaines de mes connaissances sur ces langages étant limitées, j'ai pris la décision de me former.

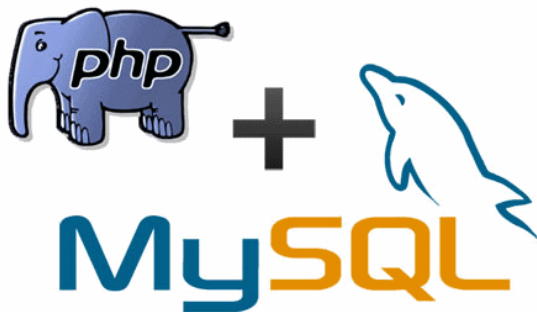
Mes formations ont eu lieu sur le MOOC OpenClassroom.



HTML 5 et CSS3 étant au cœur de l'application, j'ai eu besoin d'un rappel sur ces langages.

J'ai ainsi acquis les compétences suivantes :

- Base des langages
- Utilisation des media queries
- Utilisation des formulaires



PHP et SQL sont utilisés par l'application web ainsi que par la base de données (SQLite).

J'ai acquis les compétences suivantes :

- Base des langages



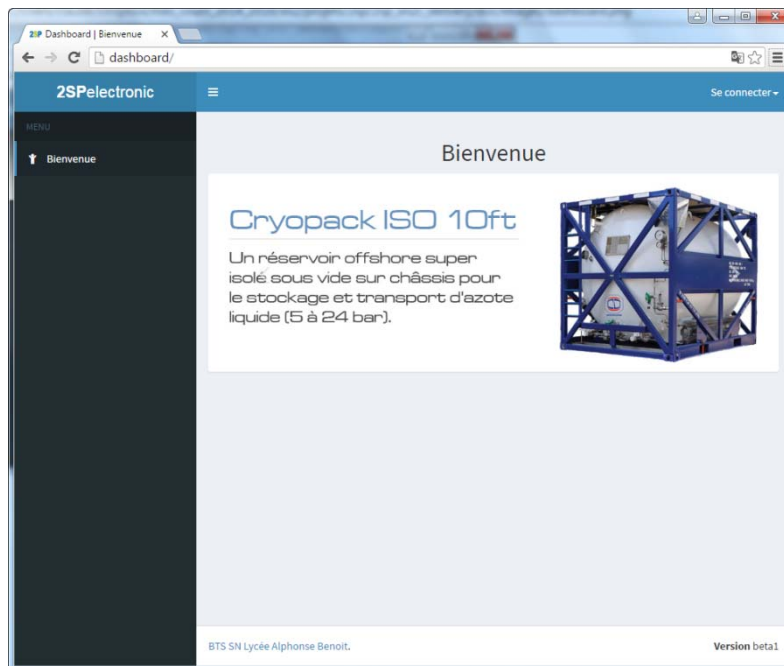
Les codes du projet étant stockés par Bitbucket, j'ai dû me former à l'utilisation de Git.



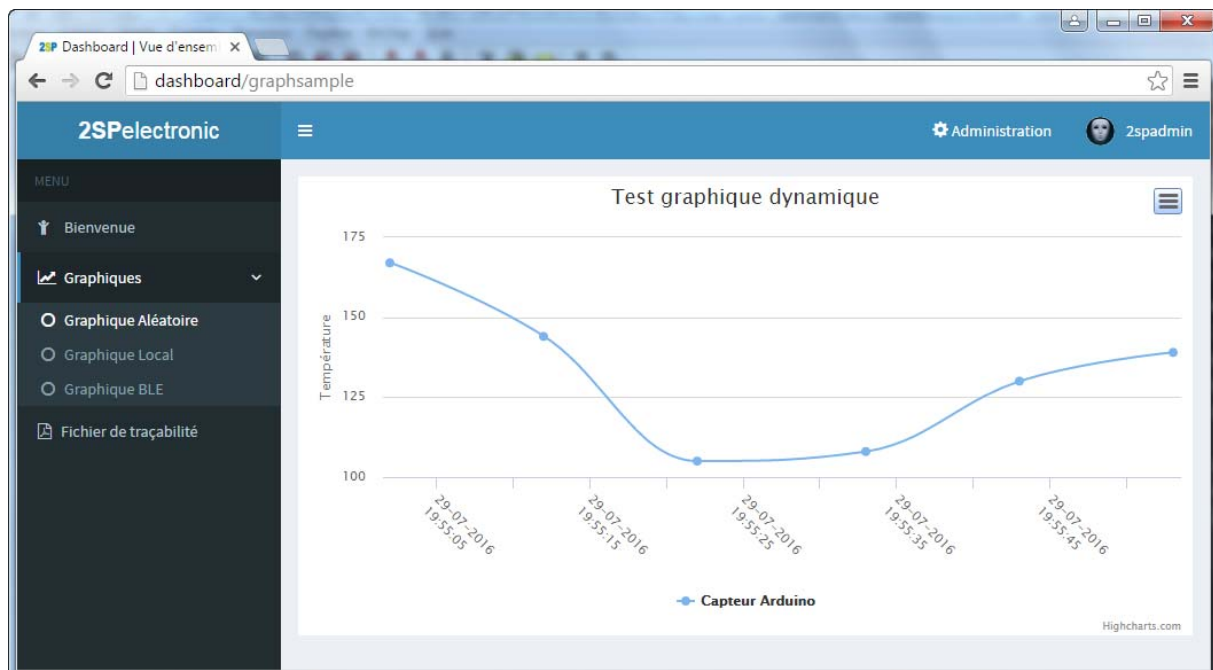
L'application web du projet utilise des scripts en JavaScript, notamment pour les graphiques.

b) Mise en place de l'application web existante :

La première action réalisée a été de remettre en place le site web existant pour en étudier le fonctionnement.



Celui-ci permet de superviser le niveau des cuves au travers de profils d'utilisateurs. De plus, l'application permet d'afficher le niveau des cuves en graphiques grâce à l'utilisation de la librairie HighCharts.



c) Simulateur de données et transfert Bluetooth :

La seconde action est d'intégrer les données des cuves à la base de données. Le problème étant que par manque de matériel nous n'avons pas pu réaliser de solution utilisant le module Bluetooth (RN4020) sur les raspberry.

La solution retenue a été de réaliser un programme simulateur envoyant les données à un autre programme Moniteur au travers des files de messages Posix.

Le programme Moniteur a pour charge l'intégration des données à la base de données.

d) Le programme Simulateur :

Ce programme commence par initialiser une file de messages dans laquelle les opérations de lecture et d'écriture sont non bloquantes. Selon que ce soit le simulateur ou le moniteur qui ouvre la file, celle-ci est ouverte différemment. Si c'est le simulateur qui l'ouvre il la crée. Si c'est le moniteur il se contente de l'interroger.

Par la suite le simulateur se connecte à la base de données pour récupérer la liste des cuves monitorées par le système grâce à la requête SQL suivante :

```
const char * sql = "SELECT CM.idCuve, CM.nom, C.capacite, CM.seuilMin,
CM.seuilMax"
    ", CM.intervalleMesure, Z.denomination, MB.adresseMAC,
MB.referenceModule "
    "FROM CuvesMonitorrees AS CM "
    "INNER JOIN ModulesBLE AS MB ON CM.idModuleBLE=MB.idModuleBLE "
    "INNER JOIN Zones AS Z ON CM.idZone=Z.idZone "
    "INNER JOIN Cuves AS C ON CM.idCuve = C.idCuve "
    "ORDER BY CM.intervalleMesure"
    ;
```

Le simulateur crée ensuite un module RN4020 virtuel pour chaque cuve, celui-ci émule le fonctionnement réel du module. Le simulateur attend ensuite la commande du monitor pour la connexion d'un module RN4020. Cette commande contient l'adresse MAC, unique par définition, qui lui permet d'effectuer une connexion virtuelle du module RN4020 virtuel. Le simulateur passe ensuite le module RN4020 virtuel en mode **MLDP** qui permet le transfert de données en Bluetooth.

Le simulateur attend ensuite l'heure envoyée par le moniteur qui permet de synchroniser les horloges des deux programmes. (Ceci étant important pour, à terme, synchroniser l'heure d'envoi des données, et ainsi économiser la batterie des modules esclaves).

Une fois l'heure synchronisée, le module construit la trame d'envoi de données de la cuve dont l'adresse MAC a été envoyée, qui est de la forme suivante :

```
sprintf(mldpData, "%s;%.02F", isoTime.c_str(), fakeLevel);
```

Où fakeLevel est le niveau de cuve généré, isoTime.c_str() est l'heure précédemment synchronisée.

Le simulateur envoie par la suite une demande de déconnexion pour arrêter le processus.



e) Le programme Moniteur :

Le programme Moniteur plus simple se contente d'envoyer une trame de connexion avec l'adresse MAC de la cuve qui doit être mesurée.

Il envoie ensuite l'heure ISO du système. Et attend par la suite la trame de retour avec l'heure du simulateur ainsi que le niveau de la cuve.

Après la réception de la demande de déconnexion, il envoie la trame de déconnexion au module RN4020 virtuel.

```

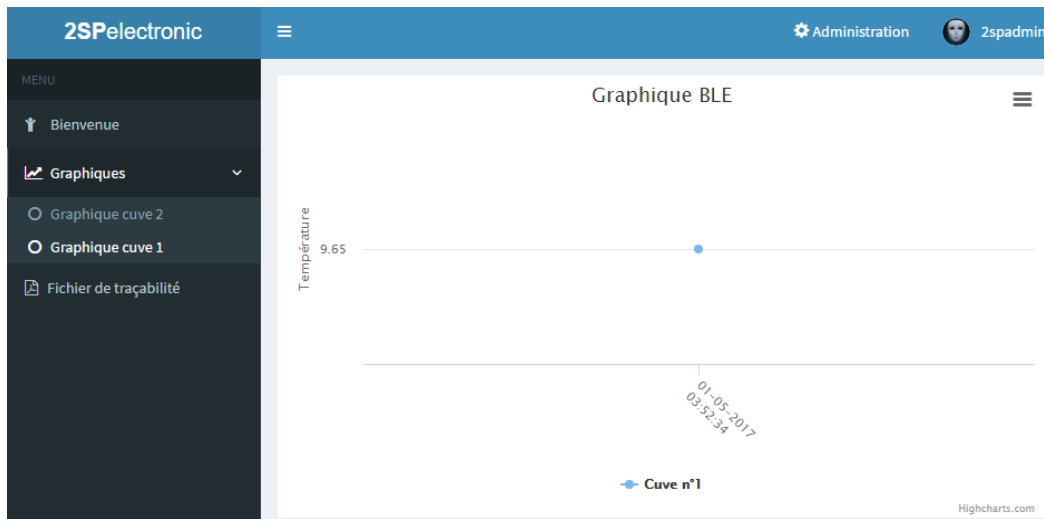
-----
Moniteur
-----
Connexion au RN4020@0024E40EA482(Withings Pulse)
> Données de cuve :
> . dénomination : CryoTank 02
> . capacité : 200
> . seuils : 20 <-> 30
> . zone : Salle EC
    <-(out)- [E][12][0024E40EA482]
    -(in)-> [e][9][Connected]
    -(in)-> [I][0][]
    <-(out)- [i][4][MLDP]
Traitement du passage en mode MLDP
    <-(out)- [$][24][2017-03-07T10:32:27+0000]
Envoi de l'heure en mode MLDP (2017-03-07T10:32:27+0000)
    -(in)-> [$][30][2017-03-07T10:32:28+0000;26.67]
Niveau de cuve reçu à 2017-03-07T10:32:28+0000 : 26.67
    -(in)-> [K][0][]
Demande de déconnexion
    <-(out)- [k][14][Connection End]
Déconnexion du RN4020@0024E40EA482
-----

```

Figure 12: Le moniteur de niveau

f) Prise en main et amélioration de l'application web :

La deuxième itération a eu pour but d'améliorer l'ancienne application web pour l'adapter à la nouvelle architecture de la base de données ainsi que d'améliorer la partie des graphiques de niveau du site.



Pour adapter l'application à la nouvelle base de données il a fallu modifier la requête de la base de données :

```
SELECT
RelevésMesures.date, // on récupère la date du relevé
RelevésMesures_cibler_CuvesMonitorees.RelevésMesures_idReleveMesure,
// on récupère l'id du relevé
RelevésMesures_cibler_CuvesMonitorees.niveau //on récupère enfin le niveau
FROM RelevésMesures INNER JOIN RelevésMesures_cibler_CuvesMonitorees
//on indique les tables dans lesquelles chercher les données
WHERE RelevésMesures_cibler_CuvesMonitorees.CuvesMonitorees_idCuveMonitoree=1
// on ne récupère que la cuve n°1
AND
RelevésMesures_cibler_CuvesMonitorees.RelevésMesures_idReleveMesure=RelevésMesures.idReleveMesure
//Les mesures doivent correspondre à leurs id de mesure
ORDER BY RelevésMesures_cibler_CuvesMonitorees.RelevésMesures_idReleveMesure
// on trie par id de relevé
DESC LIMIT 1; //on récupère la dernière mesure pour faire un point
```

Cette modification a lieu dans le fichier GraphController.php qui gère les fonctions s'occupant des données des graphiques des niveaux. Plus précisément dans la fonction **retrieveData** que voilà :

```
public function retrieveDataBLE(Application $app) {
    try {
        $db = new \SQLite3('../db/db2sp.db');
        $dataSet1 = $db->query('SELECT RelevésMesures.date,
RelevésMesures_cibler_CuvesMonitorees.niveau FROM RelevésMesures INNER JOIN
RelevésMesures_cibler_CuvesMonitorees ON
RelevésMesures_cibler_CuvesMonitorees.RelevésMesures_idReleveMesure=RelevésMesures.idReleveMesure
ORDER BY RelevésMesures_cibler_CuvesMonitorees.RelevésMesures_idReleveMesure DESC LIMIT 1;');
        $db = null;
    }
    catch (Exception $exception) {
        echo '<p>Error connecting to the database!</p>';
    }
    if($dataSet1) {
        // récupère les valeurs du dernier enregistrement
        $endRow = $dataSet1->fetchArray();
        //Si la bdd est vide le tableau sera [-1,-1], aucun point ne sera affiché
    }
}
```



```

        if (! $sendRow) {
            $ret = array(-1,-1);
        }
        else {
            // Créer un tableau PHP et l'envoyer en JSON
            $delta = 0 + $sendRow['niveau'];
            $t = $sendRow['date'] * 1000;
            $ret = array($t, $delta);
        }
    }
    return $app->json($ret);
}

```

Le deuxième objectif de la deuxième itération est de faire apparaître un graphique pour chaque cuve qui est monitorée.

L'objectif final est de réaliser cela de façon automatique, mais pour cette itération je présenterai comment le réaliser de façon manuelle.

Tout d'abord il faut réaliser deux fonctions dans **GraphController.php** portant le nom de la cuve que l'on veut ajouter avec la bonne requête SQL. (Ex : **showCuve1Action** et **retrieveDataCuve1**). Ces fonctions ont pour but de chercher les données dans la base de données et de les mettre à disposition du graphique ainsi que d'afficher le graphique.

Ces fonctions auront besoin d'avoir une route définie dans **routes.php** de cette façon :

```

$app->match('/graphCuve1',
"dashboard\Controller\GraphController::showCuve1Action")-
>bind('graph_Cuve1');

$app->match('/graphCuve1/livedata',
"dashboard\Controller\GraphController::retrieveDataCuve1")-
>bind('graph_Cuve1_livedata');

```

Ensuite il faut créer le fichier html qui affichera le graphique. Dans notre exemple celui-ci s'appellera **graphCuve1.html.twig**.

```

{% extends 'layout.html.twig' %}

{% block title %}Vue d'ensemble{% endblock %}

{% block content %}

<div id="container" style="min-width: 300px; height: 400px; margin:
0 auto></div>

{% endblock %}

{% block script %}

<script src="https://code.highcharts.com/highcharts.js"></script>

<script
src="https://code.highcharts.com/modules/exporting.js"></script>

<script>

    var chart;

    function requestData() {

```



```

$.ajax({
    url: '{ path('graph_Cuve1_livedata') }',

    success: function(point) {
        var series = chart.series[0];
        var shift = series.data.length > 5;
        var lastIdx = series.data.length > 0 ?
series.data.length - 1 : null;

        if (lastIdx != null) {
            console.log("series.data[" + lastIdx + "]
: " + series.data[lastIdx].x );
            console.log("point[0] : " + point[0] );
        }

        // Ajoute le point
        if((point[0] != -1) && (lastIdx == null ||
(series.data[ lastIdx ].x != point[0]))) {
            chart.series[0].addPoint(point, false, shift);
            chart.redraw();
        }

        // effectue la fonction RequestData toute les
10secondes

        setTimeout(requestData, 10000);
    },

    cache: false,

    complete: function(resultat, statut) {
        //alert("Ajax Complete : " + resultat + " / " +
statut);
    },

    error: function(exception){alert('Exception: ' +
exception);}

```




```
});  
  
}  
  
$(document).ready(function() {  
    Highcharts.setOptions({  
        global : {  
            useUTC : false  
        }  
    });  
    chart = new Highcharts.Chart({  
        chart: {  
            renderTo: 'container',  
            type: 'spline',  
            events: {  
                load: requestData  
            }  
        },  
        title: {  
            text: 'Graphique Cuve 1'  
        },  
        xAxis: {  
            //categories : [],  
            labels: {  
                formatter: function() {  
                    return Highcharts.dateFormat('%d-%m-%Y<br/>%H:%M:%S', this.value)  
                },  
                rotation: 45,  
                step: 2,  
                align: 'left',  
                y: 20  
            }  
        },  
        yAxis: {
```



```

        title: {
            text: 'Température'
        }
    },
    series: [{
        name: 'Cuve n°1',
        data: []
    }]
});
});
</script>
{% endblock %}

```

Enfin il faut insérer la page créée dans le menu du site web. Ceci se fait dans le fichier layout.html.twig

```

<ul class="treeview-menu">
    <li {% if app.request.attributes.get('_route') ==
'graph_Cuve1' %}class="active"{% endif %}><a href="{{
path('graph_Cuve1') }}"><i class="fa fa-circle-o"></i>Graphique cuve
1</a></li>
</ul>

```



II) Travail accompli dans la troisième itération :

1) Création d'une page de vue d'ensemble des cuves :

a) Défaut de l'application préexistante :

L'application qui existait l'année dernière ne proposait qu'une vue de supervision par cuve. En effet l'application ne devant fonctionner que pour une cuve, une vue centralisant toutes les cuves n'était pas nécessaire. Des graphiques étaient donc présents pour chaque cuve.

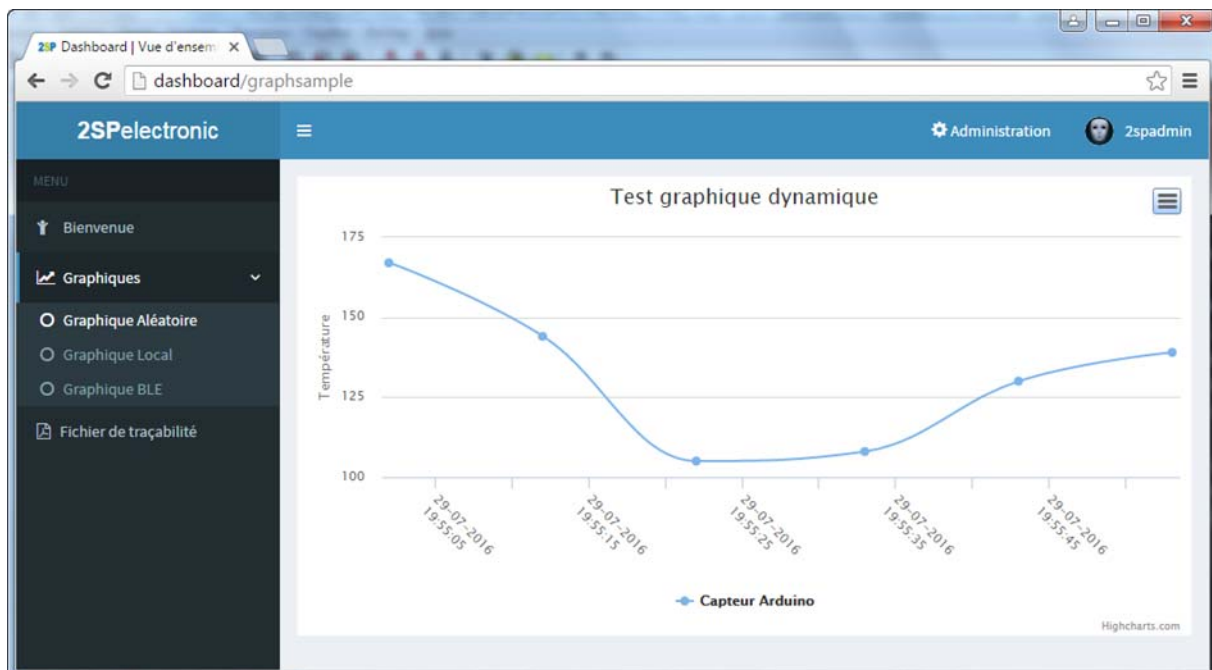


Figure 13: Graphique d'une cuve

Bien qu'étant intéressante pour surveiller l'évolution du niveau d'une cuve, cette architecture seule ne suffit pas pour la supervision d'un parc de cuves fonctionnant par zone.

La solution retenue fut celle de créer une page web, rassemblant toutes les cuves, triées par zones grâce à des onglets, permettant ainsi d'avoir une vue d'ensemble du système. Les cuves seraient visualisées grâce à des jauges intuitives.



Figure 14: Objectif pour la troisième itération

b) Mise en œuvre :

Une page fut créée, appelée `infoscuves.html.twig`, cette page est invoquée par le controller `TankController.php`. Ce controller appelle automatiquement le dao `dao.zone` qui va créer des objets zone pour chaque zone que contient la base de données grâce à la requête suivante :

```
$sql = "SELECT * FROM Zones ORDER BY idZone";
```

Un objet zone lorsqu'il est créé contient le numéro de la zone, sa dénomination, sa localisation ainsi que le nombre de cuves qu'une zone contient.

La page `infoscuves` va ensuite ajouter des onglets en fonction du nombre de zones et leur attribuer un titre qui contient le nom de la zone et le nombre de cuves dans cette zone.

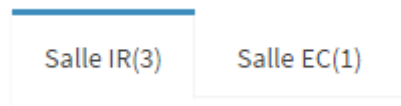


Figure 15: Onglets des zones de la vue d'ensemble

```
{% for zone in zones %}
<li {% if first %}class="active" {% endif %}><a
href="#ztab_{{ zone.id }}" data-toggle="tab" title="{{ zone.position }}">{{
zone.label }} ({{ zone.nbTanks }})</a></li>
{% endfor %}
```

On remarque ainsi l'appel des objets zones ainsi que les valeurs qu'ils contiennent (ex : `zone.nbTanks`).

Chaque onglet est un lien vers un contenant de la page qui contient les jauges. Celle-ci sont codées en JavaScript via le plugin Highcharts.



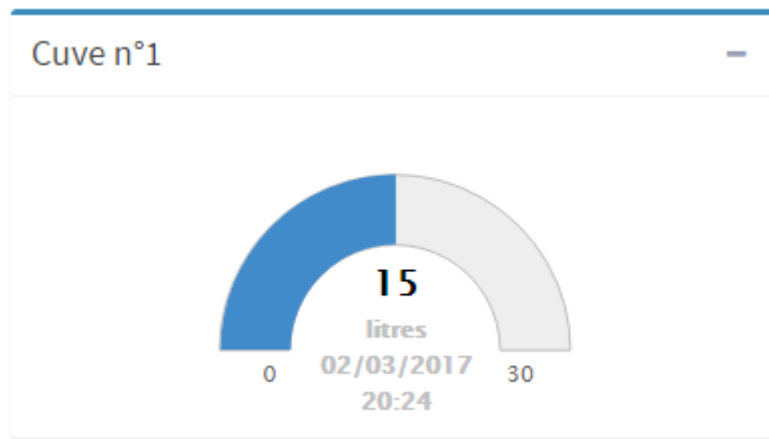


Figure 16: Une jauge de Highcharts

Pour qu'une jauge se mette à jour, il faut que côté client une nouvelle connexion soit effectuée vers le serveur, pour cela l'application utilise la technologie AJAX.

Une nouvelle page web est ainsi créée, fournissant les informations demandées au format JSON.

```
[{"idCuveMonitoree": "1", "nom": "CryoTank 01", "niveau": "15.0", "RSSI": null, "date": "1488479041"}, {"idCuveMonitoree": "3", "nom": "CryoTank 03", "niveau": "33.09", "RSSI": null, "date": "1488478981"}, {"idCuveMonitoree": "4", "nom": "CryoTank 04", "niveau": "99.92", "RSSI": null, "date": "1488478801"}]
```

Figure 17: Données des cuves de la zone 1 au format JSON

Cette page est invoquée grâce au controller ApiController.php, qui fait appel aux objets dao cuvemontoree, où chaque objet représente une cuve monitorée de la base de données. Des requête SQL lors de la création de ces objets permettent de récupérer toutes les informations les concernant. De plus une requête a été conçue pour récupérer dans la base de données tous les niveaux des cuves monitorées d'une zone spécifiée.

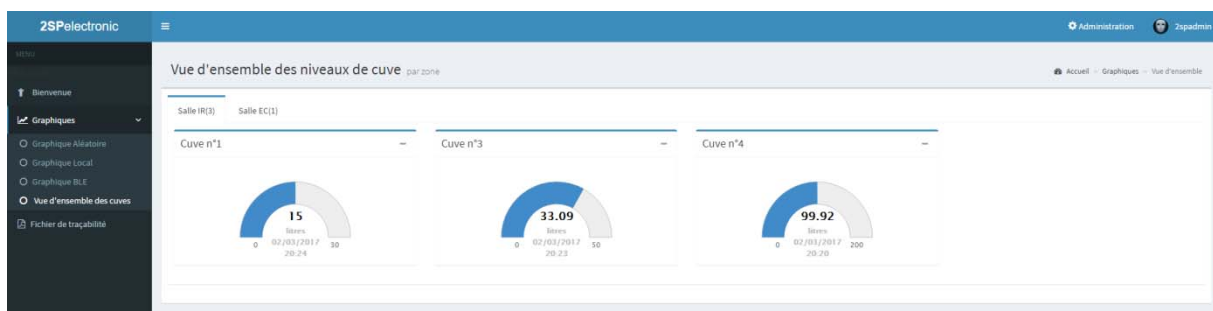


Figure 18: Rendu final de la page infoscuves

2) Conclusion :

a) Bilan :

Pour faire le bilan de ce projet il faut tout d'abord revenir sur les difficultés rencontrées. La cohésion de l'équipe fut un des facteurs d'obstacles car malgré une bonne entente des membres, et les efforts fournis par chacun pour communiquer autour du projet, certaines questions concernant la prévision furent réglées tardivement, ce qui a déteint sur le résultat final. De plus un stress important fut généré par ce projet qui fut mal géré personnellement.

Pour ma part, ces erreurs proviennent probablement du fait que ce fut ma première expérience sur la gestion et le déroulement d'un projet d'une envergure importante.

Cependant j'en retire un grand bénéfice car le défi fut à la hauteur de mes attentes. Je souhaite faire de la gestion de projet ma future vie professionnelle, et j'espère que je saurai tirer profit de mes erreurs présentes.

b) Pistes d'amélioration :

Il me semble qu'il faudrait prendre en considération d'améliorer davantage l'application web pour pouvoir créer une page qui rassemblerait toutes les informations d'une cuve, tel que son niveau en jauge, sa contenance, sa date de création ainsi que son graphique d'évolution, le tout accessible depuis la page de vue d'ensemble des cuves par zone, et dont les paramètres seraient modifiables depuis cette page même.

Je tiens à remercier tout particulièrement mon professeur responsable **Mr Claude DEFRANCE** pour sa patience admirable, et son soutien sans faille.






```

    </div>
    <!-- /.tab-content -->
  </div>
  <!-- nav-tabs-custom -->
</div>
<!-- /.box-body -->
</div>
<!-- /.box -->
</div>
<!-- /.row -->
</section>
{% endblock %}

{% block script %}
<!-- page script -->

<script type="text/javascript"
src="/lib/highcharts/highcharts.js"></script>
<script type="text/javascript" src="/lib/highcharts/highcharts-
more.js"></script>
<script type="text/javascript" src="/lib/highcharts/modules/solid-
gauge.js"></script>

<script>
$( document ).ready(function() {
  var levelsGauges = [];
  var intervalId = null;

  // On configure Higcharts pour le décalage horaire en France
  Highcharts.setOptions({
    global: {
      timezoneOffset: -2 * 60
    }
  });

  // Options communes à toutes les courbes
  var commonGaugeOptions = {
    chart: {
      type: 'solidgauge',
      //events: {
      //  load: requestData
      //}
    },
    title: null,
    pane: {
      center: ['50%', '85%'],
      size: "140%",
      startAngle: -90,
      endAngle: 90,
      background: {
        backgroundColor: (Highcharts.theme &&
Highcharts.theme.background2) || '#EEE',
        innerRadius: '60%',
        outerRadius: '100%',

```



```

        shape: 'arc'
      },
    },
    tooltip: {
      enabled: false
    },
    yAxis: {
      lineWidth: 0,
      minorTickInterval: null,
      //tickPixelInterval: 240,
      tickAmount: 1,
      startOnTick: false,
      endOnTick: false,
      tickWidth: 0,
      title: {
        y: -60
      },
      labels: {
        enabled: true,
        //x : 35, y : -10,
        x: 0,
        y: 16,
      },
      //min: -194,
      //max: -182,
      stops: [
        //:TODO: A déterminer avec les seuils min et max de
chaque cuve => à déporter dans les options spécifiques à chaque cuve
        [1.00, '#428bca'], // red
        //[0.00, '#00cc00'], // green
        //[0.41, '#00cc00'], // green
        //[0.42, '#ff9900'], // orange
        //[0.58, '#ff9900'], // orange
        //[0.59, '#ff0000'], // red
        //[1.00, '#ff0000'], // red
      ],
    },
    credits: {
      enabled: false
    },
    plotOptions: {
      solidgauge: {
        dataLabels: {
          y: 40,
          borderWidth: 0,
          useHTML: true
        }
      }
    }
  };

  /**
   * Récupère de l'appli web les niveaux courants des cuves de la
   zone courante, met à jour les gauges

```



```

    * et lance un timer de 2s pour réitérer l'opération
    */
function updateCurrentLevels() {
    var target = $(".nav-tabs li.active a").attr("href");
    //console.log("onglet courant = " + target);
    var basePath =
"http://dashboard/api/monitoredtank/zonelevels/";
    //
    var urlWebApp = basePath +
target.substring(target.lastIndexOf("_") + 1);

    console.log("urlWebApp for levels : " + urlWebApp);
    $.getJSON(urlWebApp, function (levels) {
        $.each(levels, function (levelIdx, levelData) {
            levelsGauges[ levelIdx ].series[0].removePoint(0,
false);

            var currLevel = Number(levelData.niveau);
            var valueDate = Number(levelData.date)*1000;
            levelsGauges[ levelIdx
].series[0].addPoint([valueDate, currLevel], true);
            console.log("series[0].data[0] : [" + levelsGauges[
levelIdx ].series[0].data[0].x
+ ", " + levelsGauges[ levelIdx
].series[0].data[0].y + "]);
        })
    }).success(function() {
        if(intervalId === null) {
            intervalId = setInterval(updateCurrentLevels, 2000);
        }
    }).error(function() {
        console.log("Erreur getJSON(" + urlWebApp + ")");
    });
}

/**
 * Fonction appelée lorsqu'on change d'onglet
 */

var nbTanks;

/**
 * Fonction de callback appelée sur chaque sélection d'onglet.
 * Son rôle est d'afficher un ensemble de boîtes Bootstrap dans
lesquelles seront affichées
 * des jauges HighCharts qui reflètent en "temps réel" le niveau
des cuves
 */
$('.nav-tabs a').on('shown.bs.tab', function (event) {
    var activeTabLabel = $(event.target).text();
    //console.log("activeTabLabel = " + activeTabLabel);
    var target = $(event.target).attr("href"); // activated tab
    //console.log("target = " + target);
    var basePath = "http://dashboard/api/monitoredtank/zone/";

```



```
//
var urlWebApp = basePath +
target.substring(target.lastIndexOf("_") + 1);
console.log("urlWebApp : " + urlWebApp);

var containerBaseName = "container-gauge";
var containerName = "";

// Reinitialiser à chaque changement d'onglet le tableau qui
mémoise les gauges de niveau
levelsGauges = [];

if(intervalId !== null) {
  clearInterval(intervalId);
  intervalId = null;
}
// On efface le contenu du panneau
$(target).empty();

// Récupérer les infos sur les cuves de la zone actuelle
$.getJSON(urlWebApp, function (mtks) {
  // Crée des containers JQuery vides pour utilisation
ultérieure
  $row = $(); // variable de travail pour élaborer une
ligne de boîtes
  $wrap = $(); // variable de travail pour élaborer un
certan nombre de lignes de boîtes
  $html = $(); // variable qui contient le contenu complet
du panneau

  // Nombre max de boîtes par ligne
  colsPerWrap = 4;

  // Mémoise le nb de cuves présentes dans la zone
  nbTanks = mtks.length;
  //console.log(mtk.length);

  // Fixe les options communes pour les courbes
  //Highcharts.setOptions(commonGaugeOptions);

  // Ajoute pour chaque cuve de la zone une "boîte"
contenant les infos sur celle-ci et un container pour une gauge
  $.each(mtk, function (mtkIdx, mtkData) {
    // Crée un objet JQuery pour chaque boîte associée à
1 cuve
    var $col = $("", {
      'class': 'col-md-3',
      'html': '<div class="box box-primary"><div
class="box-header with-border"><h3 class="box-title">Cuve n°'
      + mtkData.id
      + '</h3><div class="box-tools pull-
right"><button type="button" class="btn btn-box-tool" data-
```



```

widget="collapse"><i class="fa fa-
minus"></i></button></div></div><div class="box-body">
    //+ '<p> Id : '
    //+ mtkData.id
    //+ ' / nom : '
    //+ mtkData.nom
    //+ ' / seuilMin : '
    //+ mtkData.seuilMin
    //+ ' / seuilMax : '
    //+ mtkData.seuilMax
    //+ ' / intervalleMesure : '
    //+ mtkData.intervalleMesure
    //+ '\</p>'
    + '<div id="container-gauge' + mtkData.id
    + '" style="height:150px" ></div></div></div>'
  });

  // Ajouter cette boîte à la ligne
  $row = $row.add($col);

  // "Vide" les containers lorsqu'on arrive en fin de
  ligne ou fin de liste
  if (((mtkIdx + 1) % colsPerWrap === 0) || (mtkIdx ==
  mtkData.length - 1)) {
    // Crée un bloc pour chaque ligne
    var $wrap = $("<div/>", { 'class': 'row' });
    $wrap.append($row);
    // Vide la ligne pour la prochaine itération
    $rows = $();
    // Ajoute le bloc au contenu HTML du panneau
    $html = $html.add($wrap);
  }

  }); // $.each()

  // Injecte le html dans le panneau de l'onglet
  $(target).append($html);

  // Insère une gauge semi-circulaire dans le container
  associé de chaque cuve
  // :TODO: essayer de représenter les seuils => voir
  https://jsfiddle.net/ym2tvzy7/1/
  $.each(mtkData, function (mtkIdx, mtkData) {
    console.log("mtkData.seuilMin : " +
    mtkData.seuilMin);
    // créer courbe pour chaque cuve
    containerName = containerBaseName + mtkData.id;
    console.log("=> " + containerName);

    levelsGauges.push(Highcharts.chart(containerName,
    Highcharts.merge(commonGaugeOptions, {
      yAxis: {
        min : 0,
        max : mtkData.capacite,

```



```

        tickPositions: [0, mtkData.capacite],
        //tickInterval:(mtkData.capacite/1000)
    },
    series: [{
        name: 'Niveau',
        data: [],
        dataLabels: {
            formatter: function() {
                return '<div style="text-align:center"><span style="font-size:18px;color:' +
                    ((Highcharts.theme &&
                    Highcharts.theme.contrastTextColor) || 'black') + '">' +
                    this.series.data[0].y + '</span><br/>' +
                    '<span style="font-size:12px;color:silver">litres<br />' +
                    Highcharts.dateFormat('%d/%m/%Y<br/>%H:%M', this.series.data[0].x) +
                    '</span></div>'
            }
        },
        tooltip: {
            valueSuffix: ' l'
        }
    }]
    }));
}); // $.each()

}).success(function() {
    // Amorce la mise à jour de des données
    //console.log("levelsGauges.length : " +
    levelsGauges[0].container);
    //intervalId = setInterval(updateCurrentLevels, 2000);
    updateCurrentLevels();
}).error(function() {
    console.log("Erreur getJSON()");
}); // $.getJSON()

}); // $('nav-tabs a').on('shown.bs.tab', function(event)

// Déclenche le ré-affichage de l'onglet "active" pour mettre à
jour son contenu
//$('nav-tabs a:first').tab('show'); -> ne fonctionne pas
$('nav-tabs a').trigger('shown.bs.tab'); // -> fonctionne à peu
près mais provoque 2 requetes JSON de recuperation des niveaux
actuels => les gauges ne sont pas toujours correctement renseignées
});
</script>
{% endblock %}

```



2) Journal de bord :

Journal de bord.		
Date	Intervenant	Objet
09/01/2017	X	<input checked="" type="checkbox"/> Mise en fonctionnalité de la Raspberry - SSH - Compte utilisateur <input checked="" type="checkbox"/> Mise en place de l'environnement de travail - Drive - Bitbucket
10/01/2017	X	<input checked="" type="checkbox"/> Tutoriel de mise en route Bitbucket <input checked="" type="checkbox"/> Mise en fonctionnalité de la Raspberry - Répertoire Bitbucket
12/01/2017 au 16/01/2017	X	<input checked="" type="checkbox"/> Tâche de mise en œuvre HTML/CSS
17/01/2017 au 19/01/2017	X	<input checked="" type="checkbox"/> Tâche de mise en œuvre PHP
23/01/2017	X	<input checked="" type="checkbox"/> Mise en place du site web existant
24/01/2017	M.DEFRANCE	<input checked="" type="checkbox"/> Réunion avec le client pour fixer les objectifs <input checked="" type="checkbox"/> Mise en place du site web existant
26/01/2017	X	<input checked="" type="checkbox"/> Tâche de mise en œuvre PHP/Silex
30/01/2017 au 31/01/2017	X	<input checked="" type="checkbox"/> Programmation interface RN4020
02/02/2017	X	<input checked="" type="checkbox"/> Tâche de mise en œuvre PHP/Silex
06/02/2017	X	<input checked="" type="checkbox"/> Planification du projet sur Projet 2010 <input checked="" type="checkbox"/> Réunion avec le client
07/02/2017	X	<input checked="" type="checkbox"/> Planification du projet sur Projet 2010 <input checked="" type="checkbox"/> Documentation du projet
28/02/2017 au 03/03/2017	X	<input checked="" type="checkbox"/> Réalisation du script d'injection des données
• 21/03/2017	• Tâches réalisées ou en cours : - Environnement de développement (Netbeans) configuré pour compiler correctement la	



	<p>librairie partagée 2spSharedLib et l'exécutable du simulateur de Cuves (MultiTanksMonitor) → 100%</p> <ul style="list-style-type: none"> • Définition des tâches à Réaliser pour le 28/03/2017 <p>Refaire fonctionner l'ancienne appli web avec la nouvelle BDD 20%</p> <p>Savoir utiliser le simulateur de cuves pour mettre à jour la BDD et ainsi visualiser une évolution de la courbe de suivi de niveau de la cuve unique prise en charge actuellement par l'appli web 100%</p>
• 28/03/2017	<ul style="list-style-type: none"> • Tâches réalisées ou en cours : <p>Refaire fonctionner l'ancienne appli web avec la nouvelle BDD 20%</p> <p>Suite à la restructuration de la BDD, l'appli web ne pioche plus les bonnes valeurs dans les bonnes tables.</p> <p>Savoir utiliser le simulateur de cuves pour mettre à jour la BDD et ainsi visualiser une évolution de la courbe de suivi de niveau de la cuve unique prise en charge actuellement par l'appli web 100%</p> <ul style="list-style-type: none"> • Définition des tâches à Réaliser pour le 04/04/2017 <p>Modifier l'appli web pour exécuter les bonnes requêtes SQL</p> <p>Réparer l'erreur lors de l'édition de cuves monitorées</p> <p>Ajouter sur BitBucket, l'appli web dans 2sp_2017 (code source, composer.json)</p>
• 04/04/2017	<ul style="list-style-type: none"> • Tâches réalisées ou en cours : <p>Modifier l'appli web pour exécuter les bonnes requêtes SQL, pour afficher les valeurs de la cuve 1. 100%</p> <p>Graphcontroller.php modifié</p> <p>Ajouter sur BitBucket, l'appli web dans 2sp_2017 (code source, composer.json). 100%</p> <ul style="list-style-type: none"> • Définition des tâches à Réaliser pour le 25/04/2017 <p>Réparer l'erreur lors de l'édition de cuves monitorées. 10%</p> <p>Diagnostic de l'erreur : mauvais type d'objet.</p> <p>Regarder les autres onglets pour voir comment c'est traité.</p> <p>Réfléchir au plan du dossier de la revue 2.</p> <p>Faire 2 courbes pour 2 cuves.</p>

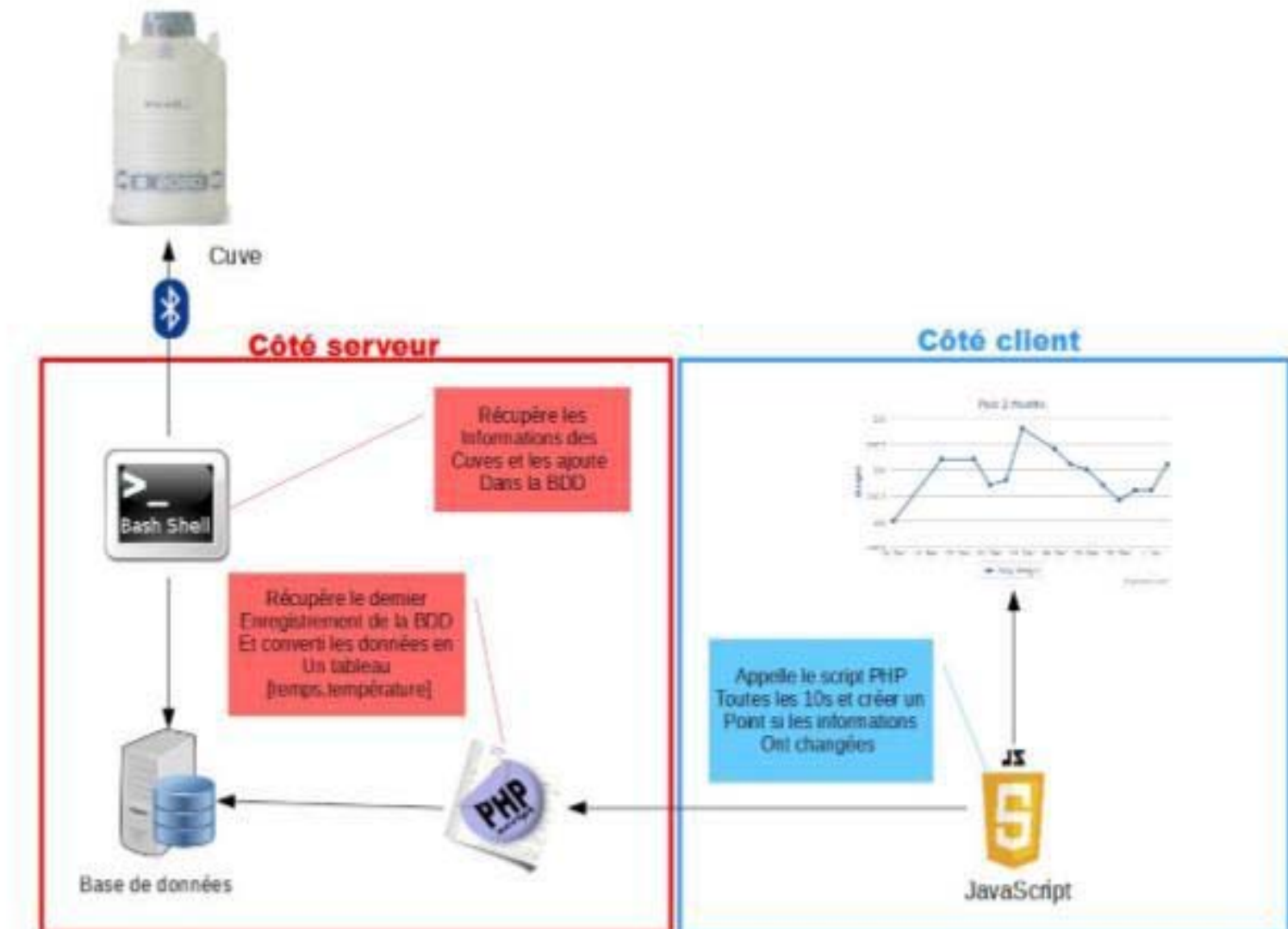


Etudiant IR2 (STIN Vincent) :

I) Rappel du projet :

Pour notre projet, nous avons eu le choix parmi 4 sujets, je me suis donc inscrit pour le projet « Surveillance de cuves cryogéniques » donné par l'entreprise 2SP et j'ai été assigné à celui-ci. Le projet consiste à surveiller le niveau de produits qui se trouvent dans des cuves cryogéniques. Pour cela nous avons des capteurs qui reçoivent des données qui sont par la suite stockées dans une base de données (BDD). Une fois stockées, les données pourront être récupérées pour être affichées sur un site qui servira de passerelle entre le client et les mesures effectuées. Le client aura aussi à sa disposition un moyen de récupérer les informations qu'il désire sous forme de PDF et il pourra y accéder même depuis son téléphone.

(Schéma résumant le système (cf annexe page 104))



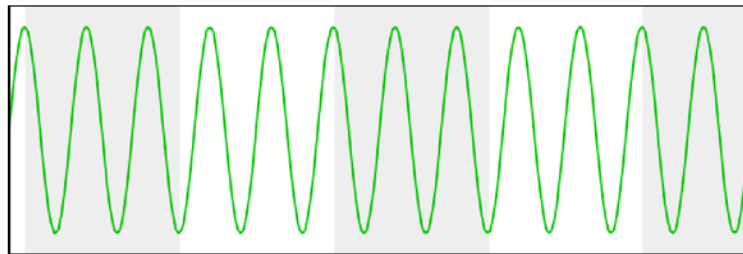
Pour notre projet, notre client désire que la communication entre le coffret maître et le coffret esclave qui se fait par Bluetooth soit le plus économique possible en énergie . Pour ce faire, l'électronicien du groupe a donc du utiliser une modulation, la modulation sert à convertir un signal en un signal adapté au mode de transmission utilisé.

Il à donc réfléchi à différentes modulations et il a fini par choisir entre la modulation FSK (Frequency shift keying, en français MDF pour modulation par déplacement de fréquence) qui sont utilisées pour la communication Bluetooth :

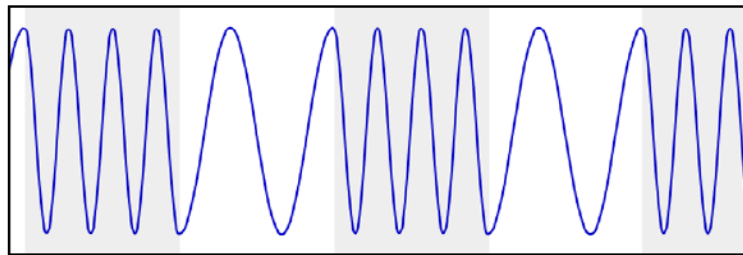
On peut voir ci-dessous les différents signaux liés à la modulation FSK :



Signal binaire



Onde porteuse



Signal modulé

On peut voir que le signal binaire passe du 0 à 1 quand il y a un changement de fréquence.

Et la modulation GFSK (Gaussian frequency shift keying) :

Cette modulation est semblable à la FSK mais le signal passe d'abord par un filtre Gaussien avant d'être traité par un modulateur. Cela apporte un changement de phase plus étroit et donc une meilleure efficacité tout en réduisant la puissance utilisée et donc de réduire les interférences qu'il pourrait y avoir.

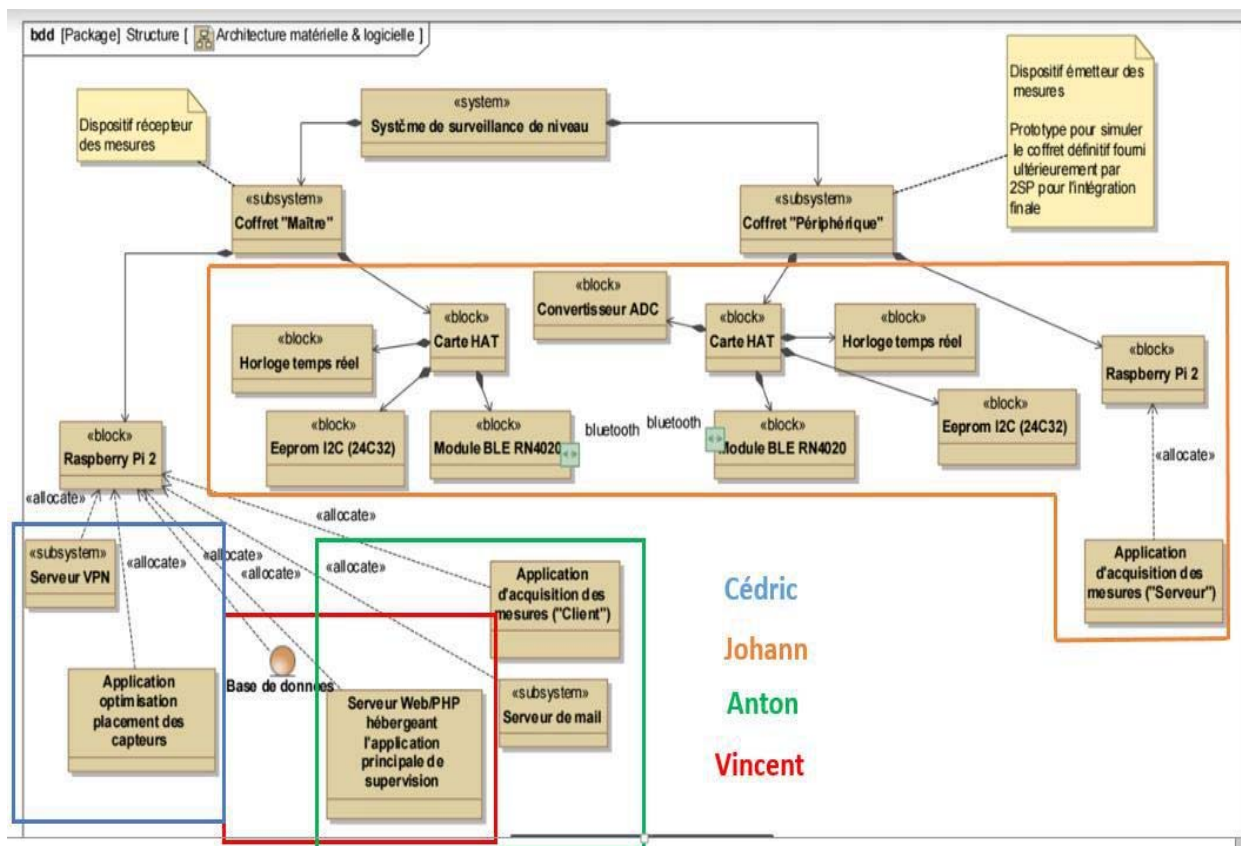
La GFSK est donc une FSK améliorée, qui permet de gagner également en bande passante.

II) Présentation de mes tâches :

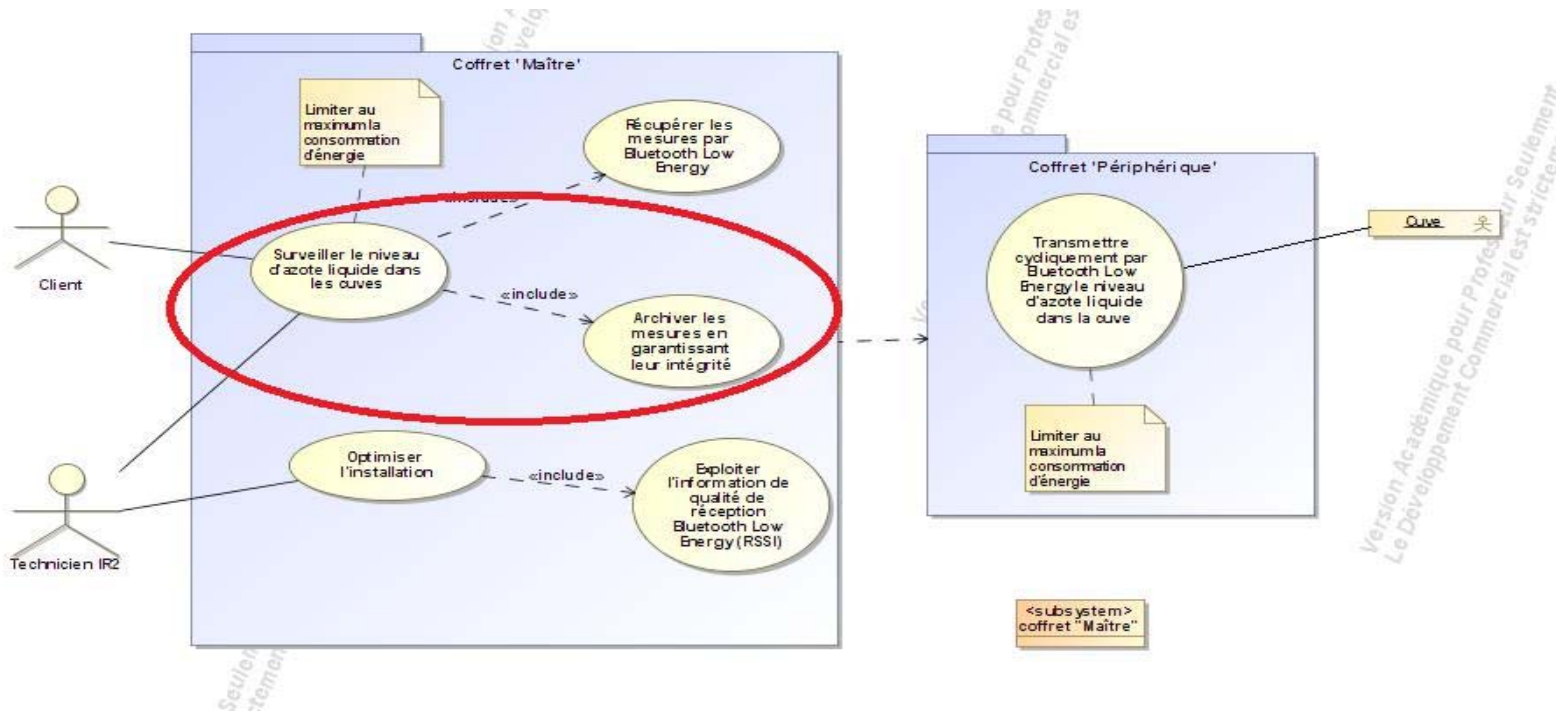
Pour le projet, je dois concevoir la partie du site web en charge de la génération des fichiers PDF qui feront office de moyen de traçabilité. Le PDF devra afficher les informations de la BDD (au format SQLite) à partir de critères de sélection choisis et je dois mettre en œuvre un système de chiffrement des informations de la base de donnée pour interdire toute falsification des données. Une personne quelconque ne doit effectivement pas pouvoir lire les vraies informations relatives aux cuves. Elles doivent être uniquement lisibles par les personnes autorisées.

III) Les diagrammes :

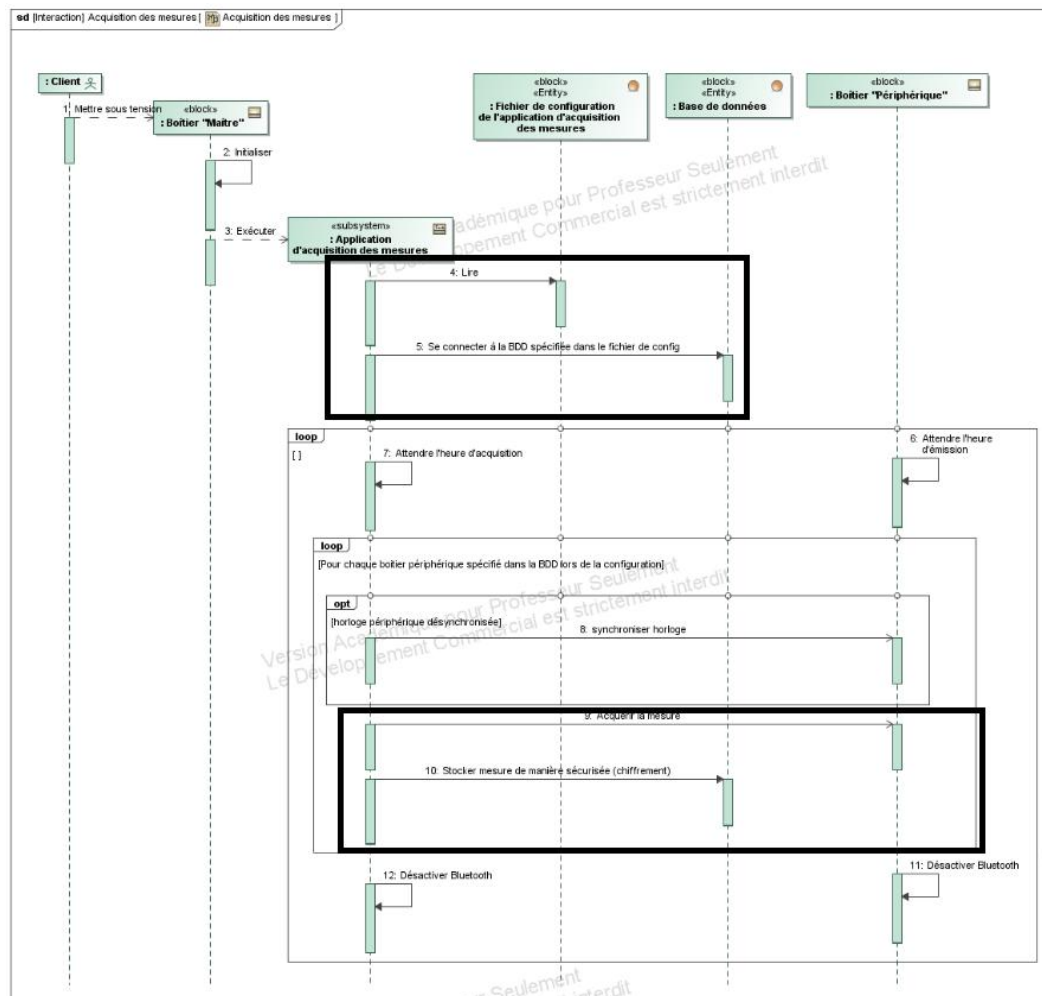
Voici le **diagramme de bloc** qui exprime ce que l'équipe doit gérer tout au long du projet. Nous avons entouré ce que chaque étudiant doit gérer (cf annexes pages 105-106) :



Voici ci-après le diagramme des cas d'utilisation qui montre les frontières de mon intervention (cf annexe page 107) :



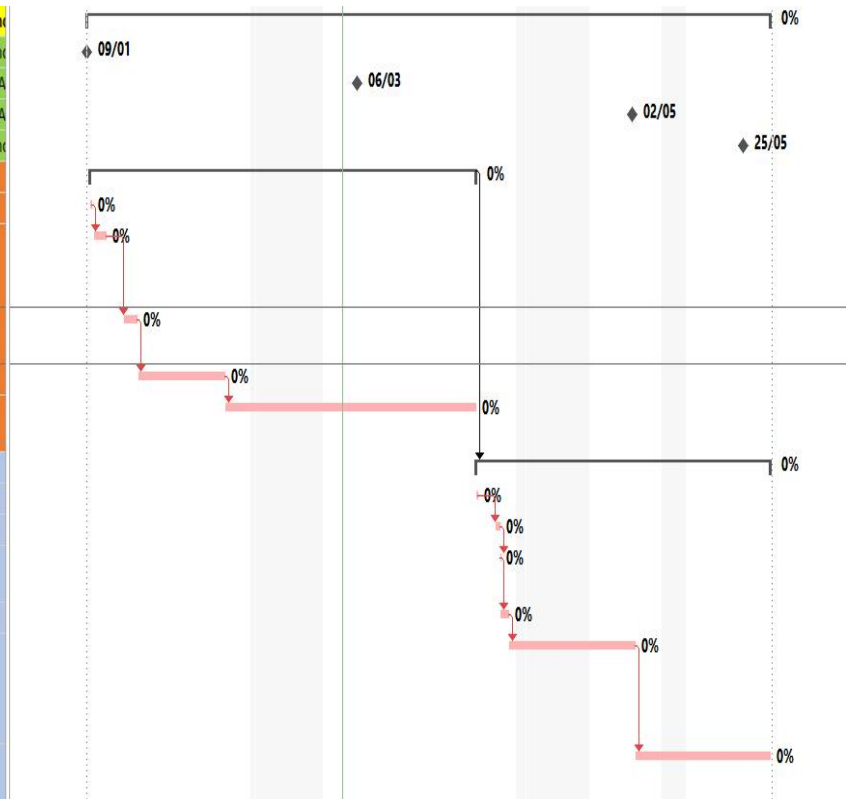
Voici le diagramme de séquence qui montre les frontières de mon intervention (cf annexes pages 108-109) :



IV) Planification prévisionnelle :

Pour gérer au mieux mon temps j'ai tout d'abord créé un **diagramme de GANTT (que l'on peut voir en dessous (cf annexe page 110))** pour pouvoir prévoir la durée de chaque tâche. Pour la fin du projet j'aimerais pouvoir générer un PDF selon des critères pré-définis et savoir comment crypter les données.

Projet 2SP	190 h	Lun 09/01/17	Mar 30/05/17	Anton Defrant
Début du projet	0 h	Lun 09/01/17	Lun 09/01/17	Anton Defrant
Revue 1	0 h	Lun 06/03/17	Lun 06/03/17	Cédric Célié
Revue 2	0 h	Mar 02/05/17	Mar 02/05/17	Cédric Célié
Remise du projet	0 h	Jeu 25/05/17	Jeu 25/05/17	Anton Defrant
Génération PDF	122 h	Lun 09/01/17	Jeu 30/03/17	Vincent Stin
Analyse du projet	3 h	Lun 09/01/17	Lun 09/01/17	Vincent Stin
Analyse et conception de la génération PDF	10 h	Mar 10/01/17	Jeu 12/01/17	7 Vincent Stin
Assimilation de la librairie FPDF	10 h	Lun 16/01/17	Jeu 19/01/17	8 Vincent Stin
Framework PHP Site	30 h	Jeu 19/01/17	Lun 06/02/17	9 Vincent Stin
Codage de la génération PDF	69 h	Lun 06/02/17	Jeu 30/03/17	10 Vincent Stin
Cryptage	68 h	Jeu 30/03/17	Mar 30/05/17	6 Vincent Stin
Manuel utilisateur	2 h	Jeu 30/03/17	Jeu 30/03/17	Vincent Stin
Cross Compilation	4 h	Lun 03/04/17	Mar 04/04/17	13 Vincent Stin
Dossier de développement	2 h	Mar 04/04/17	Mar 04/04/17	14 Vincent Stin
Guide installation	2 h	Mar 04/04/17	Jeu 06/04/17	15 Vincent Stin
Analyse et conception du chiffrement de la BDD	20 h	Jeu 06/04/17	Mar 02/05/17	16 Vincent Stin
Chiffrement de la BDD	38 h	Mar 02/05/17	Mar 30/05/17	17 Vincent Stin

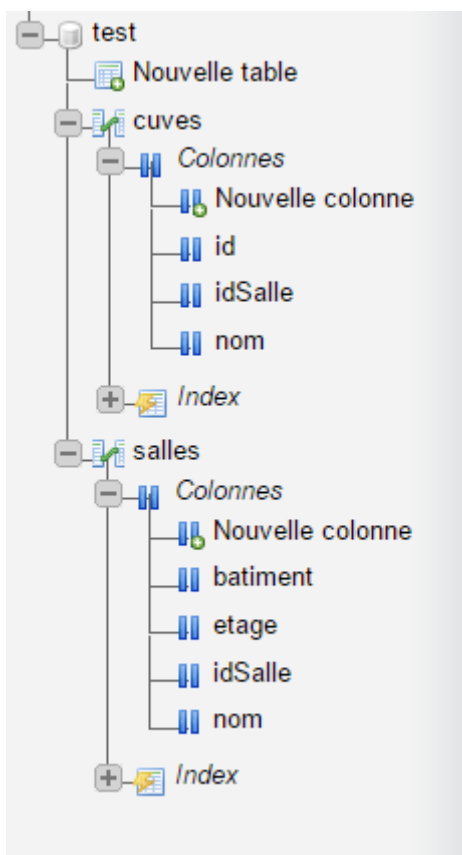


V) Travail effectué pour les revues 1 et 3:

Itération 1 : Prise en main de PHP sur mon PC.

J'ai utilisé un logiciel (WAMP) pour créer une base de donnée simple de type MySQL et administrée par PHP MyAdmin. Son rôle était de me permettre de m'approprier les différents outils logiciels sans impacter la BDD d'origine qui avait été mise en place l'an dernier par l'équipe dont on devait poursuivre le travail cette année .

(Structure de la BDD)



(Aspect de la connexion à la BDD. Pour s'y connecter, l'identifiant est "root" et le mot de passe est vide : "")



The screenshot shows the phpMyAdmin login interface. At the top, there is a logo with a sailboat and the text 'phpMyAdmin'. Below it, the text 'Bienvenue dans phpMyAdmin' is displayed. There are two main sections: 'Langue - Language' with a dropdown menu set to 'Français - French', and 'Connexion' with a login form. The login form has fields for 'Utilisateur :' (containing 'root') and 'Mot de passe :'. At the bottom right of the form is an 'Exécuter' button.

J'ai aussi créé un site internet simple de test disposant uniquement d'un bouton pour pouvoir générer un fichier PDF à travers un formulaire HTML et ceci, sans avoir à utiliser le site web réalisé l'an dernier qui mettait à disposition des fonctionnalités qui ne sont pas en rapport avec ce que je devais faire.

(Code du site)

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset ="urf-8" /> //Définit le format de
caractères en utf-8
```

```
<title>2SP</title> //Titre de l'onglet
```

```
</head>
```

```
<body>
```

```
<h1>Bienvenue sur le site du projet 2SP</h1> //Titre de la page
```

```
<div class="bouton"> // création d'un bouton
```

```
<form method="post" action="Receive.php"> //Le bouton
exécute le fichier "Receive.php"
```

Equipe projet 2SP




```

        <input type="submit" name="Submit" value="Générer le PDF">
//Définit le nom du bouton
    </form>

</div>

</body>

</html>

```

(résultat de la page)







Bienvenue sur le site du projet 2SP

Générer le PDF

Itération 2 : Accès aux données de la BDD depuis PDO.

Grâce au script PHP appelé sur appui sur ce bouton, j'ai réussi à récupérer les données de la BDD et les afficher sur une page internet. J'ai utilisé dans ce script l'extension PDO de PHP qui permet pratiquement de s'affranchir du type de la BDD sous jacente en offrant une interface uniforme.

(données dans la BDD)

				id	nom	idSalle
<input type="checkbox"/>		Modifier		Copier		Effacer
2	CryoTank02	2				
<input type="checkbox"/>		Modifier		Copier		Effacer
3	cryotank001	1				

(code du bouton)

```

<?php

    echo '<strong> Donnees sur les cuves : </strong>';
//Affichage d'une chaine de caractères

    echo '<br>'; //Retour à la ligne


    $bdd = new
PDO('mysql:host=localhost;dbname=test;charset=utf8', 'root', '');
//Connexion à la BDD

```

Equipe projet 2SP



```
$reponse = $bdd->query('SELECT * FROM cuves');
//Requête pour extraire les données de cuve

while ($donnees = $reponse->fetch()){ //Boucle pour
récupérer toutes les lignes de la table "cuves"

    echo '<br>'; //Retour à la ligne

    echo "<strong> Information sur les cuves : </strong>";
//Affichage d'une chaîne de caractère

    echo $donnees['nom']; //Affichage de la variable
"données"

}

?>
```

(résultat sur la page internet)

Donnees sur les cuves :

Information sur les cuves : CryoTank001

Information sur les cuves : CryoTank02

Itération 3 : Génération d'un PDF statique.

J'ai ensuite réussi à faire en sorte de créer un PDF sur une nouvelle page et d'y afficher du texte en utilisant la classe FPDF de PHP (cf annexes pages 99-103) qui, comme son nom l'évoque, permet de générer un PDF.

(code du bouton)

```
<?php

require('FPDF/fpdf.php'); //Inclus la classe FPDF

$pdf = new PDF('P', 'mm', 'A4'); //Instanciation de la
classe dérivée

$pdf->AddPage(); //Création d'une page vide

$pdf->SetFont('Arial','B',16); //Configuration
$pdf->Cell(40,10,'Donnees sur les cuves :'); //Chaîne de
caractères à un emplacement défini
```

Equipe projet 2SP



```
$pdf->Output(); //Génération de la page
```

```
?>
```

On inclut l'extension qui permet de générer du PDF, on instancie la classe FPDF de façon à pouvoir accéder aux méthodes qu'elles proposent, on configure le format de sortie, on crée une nouvelle page, on configure la police de caractères, on affiche un texte aux coordonnées (40,10) et on génère enfin le fichier PDF avec la méthode **Output()**. (cf annexe page 103)

(résultat)



Itération 4 : Génération d'un PDF avec des données issues d'une BDD.

J'ai finalement combiné les 2 codes pour pouvoir afficher les données de la BDD dans un PDF. Pour cela j'ai **spécialisé** la classe FPDF pour l'adapter à mes besoins

(code)

```
<?php
```

```
require('FPDF/fpdf.php'); //Inclus la classe FPDF
```

```
class PDF extends FPDF //héritage des méthodes de FPDF
```

```
{
```

```
    function __construct() { //Constructeur de la classe
dérivée
```



```
parent::__construct('P', 'mm', 'A4'); //Appel du
constructeur de la classe mère
}

/**
 * Méthode de chargement des données depuis la BDD
 */

function LoadData(){
    try {
        $releve = array();
        $bdd = new
PDO('sqlite:C:\Users\Vincent\Desktop\Mesdocuments\Cours\Projet\2SP.s
qlite'); //Connexion à la BDD

        $sql = 'SELECT * FROM Cuves'; //Requête pour
extraire les données de cuve
        $i = 0;
        $result = $bdd->query($sql); //Exécution de la
requête avec sauvegarde du result set dans le tableau $result
        foreach($result as $row) {
            $releve[ $i ] = $row;
            $i++;
        }
    }
    catch (Exception $exception) {
        echo '<p>Erreur de connexion à la BDD</p>';
    }
    return $releve;
}

}

$pdf = new PDF(); //Instanciación de la classe dérivée
$pdf->AddPage(); //Création d'une page vide
$pdf->SetFont('Arial','B',16); //Configuration
```



```
$data = $pdf->LoadData(); //Chargement des données

foreach($data as $line) { //Parcours et extraction des données pour
les insérer dans la page

$pdf->MultiCell(0, 10, '$line['nom'].': '$line
['idSalle'].': '$line['id']);

$pdf->Ln(5); //Saut de 5 lignes
}

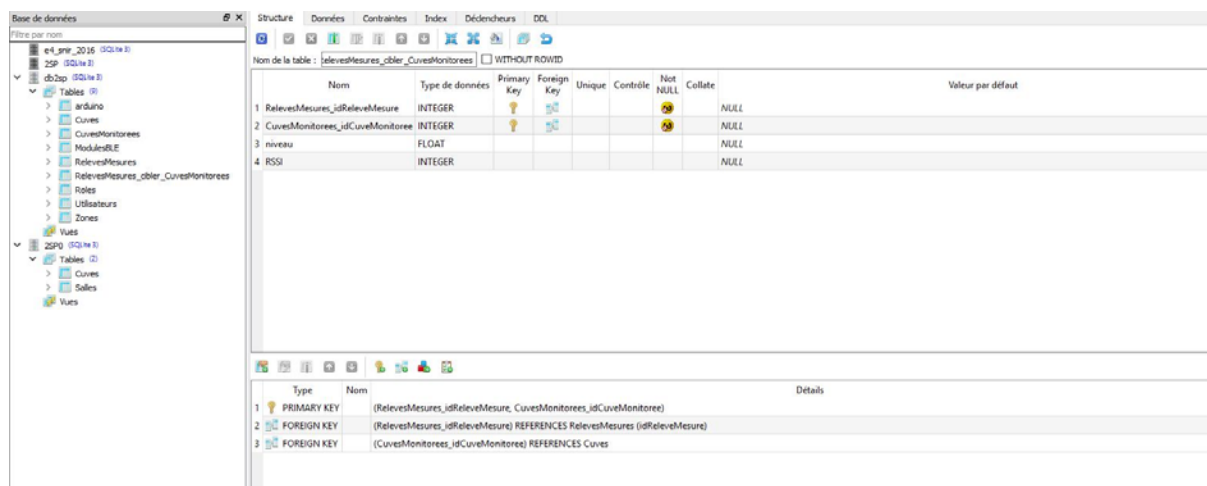
$pdf->Output(); //Génération de la page

?>
```

Itération 5 : Portage sur RaspBerry Pi.

Il fallait maintenant que je travaille avec la Raspberry et non plus en local sur mon PC j'ai donc du passer sur une BDD SQLITE car le client veut que le système soit facilement transportable. Or SQLite est non seulement beaucoup plus léger que MySQL donc plus adapté à une Raspberry Pi mais ses BDDs sont aussi plus simples car contenue intégralement dans leur propres fichiers. Je l'ai donc installée ainsi que le logiciel Lighttpd qui est le serveur web utilisé sur la Raspberry Pi à la place de Apache qui est lui aussi beaucoup plus lourd.

(Aspect de la BDD SQLITE (cf annexe page 111))



Nom	Type de données	Primary Key	Foreign Key	Unique	Contrôle	Not NULL	Collate	Valeur par défaut
1 RelevésMesures_idReleveMesure	INTEGER	Yes	No	No	No	Yes		NULL
2 CuvesMonitorées_idCuveMonitorée	INTEGER	No	Yes	No	No	Yes		NULL
3 niveau	FLOAT	No	No	No	No	Yes		NULL
4 RSSI	INTEGER	No	No	No	No	No		NULL

Type	Nom	Détails
1 PRIMARY KEY	(RelevésMesures_idReleveMesure, CuvesMonitorées_idCuveMonitorée)	
2 FOREIGN KEY	(RelevésMesures_idReleveMesure) REFERENCES RelevésMesures (idReleveMesure)	
3 FOREIGN KEY	(CuvesMonitorées_idCuveMonitorée) REFERENCES Cuves	

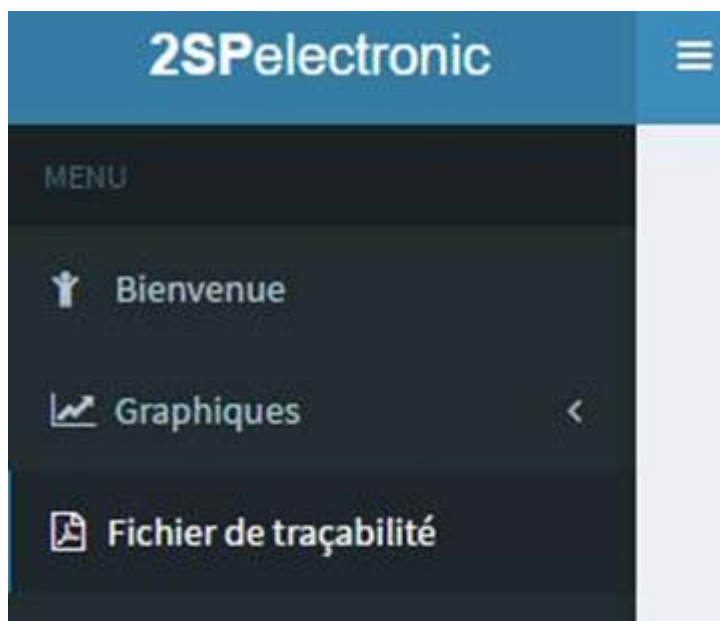


VI) Travail effectué pour la revue finale :

Itération 6 : Intégration au site existant.

La partie "traçabilité" du site permet à un utilisateur de pouvoir récupérer les données stockées dans la BDD qu'il désire.

(Aspect de la partie "traçabilité")



Une fois les installations finies, j'ai commencé par modifier la partie "traçabilité" (J'ai modifié le fichier " pdf.html.twig " qui se trouve dans le répertoire suivant " /home/pi/2sp_2017/public_html/dashboard/views ") du site pour l'adapter à mes besoins.

(code de la page)

```
<!DOCTYPE html>

<html>

  <head><!-- En-tête de la page (paramètres) -->
    <meta charset="utf-8" />
    <title>2SP</title> <!-- Nom de l'onglet -->
  </head>

  <body><!-- Corps de la page -->
    <h1>Bienvenue sur le site du projet 2SP !</h1>
    
```



```

<ul>
  <li>
    <div class="bouton">
      <form method="post" action="Receive.php" > // le
bouton exécute le fichier Receive.php
      <input type="submit" name="Submit" value="Date">
    </form>
    </div>
  </li>

  <p>Batiment :</p>
  <input type="text" list=browsers > <!-- Création d'une
ComboBox -->
  <datalist id=browsers >
    <option> 1
    <option> 2
    <option> 3
  </datalist>

  <p>Etage :</p>
  <input type="text" list=browsers > <!-- Création d'une
ComboBox -->
  <datalist id=browsers >
    <option> RDC
    <option> 1er etage
    <option> 2eme etage
  </datalist>

  <p>Cuves :</p>
  <input type="text" list=browsers > <!-- Création d'une
ComboBox -->
    <option> Cryotank 1

```



```

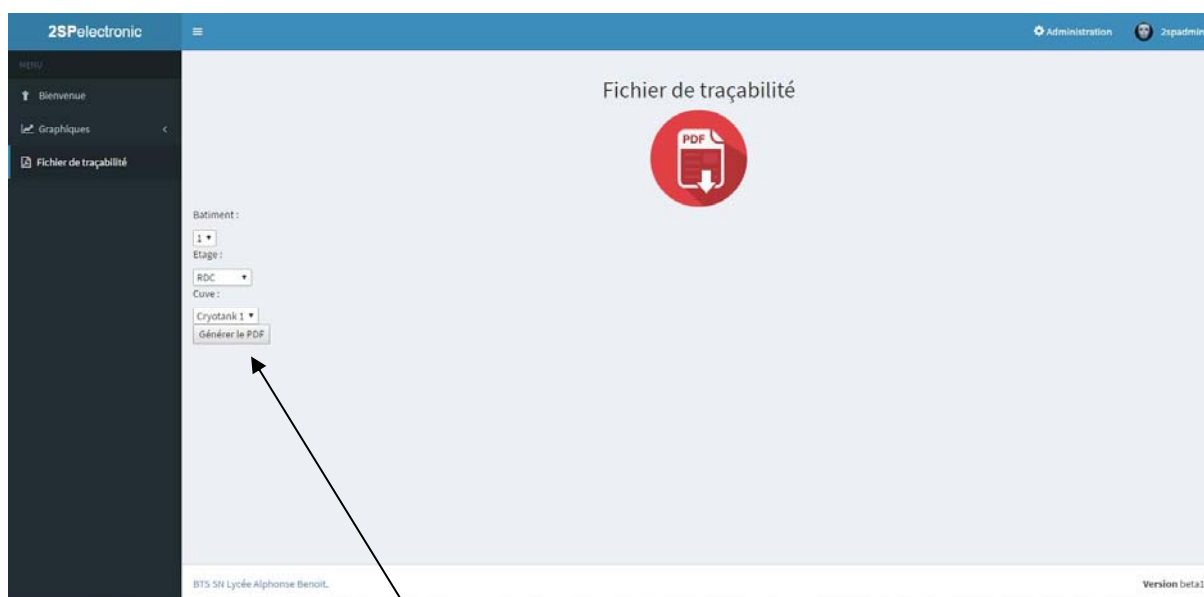
<option> Cryotank 2
<option> Cryotank 3
</datalist>

<li>
  <div class="bouton">
    <form method="post" action="Receive.php" >
      <input type="submit" name="Submit" value="Générer le
PDF">

    </form>
  </div>
</li>
</ul>
</body>
</html>
<?php

```

Résultat du code : (Seul les utilisateurs autorisés peuvent accéder à cette partie, ici je suis connecté en tant qu'administrateur, mais certains utilisateurs pourraient s'y connecter).



On peut voir le bouton "générer le PDF" et les futur ComboBox qui serviront plus tard à définir des paramètres de filtre.



(Screen d'une des ComboBox)

Batiment :

1 ▼

Etage :

RDC ▼

Cuve :

Cryotank 1 ▼

Cryotank 1

Cryotank 2

Cryotank 3

Cryotank 4

Cryotank 5

Cryotank 6

Cryotank 7

(Pour pouvoir ajouter mon code sur la RaspBerry, j'ai dû m'y connecter, pour cela, j'ai tapé la commande "ssh pi@192.168.2.23" avec le mot de passe "raspberrry". Pour pouvoir éteindre la RaspBerry en toute sécurité j'ai utilisé la commande "sudo halt -n").

J'ai ajouté et modifié mon code (J'ai ajouté le fichier " pdf.html.twig " qui se trouve au bout du chemin suivant " 2sp_2016/public_html/dashboard/web/lib/FPDF") pour qu'il puisse générer un PDF qui affiche les toutes données de la BDD.

(code)

<?php

require('fpdf.php'); //Inclus la classe FPDF

class PDF extends FPDF //héritage des méthodes de FPDF

{

function __construct() { //Constructeur de la classe dérivée

parent::__construct('P', 'mm', 'A4'); //Appel du constructeur de la classe mère

}

/**

* Méthode de chargement des données depuis la BDD

*/

Equipe projet 2SP



```
function LoadData(){
    try {
        $releve = array();
        $db = new SQLite3('../db/db2sp.db'); //
Connexion à la BDD
        $result = $db->query("SELECT * FROM
RelevésMesures_Cibler_CuvesMonitorees"); //Requête pour extraire
les données de cuve
        $i = 0;
        while($data = $result->fetchArray()) //Boucle pour
parcourir toute la BDD
        {
            $releve[$i] = $data; //Récupération des
données
            $i++;
        }
    }
    catch (Exception $exception) {
        echo '<p>Erreur de connexion à la BDD</p>';
    }
    return $releve; //Affichage de la variable releve
}

$pdf = new PDF(); //Instanciación de la clase derivada
$pdf->AddPage(); //Création d'une page vide
$pdf->SetFont('Arial','B',16); //Configuration
$pdf->Cell(40,10, 'Donnees des cuves'); //Chaîne de caractères à un
emplacement définit
$pdf->Ln(25); //Saut de 25 lignes
$data = $pdf->LoadData(); //Chargement des données
foreach($data as $line) {
    $pdf->MultiCell(0, 10, ''.$line[0].':'.$line[1].':'.$line[2]);
    //Affiche les données les unes après les autres
}
```



```
$pdf->Ln(5); //Saut de 5 lignes  
}  
$pdf->Output(); //Génération de la page  
?>
```

(résultat)

| | | |
|--|------------|--|
| | 1:1:15.5 | |
| | 2:1:20.27 | |
| | 3:1:15.61 | |
| | 4:1:17.32 | |
| | 5:1:11.9 | |
| | 6:1:15.02 | |
| | 7:1:15.72 | |
| | 8:1:20.13 | |
| | 9:1:9.56 | |
| | 10:1:18.91 | |
| | 11:1:13.42 | |

Cela affiche les différentes données les unes après les autres. J'ai ensuite modifié mon code pour qu'il puisse prendre en compte les ComboBox :

(code)

```
<?php  
require('fpdf.php'); //Inclus la classe FPDF  
  
class PDF extends FPDF //héritage des méthodes de FPDF  
{  
    function __construct() { //Constructeur de la classe dérivée  
        parent::__construct('P', 'mm', 'A4'); //Appel du  
        constructeur de la classe mère  
    }  
    /**  
    * Méthode de chargement des données depuis la BDD
```



```

*/

function LoadData(){
    try {
        $releve = array();

        $db = new SQLite3('../../db/db2sp.db'); //
Connexion à la BDD

        $result = $db->query("SELECT * FROM
RelevésMesures_Cibler_CuvesMonitorees WHERE
CuvesMonitorees_idCuveMonitoree = " . $_POST['cuve'] . ";");
//Requête pour extraire les données de cuve en prenant en compte la
valeur choisit de la ComboBox

        $i = 0;

        while($data = $result->fetchArray()) //Boucle pour
parcourir toute la BDD
        {
            $releve[$i] = $data; //Récupération des
données

            $i++;
        }
    }
    catch (Exception $exception) {
        echo '<p>Erreur de connexion à la BDD</p>';
    }
    return $releve; //Affichage de la variable releve
}

}

$pdf = new PDF(); //Instanciation de la classe dérivée
$pdf->AddPage(); //Création d'une page vide
$pdf->SetFont('Arial','B',16); //Configuration
$pdf->Cell(40,10, 'Donnees des cuves'); //Chaine de caractères à un
emplacement définit
$pdf->Ln(25); //Saut de 25 lignes
$data = $pdf->LoadData(); //Chargement des données
foreach($data as $line) {
    Equipe projet 2SP

```



```
$pdf->MultiCell(0, 10, ' '.$line[0].': '.$line[1].': '.$line[2]);  
//Affiche les donnes les unes après les autres  
$pdf->Ln(5); //Saut de 5 lignes  
}  
$pdf->Output(); //Génération de la page  
?>
```

Avec cette modification, il y a eu une sélection dans les données de la BDD et moins de données étaient affichées. Je n'ai pas réussi à aller plus loin et le code ne prend en compte qu'une des ComboBox.

Pour la partie cryptage, je n'ai pas pu la commencer mais j'aurais pu utiliser un logiciel comme SQLCipher pour crypter la BDD ou bien la crypter moi même en faisant par exemple un calcul entre la donnée et l'adresse MAC (adresse qui définit l'identité d'une carte électronique) pour que la donnée soit modifiée et que les personnes qui n'aient pas connaissance de ce calcul ne puissent pas retrouver la donnée ou bien utiliser un système de CheckSum qui est une information ajoutée au code envoyé pour que la personne qui reçoit le code puisse vérifier que l'information envoyée est bien celle qui a été reçue.

VII) Partage des informations dans le groupe :

Pour communiquer ou partager des informations/fichiers avec le groupe, nous avons créé un compte sur un service web d'hébergement de fichiers avec gestion de versions (BitBucket) basé sur le gestionnaire de version Git. Il nous a permis de garder une trace des différentes versions de nos codes. Nous avons aussi utilisé un drive pour pouvoir stocker des documents et nous nous sommes aussi servi de Trello qui est un outil de gestion de projet. Ce dernier nous a servi à suivre l'avancement de chacun tout en définissant des dates limites d'exécution pour nous permettre de gérer notre temps.

Nous avons effectué des compte rendus suite à des réunions pour savoir l'avancement de chacun et nous permettre de nous fixer des objectifs jusqu'à la suivante.



(exemple de compte rendu de réunion)

| 5. Compte-rendu d'activité. | | |
|---|-----------------------------------|---|
| Date | Participants | Activité |
| 24/04/2017 | STIN, De Francesco, Crotte, Célié | Rappel des objectifs de chacun pour la revue 2.

Réunion pour la mise au point de la partie commune du dossier Revue 2.
Suite à la demande du client (professeur). |
| <p>Rappel des objectifs Revue 2 :</p> <p>Anton : Présenter un site web avec des graphiques par zone et un résumé des cuves par zone</p> <p>Vincent : Améliorer le fichier de traçabilité PDF.</p> <p>Cédric : Dynamiser le site Web avec PUG.</p> <p>Johann : Présenter la carte finie.</p> <p>Mise au point Partie commune revue 2 :</p> <ol style="list-style-type: none"> 1) Présentation de l'entreprise 2SP 2) Le Projet (but, cible <u>clientelle</u>, <u>interet</u> par rapport à l'existant) 3) Présentation du matériel (cuves, cartes, raspberry, Convertisseur, RTC, RN4020 ...) 4) Présentation des Software <ul style="list-style-type: none"> • Le site web (Onglet, utilisateurs) • La base de données • La fiche de traçabilité • Application web d'aide au placement 5) Les codes utilisés (leurs intérêts...) <p>Nous nous sommes mis d'accord pour ce plan de la partie commune du dossier pour la seconde revue.</p> <p>Anton</p> <p>Vincent</p> <p>Cédric</p> <p>Johann</p> | | |



VIII) Conclusion :

J'ai réussi à afficher les données de la BDD sur un PDF en local et sur la Raspberry Pi sous différentes BDD (MySQL et SQLite) et j'ai un début fonctionnel de filtre avec des ComboBox qui sert à ne récupérer que les données choisies.

J'ai plusieurs solutions pour pouvoir crypter les données mais je n'ai pas pu m'y employer.

J'aurais pu gagner du temps si j'avais commencé à coder directement sur la Raspberry Pi et non sur mon PC mais cela j'en avais besoin pour mieux comprendre sur un système plus basique.

J'aurais pu gagner du temps si j'avais commencer à coder directement sur la RaspBerry et non en local mais j'en avait besoin pour mieux comprendre sur un système plus basique.

Voici un récapitulatif de ce que j'ai fait :

Itération 1 : Prise en main de PHP sur mon PC

Itération 2 : Accès aux données de la BDD depuis PDO

Itération 3 : Génération d'un PDF statique

Itération 4 : Génération d'un PDF avec des données issues d'une BDD

Itération 5 : Portage sur RaspBerry Pi

Itération 6 : Intégration au site existant

Grâce à ce projet, j'ai pu mieux comprendre comment se passe le travail en équipe et en entreprise. J'ai du être autonome et ne pas baisser les bras aux problèmes qui se présentaient à moi même si je ne les avaient jamais vu.



IX) Annexes :

1) Journal de bord :

Voici mon journal de bord qui retrace ce que j'ai accompli jusqu'à présent :

| Date | Intervenant | Objet |
|------------|-------------|--|
| 09/01/2017 | IR2 | <p><u>Assimilation du projet :</u></p> <ul style="list-style-type: none"> - Lecture du projet. - Discussion sur le projet avec les autres membres du groupe. <p><u>Recherche :</u></p> <ul style="list-style-type: none"> - Recherche sur les moyens de cryptage - Recherche sur la génération de PDF via PHP. <p><u>Création :</u></p> <ul style="list-style-type: none"> - Création du drive. - Création d'un gestionnaire de projet (Trello). |
| 10/01/2017 | IR2 | <p><u>Installation :</u></p> <ul style="list-style-type: none"> - Installation d'un Bash sous Windows pour faciliter l'utilisation de Git. <p><u>Etude de Git :</u></p> <ul style="list-style-type: none"> - Assimilation du fonctionnement de Git. |
| 12/01/2017 | IR2 | <p><u>Installation :</u></p> <ul style="list-style-type: none"> - Installation de SourceTree. <p><u>Recherche :</u></p> <ul style="list-style-type: none"> - Recherches sur le chiffrement de la BDD. |
| 16/01/2017 | IR2 | <p><u>Recherche :</u></p> <ul style="list-style-type: none"> - Recherches sur la génération de PDF. |
| 17/01/2017 | IR2 | <p><u>Création :</u></p> <ul style="list-style-type: none"> - Création des fichiers source Word pour le dossier et les revues de projet. |



| | | |
|------------|-----|--|
| | | <u>Etude :</u>
- Etude de faisabilité de la génération de PDF. |
| 19/01/2017 | IR2 | <u>Apprentissage :</u>
- Apprentissage de L'HTML sur OpenClassroom. |
| 23/01/2017 | IR2 | <u>Apprentissage :</u>
- Apprentissage de L'HTML sur OpenClassroom. |
| 24/01/2017 | IR2 | <u>Réunion :</u>
- Réunion avec le professeur responsable du projet.
- Réunion avec les membres du groupe. |
| 26/01/2017 | IR2 | <u>Aide :</u>
- Aide de l'IR1 pour la mise en place du site.

<u>Etude :</u>
- Etude du code source des étudiants de l'année passée pour la génération de PDF. |
| 30/01/2017 | IR2 | <u>Installation :</u>
- Installation du site sur la RaspBerry. |
| 31/01/2017 | IR2 | <u>Installation :</u>
- Installation du site sur la RaspBerry. |
| 02/02/2017 | IR2 | <u>Apprentissage :</u>
- Apprentissage du PHP sur OpenClassroom. |
| 06/02/2017 | IR2 | <u>Réunion :</u>
- Réunion avec les membres du groupe.

<u>Recherche :</u>
- Recherche pour mettre en forme ma tâche. |
| 07/02/2017 | IR2 | <u>Codage :</u>
- Codage d'un site pour faire des tests. |
| 09/02/2017 | IR2 | <u>Création :</u>
- Création d'une BDD test sous PHP MyAdmin avec WAMP.

<u>Recherche :</u> |



| | | |
|------------|-----|--|
| | | - Recherche pour faire apparaître les informations de la BDD sur une nouvelle page. |
| 27/02/2017 | IR2 | <u>Codage :</u>
- Codage pour faire afficher les informations de la BDD sur une nouvelle page, problème rencontré lors de l'affichage des données. |
| 28/02/2017 | IR2 | <u>Codage :</u>
- Codage pour faire afficher les informations de la BDD sur une nouvelle page.

<u>Recherche :</u>
- Recherche pour générer un PDF. |
| 02/03/2017 | IR2 | <u>Codage :</u>
- Codage pour la génération de PDF réussi, je peux afficher du texte mais je n'arrive pas encore à afficher les informations de la BDD.

<u>Recherche :</u>
- Recherche pour faire afficher des informations de la BDD sur le PDF. |
| 06/03/2017 | IR2 | <u>Recherche :</u>
- Recherche pour faire afficher des informations de la BDD sur le PDF. |
| 07/03/2017 | IR2 | <u>Recherche :</u>
- Recherche pour faire afficher des informations de la BDD sur le PDF. |
| 09/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les informations de la BDD sur le PDF. |



| | | |
|------------|-----|--|
| 13/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les informations de la BDD sur le PDF (résoudre les problèmes). |
| 14/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les informations de la BDD sur le PDF (résoudre les problèmes). |
| 16/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les informations de la BDD sur le PDF (résoudre les problèmes). |
| 20/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les informations de la BDD sur le PDF (résoudre les problèmes). |
| 21/03/2017 | IR2 | <u>Réunion :</u>
- Réunion avec le professeur et définition des objectifs de la semaine (Installer SQL Lite pour pouvoir travailler sur la BDD de la <u>RaspBerry</u> en local et modifier le code pour s'adapter à la nouvelle BDD).

<u>Installation :</u>
- Installation de SQ Lite.

<u>Recherche :</u>
- Recherche pour pouvoir adapter le code pour le BDD SQ Lite. |



| | | |
|------------|-----|--|
| 23/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les données de la BDD sur le PDF et modifications du au changement de BDD. |
| 27/03/2017 | IR2 | <u>Codage :</u>
- Codage pour afficher les données de la BDD sur le PDF et modifications du au changement de BDD. |
| 28/03/2017 | IR2 | <u>Réunion :</u>
- Réunion avec le professeur et définition des objectifs de la semaine (Générer un PDF avec les données de la BDD).

<u>Codage :</u>
- Fin du code pour pouvoir afficher les données de la BDD sur un fichier PD sous SQ Lite. |
| 30/03/2017 | IR2 | <u>Réflexion :</u>
- Réflexion pour migrer mon code sur la <u>RaspBerry</u> .

<u>Codage :</u>
- Modification du code pour l'adapter à la BDD de la <u>RaspBerry</u> et modification du code car BDD différente pour pouvoir migrer mon code dessus. |
| 01/04/2017 | IR2 | <u>Codage :</u>
- Modification du code pour l'adapter à la BDD de la <u>RaspBerry</u> et modification du code car BDD différente pour pouvoir migrer mon code dessus. |
| 04/04/2017 | IR2 | <u>Réunion :</u>
- Réunion avec le professeur et définition des objectifs de la semaine.

<u>Portabilité :</u>
- Migration sur <u>Raspberry</u> réussi. |



| | | |
|------------|-----|---|
| | | <u>Recherche :</u>
- Recherche pour pouvoir ajouter des ComboBox sur la page internet. |
| 24/04/2017 | IR2 | <u>Codage :</u>
- ComboBox ajoutés sur la page du site web.

<u>Recherche :</u>
- Recherche pour exploiter les ComboBox selon leur état. |
| 25/04/2017 | IR2 | <u>Codage :</u>
- Codage pour l'exploitation des ComboBox . |
| 27/04/2017 | IR2 | <u>Codage :</u>
- Début de code concluant pour l'exploitation des ComboBox , je peux choisir d'afficher certaines données de la BDD selon l'état des ComboBox .

<u>Recherche :</u>
- Recherche pour crypter les données. |

| | | |
|------------|-----|--|
| 02/05/2017 | IR2 | <u>Codage :</u>
- Essaie de faire fonctionner les ComboBox . |
|------------|-----|--|



| | | |
|------------|-----|--|
| 04/05/2017 | IR2 | <u>Codage :</u>
- Essaie de faire fonctionner les ComboBox. |
| 09/05/2017 | IR2 | <u>Codage :</u>
- Essaie de faire fonctionner les ComboBox. |
| 15/05/2017 | IR2 | <u>Recherche :</u>
- Recherche pour crypter les données. |
| 16/05/2017 | IR2 | <u>Recherche :</u>
- Recherche pour crypter les données. |
| 18/05/2017 | IR2 | <u>Préparation :</u>
- Préparation du dossier. |
| 22/05/2017 | IR2 | <u>Préparation :</u>
- Préparation du dossier. |



2) Documentation des fonctions FPDF utilisées :

\$pdf = new FPDF('P', 'mm', 'A4');

Constructeur qui définit une page en A4 avec pour unité de mesures le millimètre.

\$pdf->AddPage();

<http://www.fpdf.org/fr/doc/addpage.htm>

AddPage

AddPage([**string** orientation [, **mixed** size [, **int** rotation]]])

Description

Ajoute une nouvelle page au document. Si une page était en cours, la méthode Footer() est appelée pour traiter le pied de page. Puis la page est ajoutée, la position courante mise en haut à gauche en fonction des marges gauche et haute, et Header() est appelée pour afficher l'en-tête.

La police qui était en cours au moment de l'appel est automatiquement restaurée. Il n'est donc pas nécessaire d'appeler à nouveau SetFont() si vous souhaitez continuer avec la même police. Même chose pour les couleurs et l'épaisseur du trait.

L'origine du système de coordonnées est en haut à gauche et les ordonnées croissantes vont vers le bas.

Paramètres

orientation

Orientation de la page. Les valeurs possibles sont (indépendamment de la casse) :

- P ou Portrait
- L ou Landscape

La valeur par défaut est celle indiquée dans le constructeur.

size

Format de la page. Il peut s'agir d'une des valeurs ci-dessous (indépendamment de la casse) :

- A3
- A4
- A5
- Letter
- Legal



ou bien d'un tableau contenant la largeur et la hauteur (exprimées en unité utilisateur).

La valeur par défaut est celle indiquée dans le constructeur.

rotation

Angle de rotation de la page. La valeur doit être un multiple de 90 ; la rotation se fait dans le sens horaire. La valeur par défaut est 0.

\$pdf->SetFont('Arial','B',16);

<http://www.fpdf.org/fr/doc/setfont.htm>

SetFont

SetFont(**string** family [, **string** style [, **float** size]])

Description

Fixe la police utilisée pour imprimer les chaînes de caractères. Il est obligatoire d'appeler cette méthode au moins une fois avant d'imprimer du texte, sinon le document résultant ne sera pas valide.

La police peut être soit une police standard, soit une police ajoutée à l'aide de la méthode AddFont(). Les polices standard utilisent l'encodage Windows cp1252 (Europe de l'ouest). La méthode peut être appelée avant que la première page ne soit créée et la police est conservée de page en page.

Si vous souhaitez juste changer la taille courante, il est plus simple d'appeler SetFontSize().

Note : il est nécessaire que les fichiers de définition soient accessibles. Il sont recherchés successivement dans :

- Le répertoire défini par la constante FPDF_FONTPATH (si cette constante est définie)
- Le répertoire font situé dans le répertoire de fpdf.php (s'il existe)
- Les répertoires accessibles par include()

Si le fichier correspondant à la police demandée n'est pas trouvé, l'erreur "Could not include font definition file" est générée.

Paramètres

family

Famille de la police. Il peut s'agir d'un nom défini par AddFont() ou bien d'une des familles standard :

- Courier (caractères de largeur fixe)
- Helvetica ou Arial (synonymes; sans serif)
- Times (avec serif)



- Symbol (symboles)
- ZapfDingbats (symboles)

Le nom n'est pas sensible à la casse.

Il est également possible de passer une chaîne vide, auquel cas la famille courante est conservée.

style

Style de la police. Les valeurs possibles sont (indépendamment de la casse) :

- chaîne vide : normal
- B : gras
- I : italique
- U : souligné

ou une combinaison quelconque. La valeur par défaut est le style normal. Les styles gras et italique ne s'appliquent pas aux familles Symbol et ZapfDingbats.

size

Taille de la police en points.

La valeur par défaut est la taille courante. Si aucune taille n'a encore été spécifiée depuis le début du document, la valeur prise est 12.

\$pdf->Cell(40,10,'Donnees sur les cuves :');

<http://www.fpdf.org/fr/doc/cell.htm>

Cell

Cell(**float** w [, **float** h [, **string** txt [, **mixed** border [, **int** ln [, **string** align [, **boolean** fill [, **mixed** link]]]]]])

Description

Imprime une cellule (zone rectangulaire) avec éventuellement des bords, un fond et une chaîne de caractères. Le coin supérieur gauche de la cellule correspond à la position courante. Le texte peut être aligné ou centré. Après l'appel, la position courante se déplace à droite ou un retour à la ligne est effectué. Il est possible de mettre un lien sur le texte.

Si le saut de page automatique est activé et que la cellule dépasse le seuil de déclenchement, un saut de page est effectué avant de l'imprimer.

Paramètres

w

Largeur de la cellule. Si elle vaut 0, la cellule s'étend jusqu'à la marge droite de la page.



h

Hauteur de la cellule. Valeur par défaut : 0.

txt

Chaîne à imprimer. Valeur par défaut : chaîne vide.

border

Indique si des bords doivent être tracés autour de la cellule. La valeur peut être soit un nombre :

- 0 : aucun bord
- 1 : cadre

soit une chaîne contenant certains ou tous les caractères suivants (dans un ordre quelconque) :

- L : gauche
- T : haut
- R : droit
- B : bas

La valeur par défaut est 0.

ln

Indique où se déplace la position courante après l'appel à la méthode. Les valeurs possibles sont :

- 0 : à droite
- 1 : au début de la ligne suivante
- 2 : en dessous

Mettre 1 est équivalent à mettre 0 et appeler la méthode Ln() juste après. La valeur par défaut est 0.

align

Permet de centrer ou d'aligner le texte. Les valeurs possibles sont :

- L ou chaîne vide : alignement à gauche (valeur par défaut)
- C : centrage
- R : alignement à droite

fill

Indique si le fond de la cellule doit être coloré (true) ou transparent (false). Valeur par défaut : false.

link

URL ou identifiant retourné par AddLink().

\$pdf->Output();

<http://www.fpdf.org/fr/doc/output.htm>

Output

string Output([**string** dest [, **string** name [, **boolean** isUTF8]])

Description

Envoie le document vers une destination donnée : navigateur, fichier ou chaîne de caractères. Dans le cas du navigateur, on peut utiliser le visualiseur PDF ou bien forcer le téléchargement.

La méthode commence par appeler Close() si nécessaire pour terminer le document.

Paramètres

dest

Destination où envoyer le document. Le paramètre peut prendre les valeurs suivantes :

- I : envoyer en inline au navigateur. Le visualiseur PDF est utilisé.
- D : envoyer au navigateur en forçant le téléchargement, avec le nom indiqué dans name.
- F : sauver dans un fichier local, avec le nom indiqué dans name (peut inclure un répertoire).
- S : renvoyer le document sous forme de chaîne.

La valeur par défaut est I.

name

Le nom du fichier. Il est ignoré dans le cas de la destination S.

La valeur par défaut est doc.pdf.

isUTF8

Indique si name est encodé en ISO-8859-1 (false) ou en UTF-8 (true).

Ce paramètre est utilisé uniquement pour les destinations I et D.

La valeur par défaut est false.





Curve



Côté serveur



Récupère les informations des cuves et les ajoute dans la BDD

Récupérer le dernier enregistrement de la BDD et convertit les données dans un tableau



Base de données

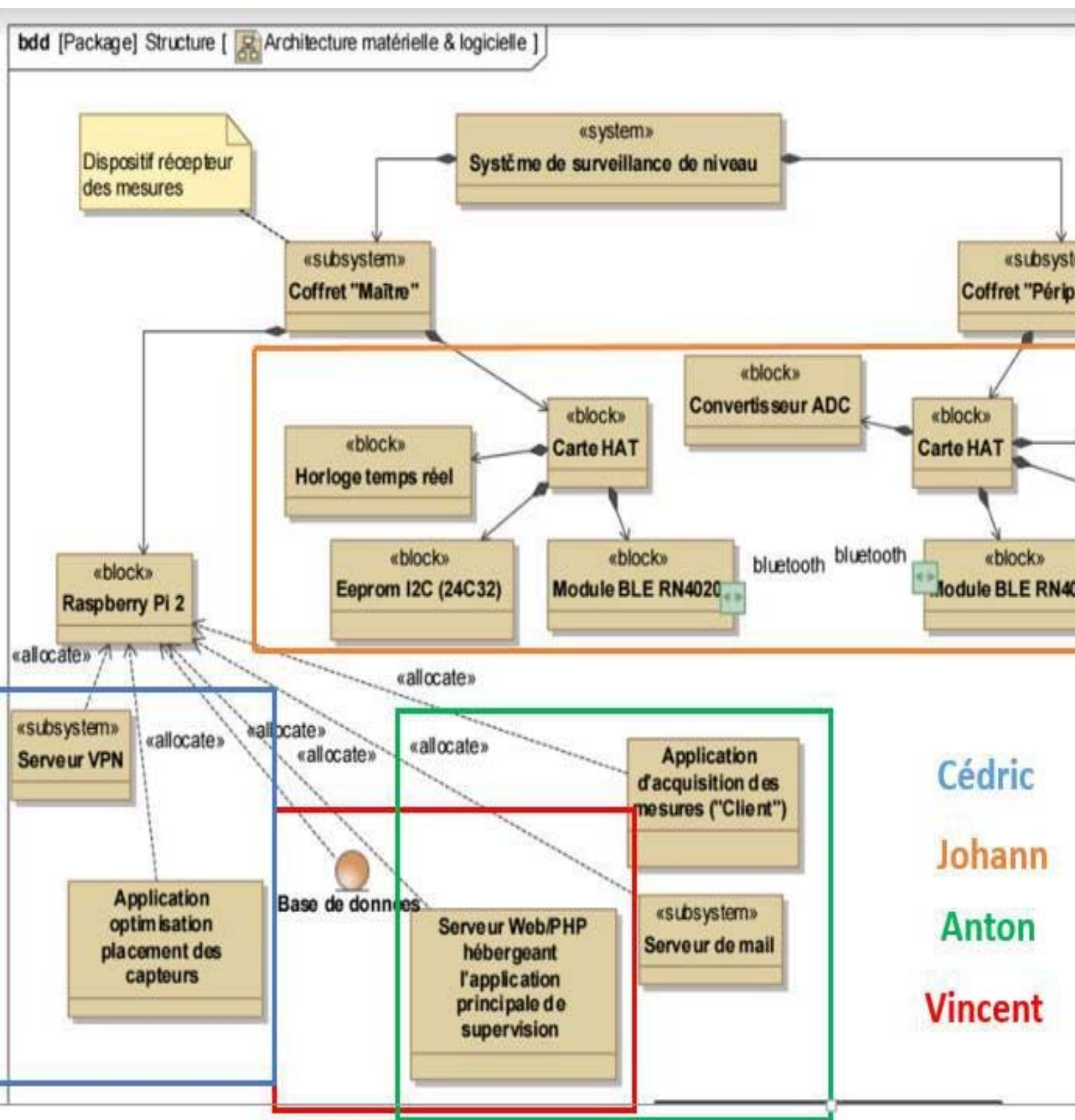


Côté client



Appel le script PHP toute les 10 secondes et créer un point si les informations ont changées





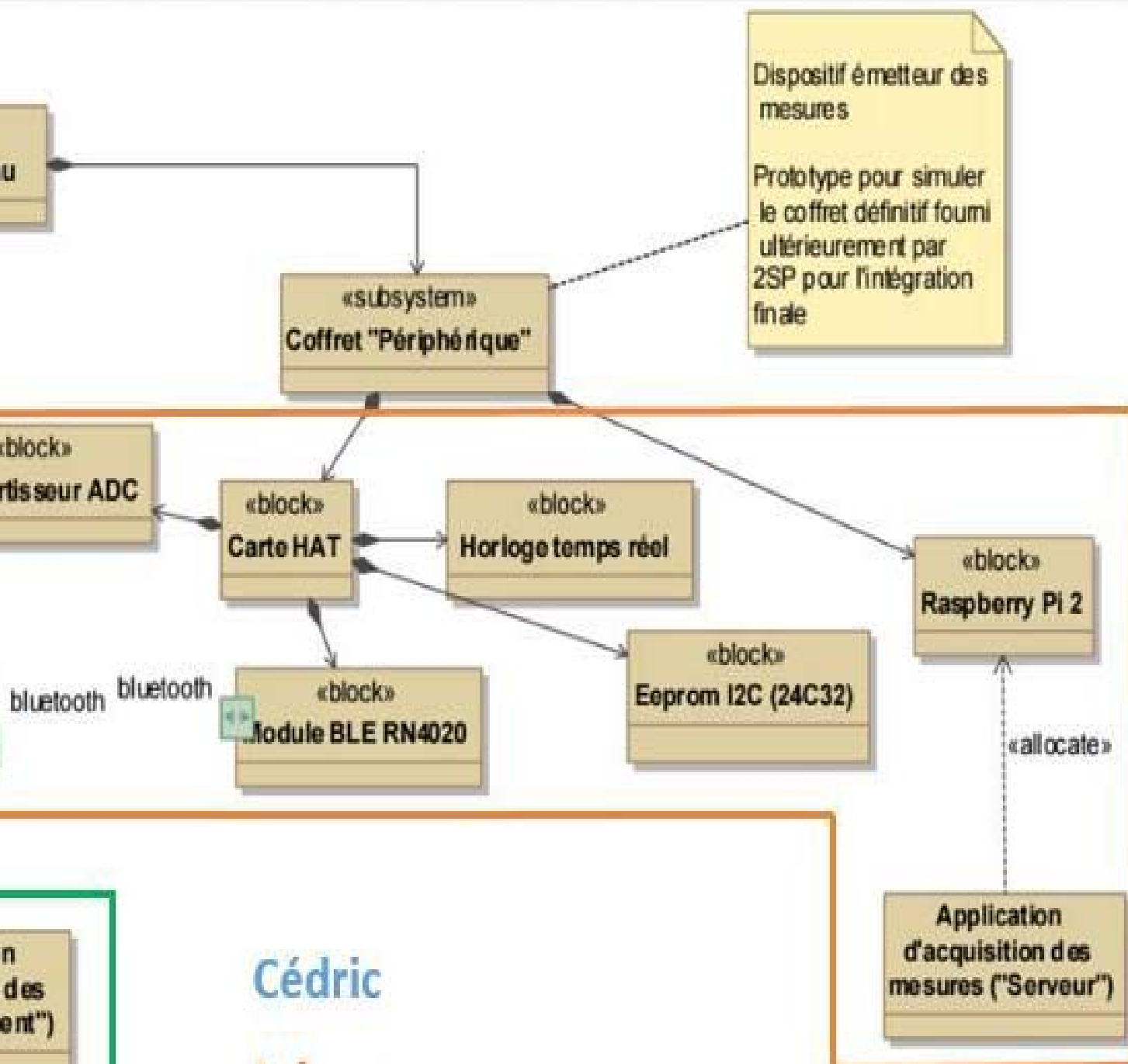
Cédric

Johann

Anton

Vincent



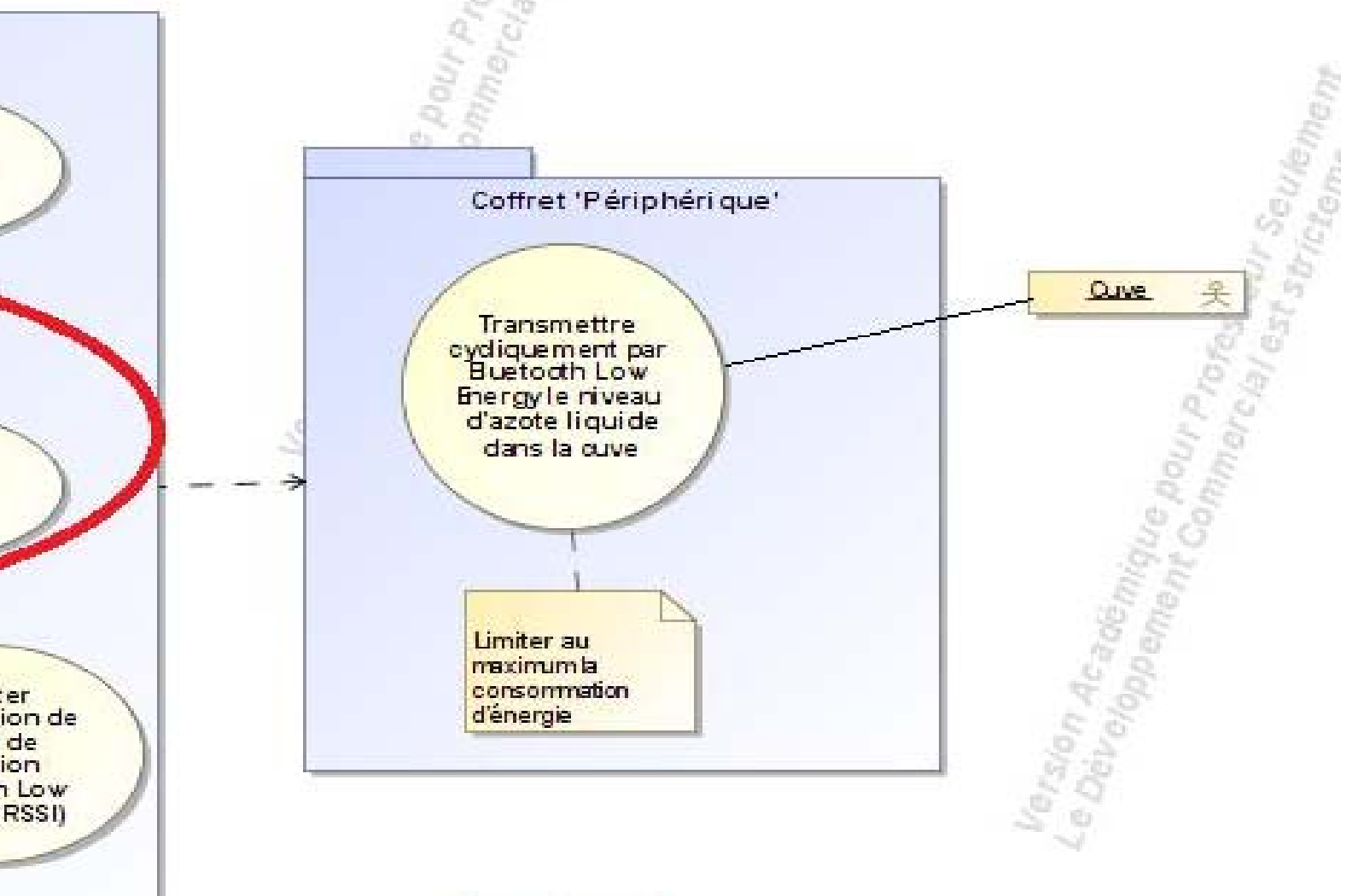
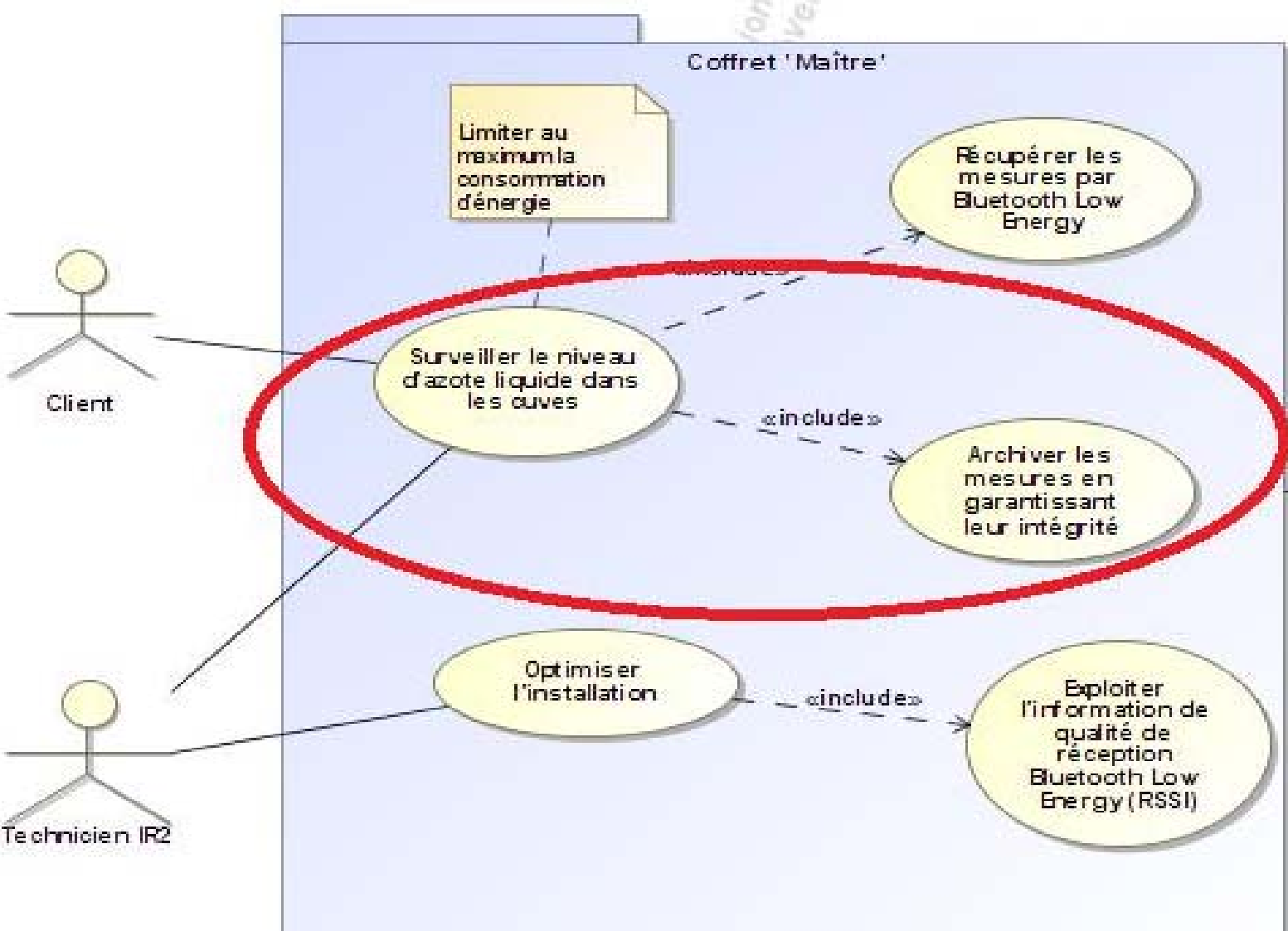


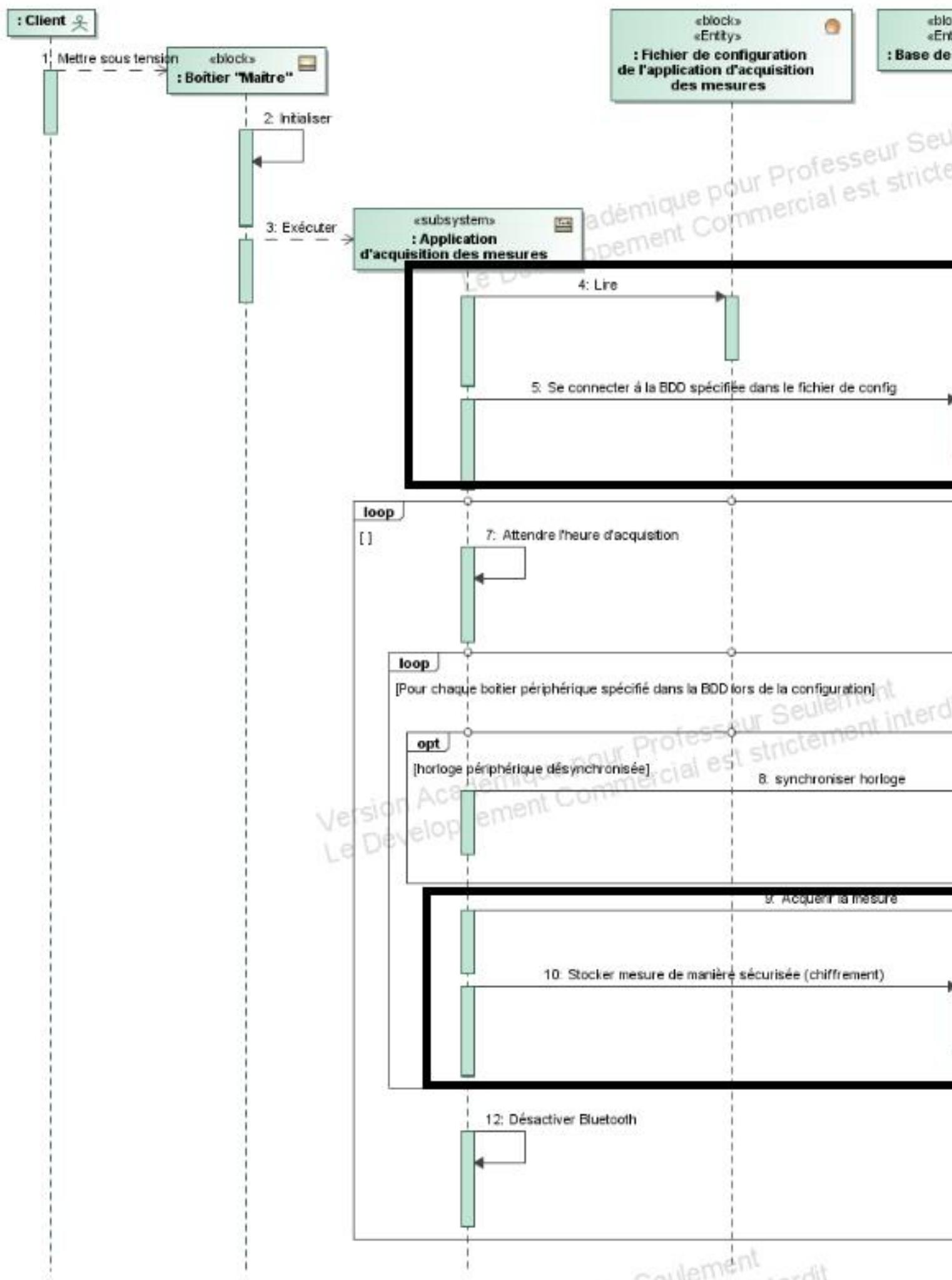
Cédric

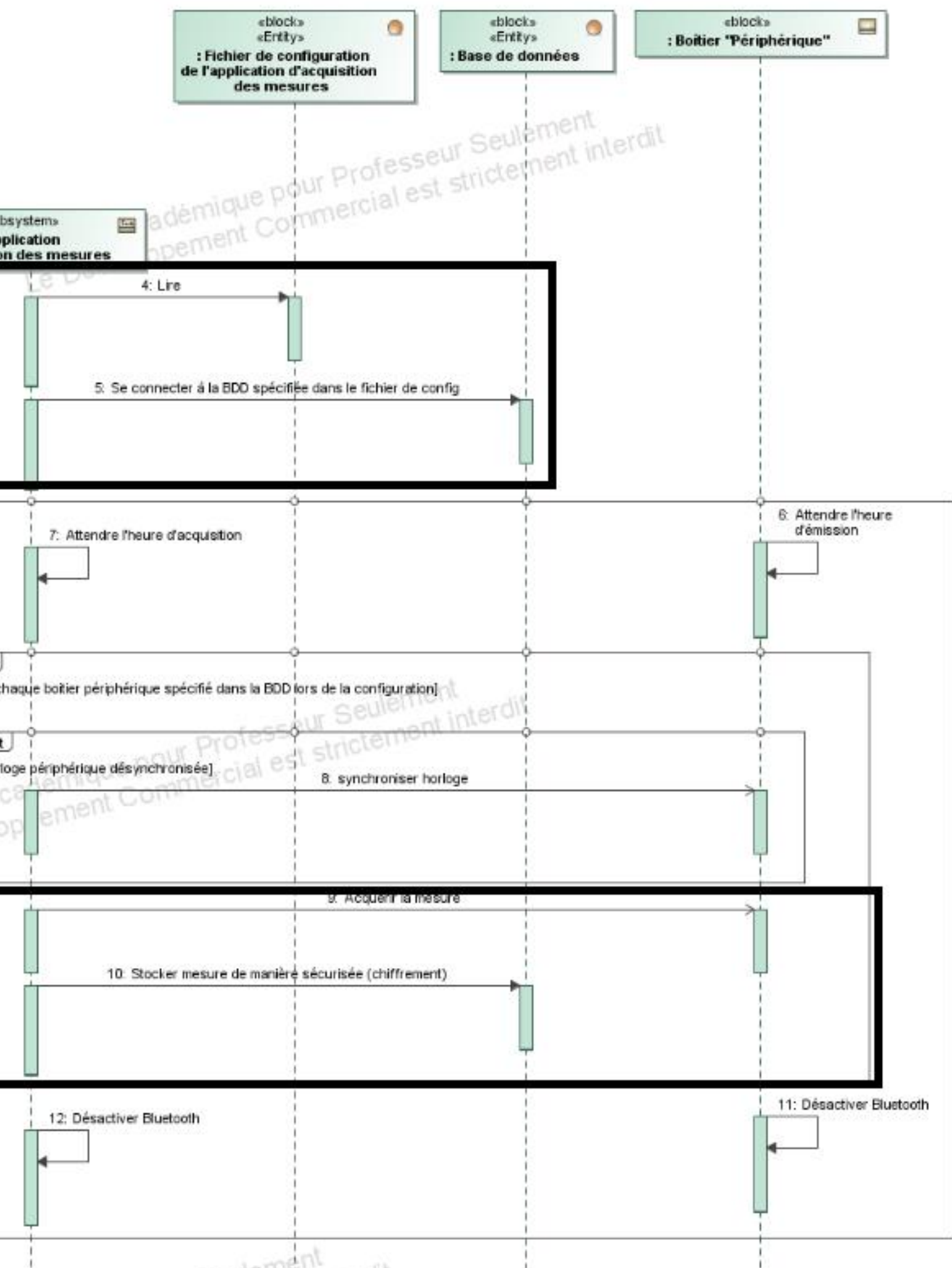
Johann

Anton

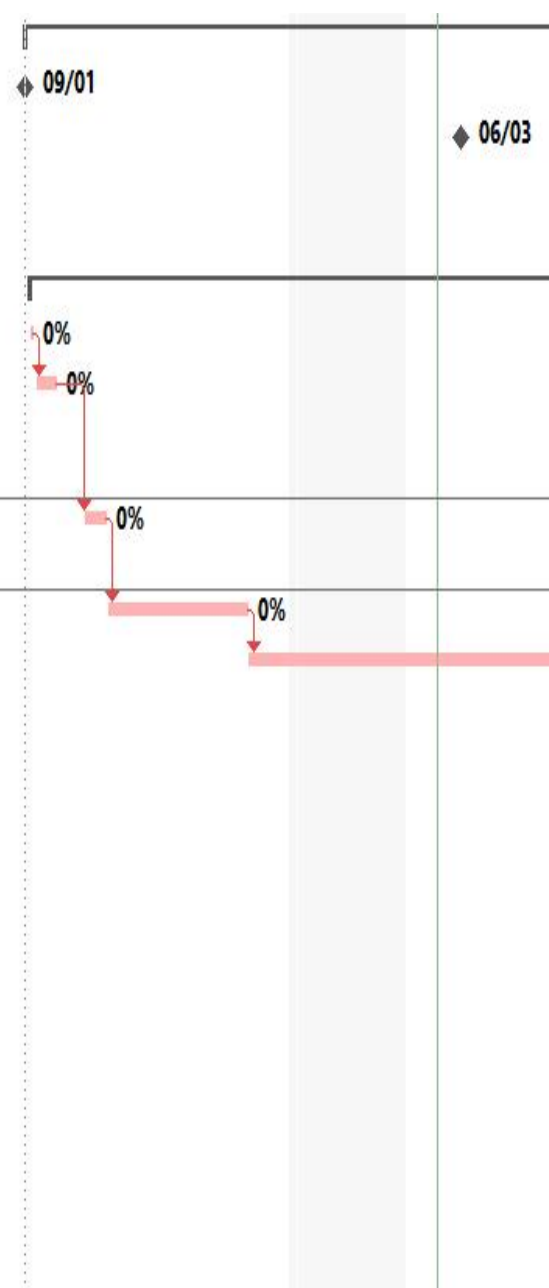
Vincent







| | | | | | |
|---|--|-------|--------------|--------------|-----------------|
| 4 | Projet 2SP | 190 h | Lun 09/01/17 | Mar 30/05/17 | Anton Defranc |
| | Début du projet | 0 h | Lun 09/01/17 | Lun 09/01/17 | Anton Defranc |
| | Revue 1 | 0 h | Lun 06/03/17 | Lun 06/03/17 | Cédric Célié;A |
| | Revue 2 | 0 h | Mar 02/05/17 | Mar 02/05/17 | Cédric Célié;A |
| | Remise du projet | 0 h | Jeu 25/05/17 | Jeu 25/05/17 | Anton Defranc |
| 4 | Génération PDF | 122 h | Lun 09/01/17 | Jeu 30/03/17 | Vincent Stin |
| | Analyse du projet | 3 h | Lun 09/01/17 | Lun 09/01/17 | Vincent Stin |
| | Analyse et conception de la génération PDF | 10 h | Mar 10/01/17 | Jeu 12/01/17 | 7 Vincent Stin |
| | Assimilation de la librairie FPDF | 10 h | Lun 16/01/17 | Jeu 19/01/17 | 8 Vincent Stin |
| | Framework PHP Sile | 30 h | Jeu 19/01/17 | Lun 06/02/17 | 9 Vincent Stin |
| | Codage de la génération PDF | 69 h | Lun 06/02/17 | Jeu 30/03/17 | 10 Vincent Stin |
| 4 | Cryptage | 68 h | Jeu 30/03/17 | Mar 30/05/17 | 6 Vincent Stin |
| | Manuel utilisateur | 2 h | Jeu 30/03/17 | Jeu 30/03/17 | Vincent Stin |
| | Cross Compilation | 4 h | Lun 03/04/17 | Mar 04/04/17 | 13 Vincent Stin |
| | Dossier de développement | 2 h | Mar 04/04/17 | Mar 04/04/17 | 14 Vincent Stin |
| | Guide installation | 2 h | Mar 04/04/17 | Jeu 06/04/17 | 15 |
| | Analyse et conception du chiffrement de la BDD | 20 h | Jeu 06/04/17 | Mar 02/05/17 | 16 Vincent Stin |
| | Chiffrement de la BDD | 38 h | Mar 02/05/17 | Mar 30/05/17 | 17 Vincent Stin |



Surveillance

Base de données





Filtre par nom

- e4_snir_2016 (SQLite 3)
- 2SP (SQLite 3)
- db2sp (SQLite 3)
 - Tables (9)
 - arduino
 - Cuves
 - CuvesMonitorees
 - ModulesBLE
 - RelevsMesures
 - RelevsMesures_cibler_CuvesMonitorees
 - Roles
 - Utilisateurs
 - Zones
 - Vues
- 2SP0 (SQLite 3)
 - Tables (2)
 - Cuves
 - Salles
 - Vues




Structure Données Contraintes Index Déclencheurs DDL

Nom de la table : RelevsMesures_cibler_CuvesMonitorees

☐ WITHOUT ROWID

| | Nom | Type de données | Primary Key | Foreign Key | Unique | Contrôle | Not NULL | Collate | |
|---|---------------------------------|-----------------|---|---|--------|----------|---|---------|------|
| 1 | RelevsMesures_idReleveMesure | INTEGER |  |  | | |  | | NULL |
| 2 | CuvesMonitorees_idCuveMonitoree | INTEGER |  |  | | |  | | NULL |
| 3 | niveau | FLOAT | | | | | | | NULL |
| 4 | RSSI | INTEGER | | | | | | | NULL |

| | Type | Nom | Détail |
|---|---|-----|--|
| 1 |  PRIMARY KEY | | (RelevsMesures_idReleveMesure, CuvesMonitorees_idCuveMonitoree) |
| 2 |  FOREIGN KEY | | (RelevsMesures_idReleveMesure) REFERENCES RelevsMesures (idReleveMesure) |
| 3 |  FOREIGN KEY | | (CuvesMonitorees_idCuveMonitoree) REFERENCES Cuves |

Etudiant IR3 (CELIE Cédric) :

I) Présentation des mes tâches :

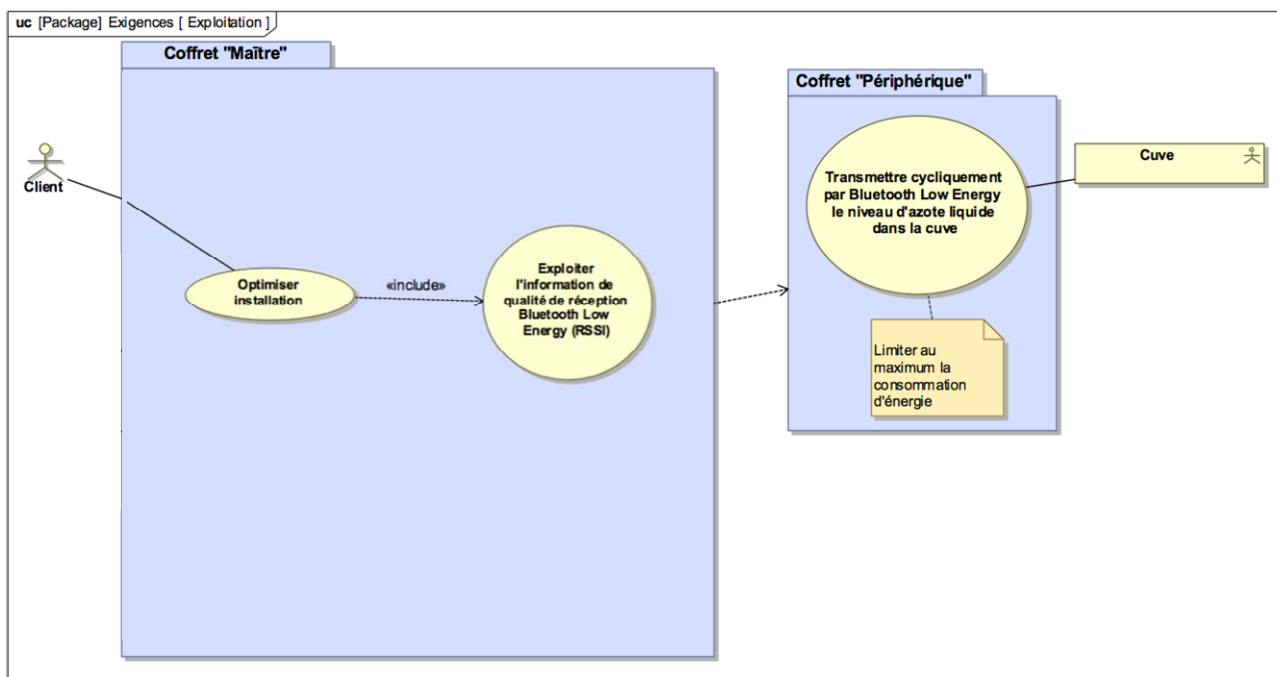
1) Objectifs :

Dans l'application de supervision, un boîtier maître constitué d'une Raspberry Pi équipée d'une carte d'extension se charge de récupérer par Bluetooth Low Energy les informations de niveau depuis un ensemble de boîtiers périphériques reliés chacun à une cuve.

Il faudra donc placer le coffret maître à un endroit stratégique pour avoir la meilleure réception Bluetooth possible de l'ensemble des cuves.

Ma première tâche consistait à coder une première application graphique d'aide au placement optimal du coffret Maître.

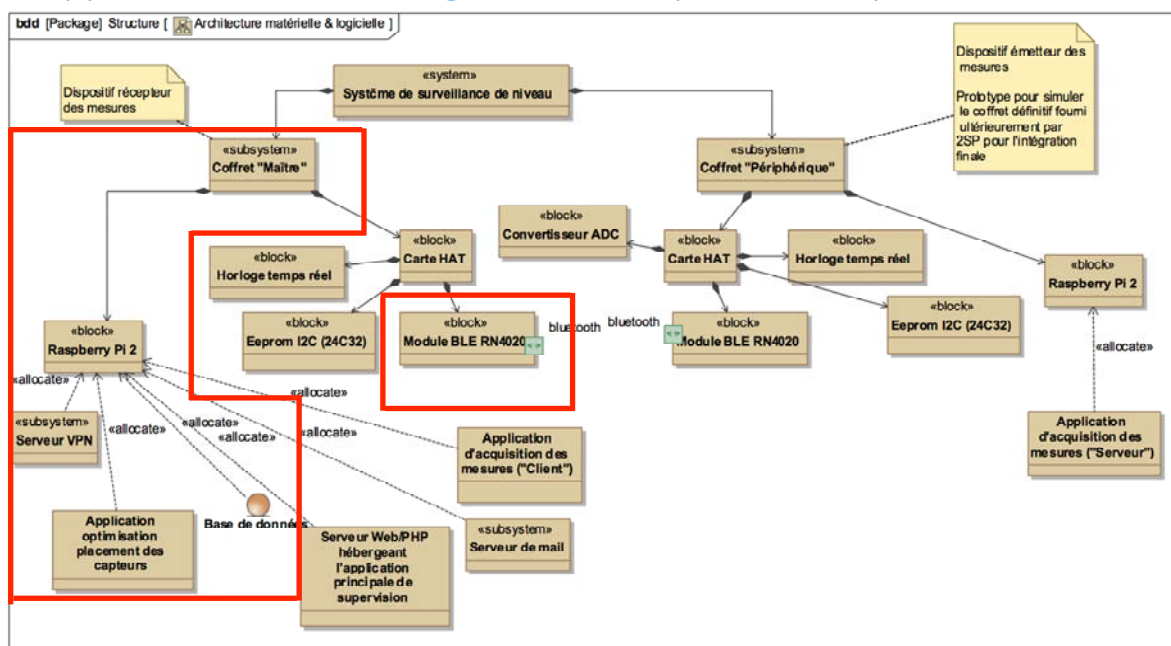
Dans le cahier des charges deux solutions étaient proposées pour créer cette application : Qt/C++ ou Node.js. J'ai choisi la solution Node.js car elle me semblait plus innovante. Ci-dessous, le cas d'utilisation associé à cette tâche.



Ma seconde tâche consistait à mettre en place un accès sécurisé depuis l'extérieur de l'entreprise au site web de supervision codé par les autres membres de mon équipe. Cet accès permettra une surveillance à distance du niveau des cuves depuis un smartphone, une tablette ou un ordinateur. Selon le cahier des charges, l'accès au site web devait être sécurisé par un VPN ou, à minima, à travers une DMZ. J'ai choisi la solution VPN car, outre la très grande sécurité qu'elle propose, elle me semblait la plus adaptée à l'infrastructure réseau en vigueur dans le lycée.



2) Rappel sur l'architecture globale et les problématiques :

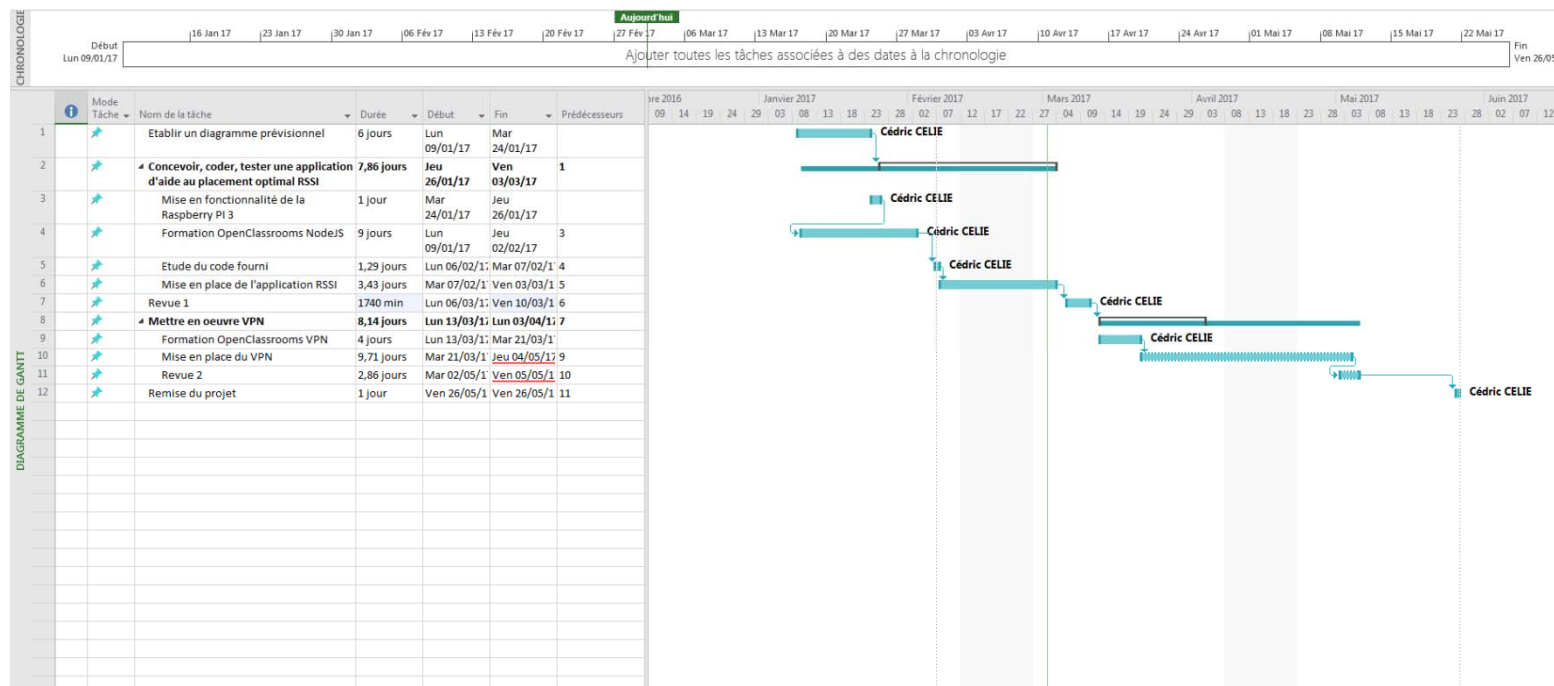


Etudiant IR 3 Cédric CELIE

Ci-dessus voici l'architecture matérielle et logicielle ainsi on peut voir encadré en rouge les matérielles et logicielles sur les quelles j'ai travaillé. Contrairement à l'implantation prévu avec de nombreuses cuves, j'ai du faire en fonction d'un seul Module Bluetooth Low Energy RN4020 périphérique équipant une carte RN4020 Bluetooth PICTail Low PICTail / PICTail Plus.



3) Planification prévisionnelle :



II) Travail accompli dans les itérations précédentes :

1) Découverte de l'écosystème Node.js :

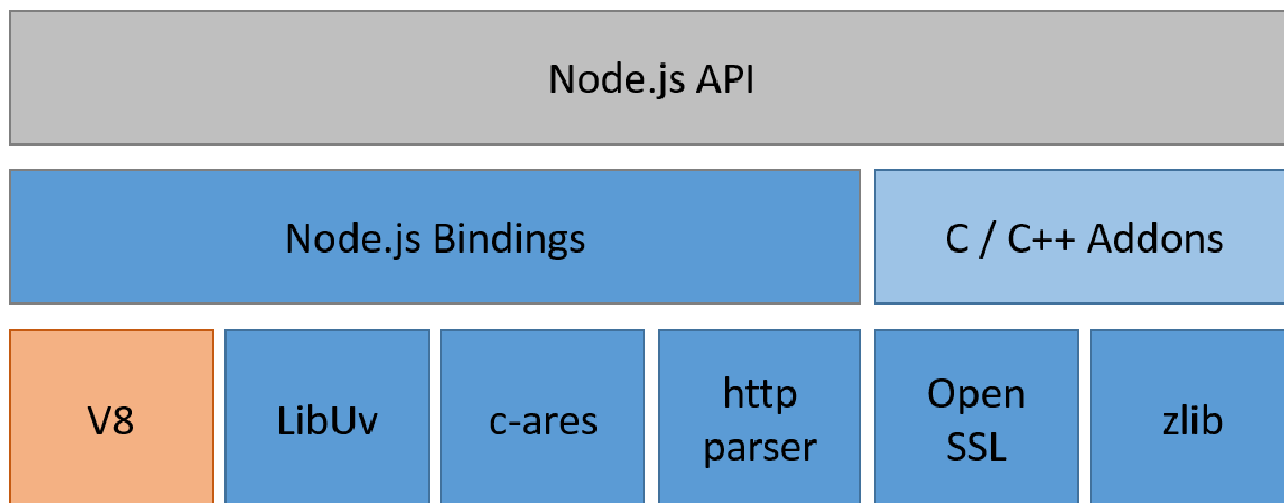
a) Node.js + Express.js :

Node.js

Node.js est une plateforme de développement Javascript. Ce n'est pas un serveur, ni un framework, c'est juste un ensemble composé de :

- Un interpréteur Javascript (V8)
- Une librairie permettant l'utilisation d'entrées/sortie asynchrones (libuv)
- De bibliothèques et d'APIs diverses permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connexions réseau ou encore créer un fichier.





Ses avantages :

- Ses qualités :
- Performance,
- Mode asynchrone,
- Gestion native du http,
- Multi-plateformes (mobile, desktop, tv),
- Portabilité du code JavaScript aussi bien côté serveur que client,
- Une communauté très active.



Ses inconvénients :

- Environnement nouveau donc pas encore mature
- Surveiller les mises à jour

Express.js

Nodes.js est un environnement de bas niveau. Express.js est un cadre d'application open-source pour Node.js conçu pour construire des applications Web. Je l'ai donc utilisé pour aller plus vite dans la création de l'application de positionnement optimal du boîtier maître.

Express.js propose par exemple des fonctions de routage qui vont permettre de rediriger les demandes des clients vers le code les traitant. Ci-dessous, un exemple de code qui occasionne le renvoi vers le client du fichier index.html lorsque celui-ci sollicite l'accès à la racine du site web.

Exemple :

```

30 app.use(express.static(__dirname + '/public')); // Utilisation des fichier ( css, js, img) dans le dossier public.
31
32 app.get('/', function(req, res, next) { // Routage de l'url
33   res.sendFile(__dirname + '/index.html') // Accès au fichier index.html lors de la demande de page ('/').
34 });
35

```



b) Module liaison série :

Pour accéder depuis Node.js à la liaison en série par laquelle on communique avec le module Bluetooth RN4020, j'ai utilisé un module qui s'appelle Serialport. Il permet de lire les infos du port série mais aussi d'y lire ou écrire.

Pour l'utiliser, on doit d'abord l'inclure avec l'instruction `require`.

```
var SerialPort = require('serialport');
```

Ensuite on instancie la classe `SerialPort` en précisant le port série que l'on veut utilisé et les options de configuration (ex. : vitesse, terminateur de ligne).

Enfin on l'utilise l'objet créé pour communiquer avec l'équipement distant sachant que les entrées/sorties sont asynchrones. Cela implique par exemple que la lecture depuis le port série se fait sur événement après enregistrement d'une fonction qui sera appelée automatiquement sur réception de données (un **callback** dans la terminologie Node.js)

Plus bas, un exemple de code qui initialise le port `/dev/ttyS0` de la Raspberry Pi avec une vitesse de 115200 bauds puis enregistre une fonction anonyme pour prendre en charge la réception de données (`→ serial.on('data', ...)`). Cette fonction décortique les trames reçues avec des expressions régulières pour en extraire les différents champs.

```
var serial = new SerialPort('/dev/ttyS0', {  
  baudRate: 115200,  
  parser: SerialPort.parsers.readline('\n')  
});  
  
serial.on('data', function(data) {  
  console.log('Data: ' + data);  
  if ( /\w*,\w*,\w*/.test(data) ) {  
    macAddr = data.match(/^[0-9,A-F]{12}/i);  
    rssi = data.match(/^[^,]*[^\n]$/i);  
    rssi = parseInt(rssi, 16);  
  }  
});
```

Les trames en question sont celles émises par les boîtiers périphériques lorsqu'ils désirent signaler leur présence au boîtier maître (mode advertising). Ces trames sont constituées en particulier : de l'adresse MAC du module RN4020 périphérique et d'un nombre indiquant le niveau de réception du module (RSSI).

Dans le code ci-dessus, ces informations sont stockées temporairement dans les variables `macAddr` et `rssi` avant stockage dans un tableau. Ce tableau informe donc des modules périphériques à proximité.



c) Module accès BDD :

De façon à informer l'utilisateur de la manière la plus conviviale possible, l'application d'optimisation de placement du boîtier maître utilise la BDD du serveur web de supervision de façon à extraire les informations sur la cuve à laquelle est associé tel ou tel module bluetooth périphérique (dénomination de la cuve, zone dans laquelle elle se trouve, référence du module RN4020).

L'accès à la BDD de type SQLite a été réalisé en utilisant le module Node.JS nommé **sqlite3**.

Ci-dessous un code qui montre la mise en œuvre du module (chargement, initialisation et récupération des champs requis à travers une requête SQL).

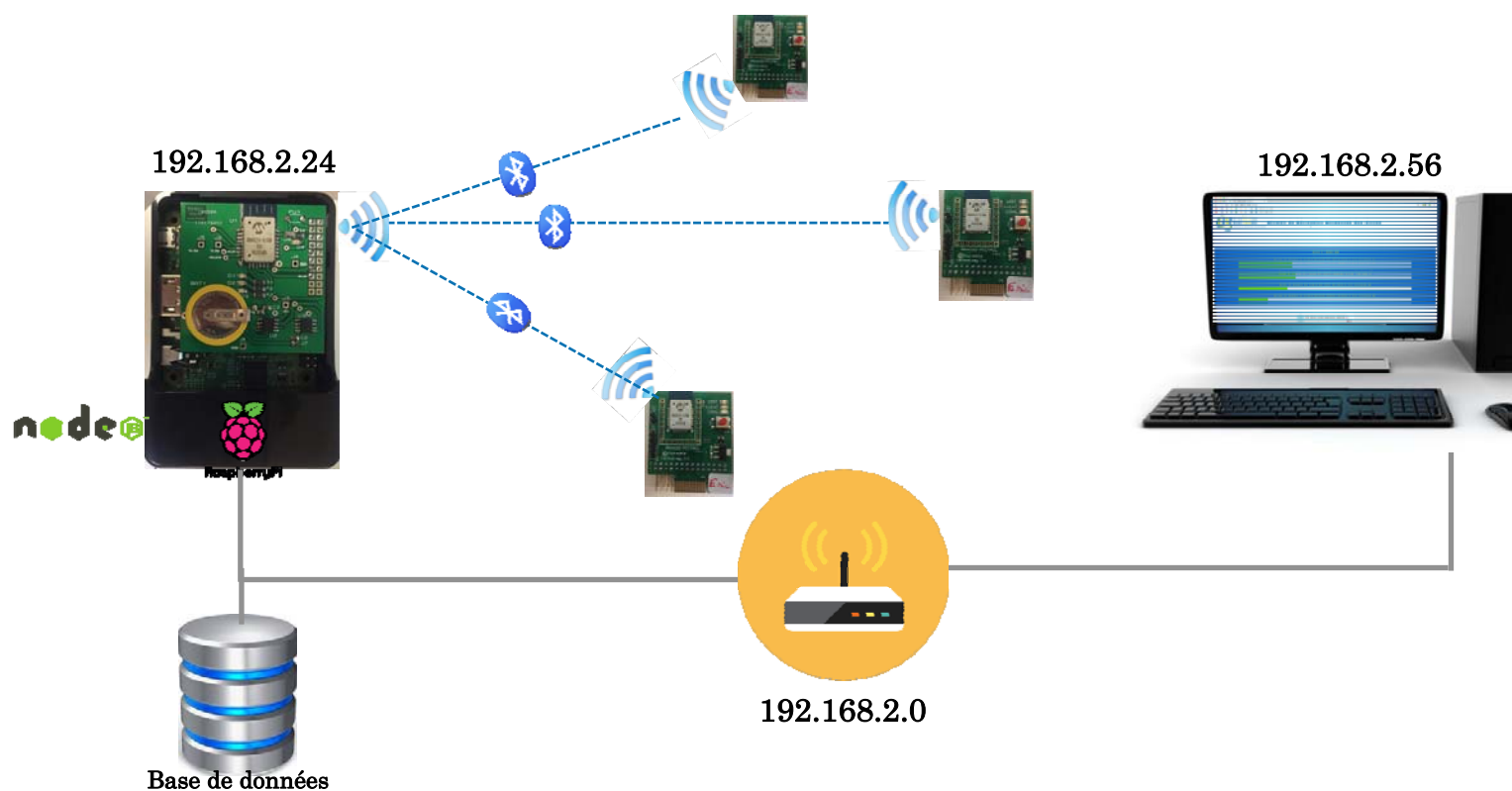
```
3 var sqlite3 = require('sqlite3').verbose();
4
5 var db = new sqlite3.Database('./../public_html/dashboard/db/db2sp.db', sqlite3.OPEN_READONLY);
6
```

```
db.serialize(function() {
  req="SELECT CM.nom, MB.referenceModule, Z.denomination FROM CuvesMonitorees A
  db.get(req, function(err, row) {
    if(row) {
      ble = row.referenceModule;
      tank = row.nom;
      zone = row.denomination;
      eventEmitter.emit('rssi');
    } else {
      console.log("Pas de référence trouvée");
      ble = tank = zone = "<N/A>";
    }
  });
});
```



2) Application d'optimisation de placement du module central :

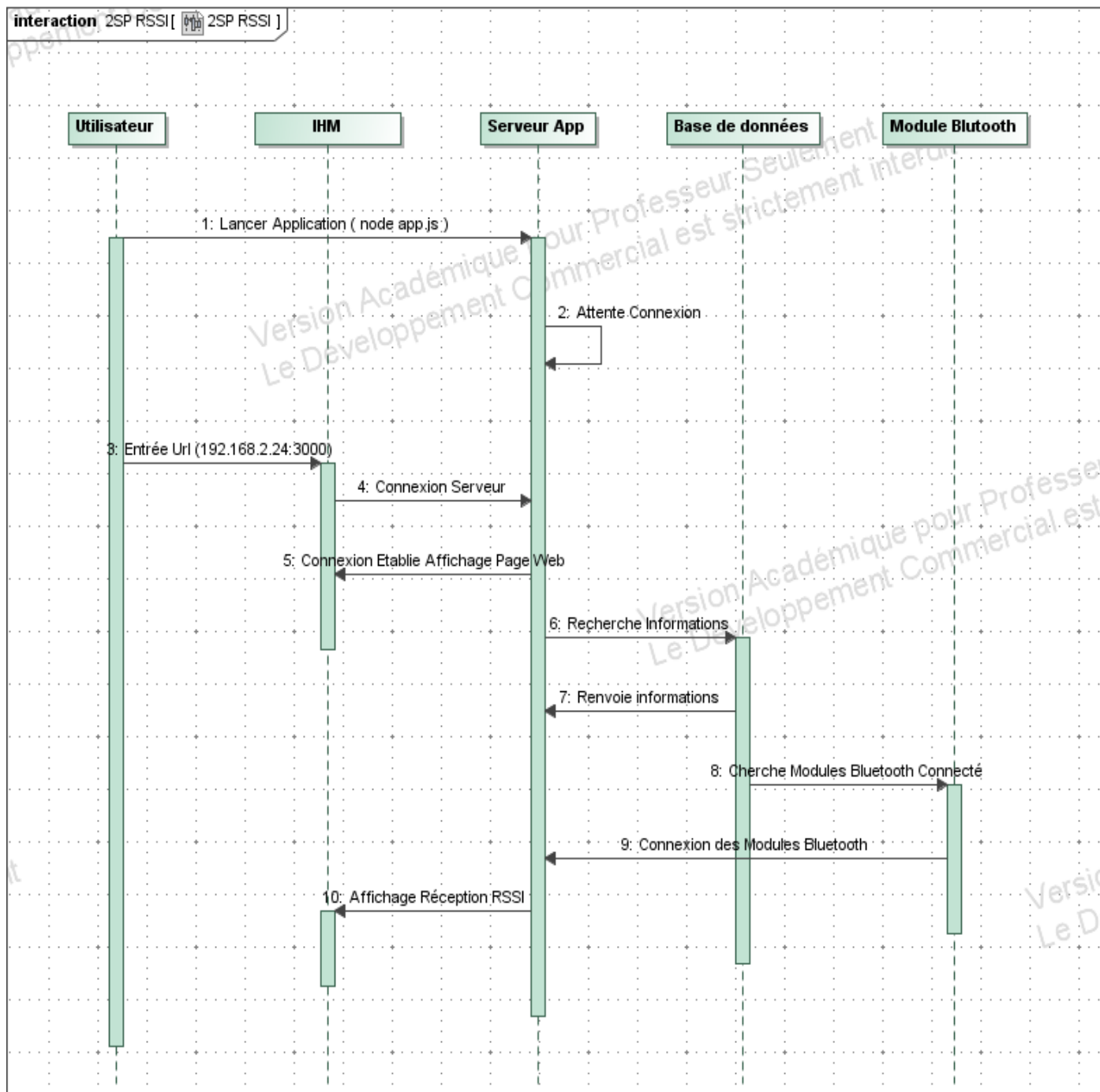
a) Synoptique de l'architecture de l'application :



Sur le schéma ci-dessus on peut y retrouver, la raspberry Pi faisant office de serveur qui communique en bluetooth Low Energy avec les Modules Bluetooth RN4020 normalement placé sur les cuves. La Raspberry Pi faisant appelle à l'environnement node.js, va chercher les informations directement dans la base de données et en se connectant à celle ci sous le même réseau grâce à une machine on peut retrouver l'IHM avec les barres de progressions de chaque cuves.

b) Fonctionnement global :





c) Mise en Oeuvre :

c.1) Détails de l'implémentation :

Une grande partie de mon projet a consisté à concevoir l'application d'aide au placement optimal du coffret naître. Elle est codée en grande partie en langage javascript (Node.js) pour la partie serveur (app.js) qui tourne sur la raspberry et pour la partie cliente intégrée dans la page web (client.js). Une fois la page web principale de l'application chargée sur le navigateur, une communication par websocket est établie entre le serveur et la partie cliente intégrée à la page web. Du HTML (index.html) et CSS (style.css) sont également utilisés pour la mise en page.

Le logiciel SQLite Studio a également été utilisé pour visualiser/modifier la base de données a des fins de test.

Pour lancer l'application, on peut se connecter en SSH à la Raspberry avec l'adresse : 192.168.2.24.

Dès que l'on est connecté et situé dans le répertoire de l'application, on lance son exécution avec la commande « `node app.js` ».

Pour accéder à l'IHM de l'application, il suffit alors se rendre sur l'URL 192.168.2.24:3000 avec un navigateur internet (port 3000 = port par défaut pour les applications Node.js).



Dans la capture d'écran ci-dessus, on voit d'une part que 4 modules Bluetooth Low Energy ont été configuré en amont dans la base de données car 4 barres de progression sont affichées et d'autres part que ces modules sont actifs puisque les barres de progression sont renseignées en vert.

J'ai indiqué plus haut qu'un seul module RN4020 périphérique était à ma disposition par l'intermédiaire d'une carte de développement Pictail. Cependant, les équipements Bluetooth Low Energy sont désormais fréquents. J'ai donc utilisé mon ordinateur Mac Book, un smartphone et même un « traceur de clés » pour émuler plusieurs module Bluetooth RN4020. On notera au passage que le Bluetooth Low Energy est bien standardisé puisque tous mes équipements sont bien détectés par mon application.

Au final, nous pouvons placer notre coffret maître où il y a la meilleure réception RSSI en déterminant par essais successifs l'endroit où les niveaux des barres de progression sont au maximum.

c.2) Principe de la détection du RSSI :

La technologie de communication utilisé est du Bluetooth Low Energy. L'avantage de ce mode de communication est la réduction du coût et de la consommation en puissance, tout en conservant une portée de communication équivalente d'un Bluetooth classique.

J'ai donc utilisé le signal RSSI (Received Signal Strength Indication) pour avoir un relevé de valeurs pour placer le coffret maître en fonction de ces mesures. Le RSSI est une mesure de la puissance en réception d'un signal reçu d'une antenne exprimé en Dbm.

Sachant que pour avoir un bon niveau RSSI il faut qu'il varie entre -30 dBm (au meilleur des cas) et -90 dBm (au moins bon des cas).

Le dBm est une abréviation du rapport de puissance en décibels (dB) entre la puissance mesurée et un milliwatt (mW).

Pour l'exprimé en Watt il faudra faire : $P_{dBm} = 10 \cdot \log P/P_0$

Dans notre cas $P_{max} = -30dBm$ donc $P = P_0^{-3} \cdot 10^{-30/10} = 1 \cdot 10^{-3} mW$

Dans notre cas $P_{min} = -90dBm$ donc $P = P_0^{-3} \cdot 10^{-90/10} = 1 \cdot 10^{-9} mW$

Son utilité est de fournir une indication sur la puissance du signal reçu.

Ainsi, en fonction de cette mesure de la puissance de réception je pourrai placer au mieux mon coffret maître.

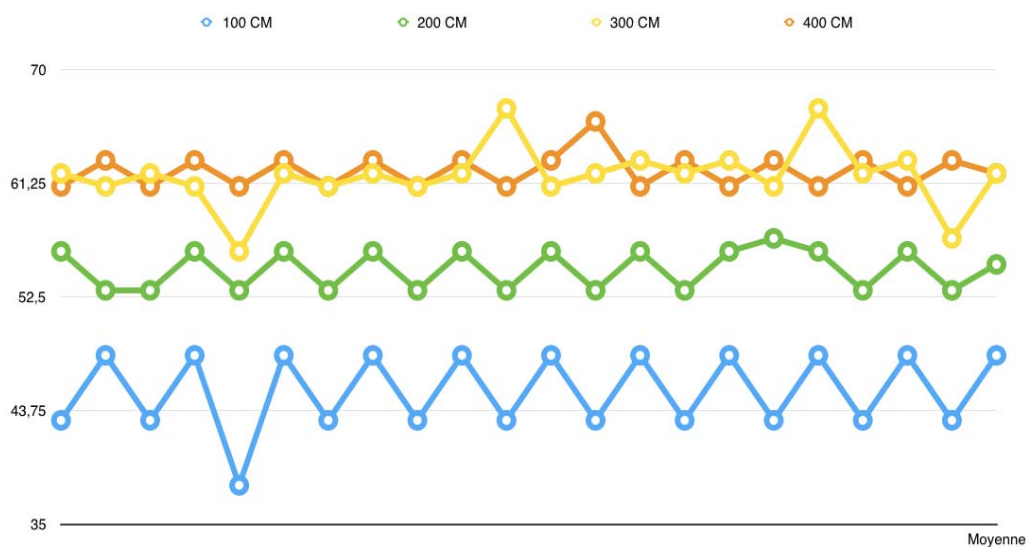
d) Problèmes rencontrés :

Problème 1: Réflexions

J'ai fait une campagne de mesure car je trouvais que le niveau RSSI variait beaucoup.

La campagne de mesure que j'ai faite était approximative car il y avait des réflexions dans les pièces lorsque le module BLE où la Raspberry se trouvait dans un coin ou contre un mur. Les résultats du tableau sont exprimés en dBm.

| | 100 CM | 200 CM | 300 CM | 400 CM |
|---------|--------|--------|--------|--------|
| | 43 | 56 | 62 | 61 |
| | 48 | 53 | 61 | 63 |
| | 43 | 53 | 62 | 61 |
| | 48 | 56 | 61 | 63 |
| | 38 | 53 | 56 | 61 |
| | 48 | 56 | 62 | 63 |
| | 43 | 53 | 61 | 61 |
| | 48 | 56 | 62 | 63 |
| | 43 | 53 | 61 | 61 |
| | 48 | 56 | 62 | 63 |
| | 43 | 53 | 67 | 61 |
| | 48 | 56 | 61 | 63 |
| | 43 | 53 | 62 | 66 |
| | 48 | 56 | 63 | 61 |
| | 43 | 53 | 62 | 63 |
| | 48 | 56 | 63 | 61 |
| | 43 | 57 | 61 | 63 |
| | 48 | 56 | 67 | 61 |
| | 43 | 53 | 62 | 63 |
| | 48 | 56 | 63 | 61 |
| | 43 | 53 | 57 | 63 |
| | 48 | 56 | 62 | 63 |
| | 43 | 53 | 62 | 63 |
| | 48 | 56 | 63 | 61 |
| | 43 | 53 | 57 | 63 |
| Moyenne | 48 | 55 | 62 | 62 |



Problème n°2 : Variations de réception RSSI

Les variations importantes de niveau RSSI peuvent s'expliquer par les phénomènes de réflexion mais j'ai aussi remarqué une anomalie dans les messages de log que j'affiche dans la console du navigateur (accessible depuis les outils de développement de Chrome par exemple).

On peut ainsi voir dans l'exemple ci-dessous, qu'un périphérique avec l'adresse MAC 80 :4E :81 :49 :4C :59 vient prendre la place du périphérique 00 :1E :C0 :1B :20 :B8 normalement associé dans la base de données au module nommé « BLE Pictail (Eric) ».

Je n'ai pas pu déterminer la cause de cette erreur car elle arrive de manière intempestive. Cependant, 2 explications sont possibles. Soit cela vient d'une erreur de logiciel soit les trames qui nous parviennent sont erronées suite par exemple à des « interférences » entre les différents modules en présence.

Exemple ci-dessous :

| | |
|----|--|
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -65 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -70 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 9801A789D198 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -67 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -65 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -65 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -65 / CryoTank 03 / Salle IR |
| 10 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 804E81494C59 / -58 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -67 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -65 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -61 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -67 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -60 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |
| 4 | BLE Pictail (Eric) / 001EC01B20B8 / -69 / CryoTank 03 / Salle IR |
| 2 | BLE Pictail (Eric) / 001EC01B20B8 / -63 / CryoTank 03 / Salle IR |

On peut voir que le nom du module est bon mais que l'adresse MAC ne correspond pas.

En conclusion, dans l'installation définitive il y aura inévitablement des réflexions. Pour corriger ce problème et celui du, au lieu d'afficher des valeurs instantanées dans les barres de progression, on peut afficher des valeurs qui sont le résultat de moyennes glissantes. Ces moyennes permettront de filtrer les valeurs « anormales ».



III) Travail accompli dans l'itération 3 :

1) Evolution application Node.js :

a) Défauts de l'application et ses solutions :

Outre les problèmes de variation du RSSI, l'application affiche toutes les barres de progression des modules BLE même si ceux-ci ne sont pas en service ou actifs.

J'ai fait des recherches pour voir comment adapter l'IHM par rapport au contenu de la BDD qui contient les informations sur les modules BLE. Une des solutions consiste à utiliser un moteur de template (en l'occurrence PUG/Jade) pour construire « dynamiquement » la page web faisant office d'IHM en fonction du contenu de la BDD

2) Mise en oeuvre :

J'ai donc créer un dossier views, où j'ai placé mon fichier index.pug ci-dessous. La syntaxe de jade ressemble au HTML mais permet d'avoir plus de souplesse et de gagner du temps.

Le but sera d'afficher grâce à une boucle les barres de réceptions seulement pour les cuves existantes dans la base de données. Pug n'étant pas encore opérationnel pour le moment, je compte faire fonctionner le tout pour la revue finale.

```
//-- Ce template Pug a été généré automatiquement depuis le fichier html de départ grâce au site
//-- http://html2pug.com/
doctype html
html
  head
    title 2SP 2017
    meta(charset='utf8')
    link(rel='stylesheet', href='./css/style.css')
    link(href='https://fonts.googleapis.com/css?family=Baloo+Da', rel='stylesheet')
  body
    .container
      header
        h1
          a(href='#') Visu capteurs
      .content
        section
          //-- Boucler à travers le tableau 'tanks' pour afficher autant de progress bar qu'il y a de cuves monitorées
          //-- dans la BDD
          each t in tanks
            fieldset
              legend Niveau de réception du "#{t.ble}" quipant la cuve "#{t.nom}"
              (#{t.zone})
              meter.rssi(id=t.nom, value='-100', min='-100', max='0')
      footer
        #logo
          img(src='./img/logo.jpg')
          |
          span © 2017 Lycée Alphonse Benoît
        script(src='./socket.io/socket.io.js')
        script(src='./js/client.js')
```

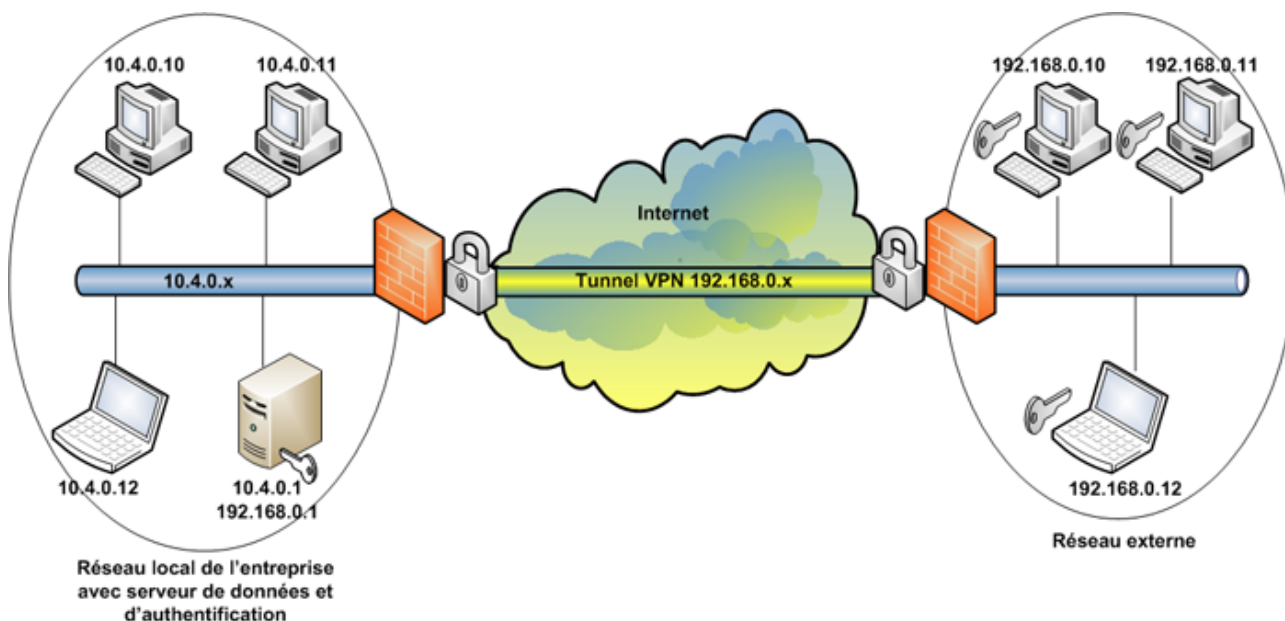


3) Accès sécurisé à l'appli web de supervision des cuves :

Le client désire accéder de manière sécurisée à l'application web depuis internet. Comme déjà précisé, la solution retenue est la mise en place d'un VPN (Virtual Private Network) à travers un serveur OpenVPN installé sur la Raspberry Pi.

a) Présentation du VPN :

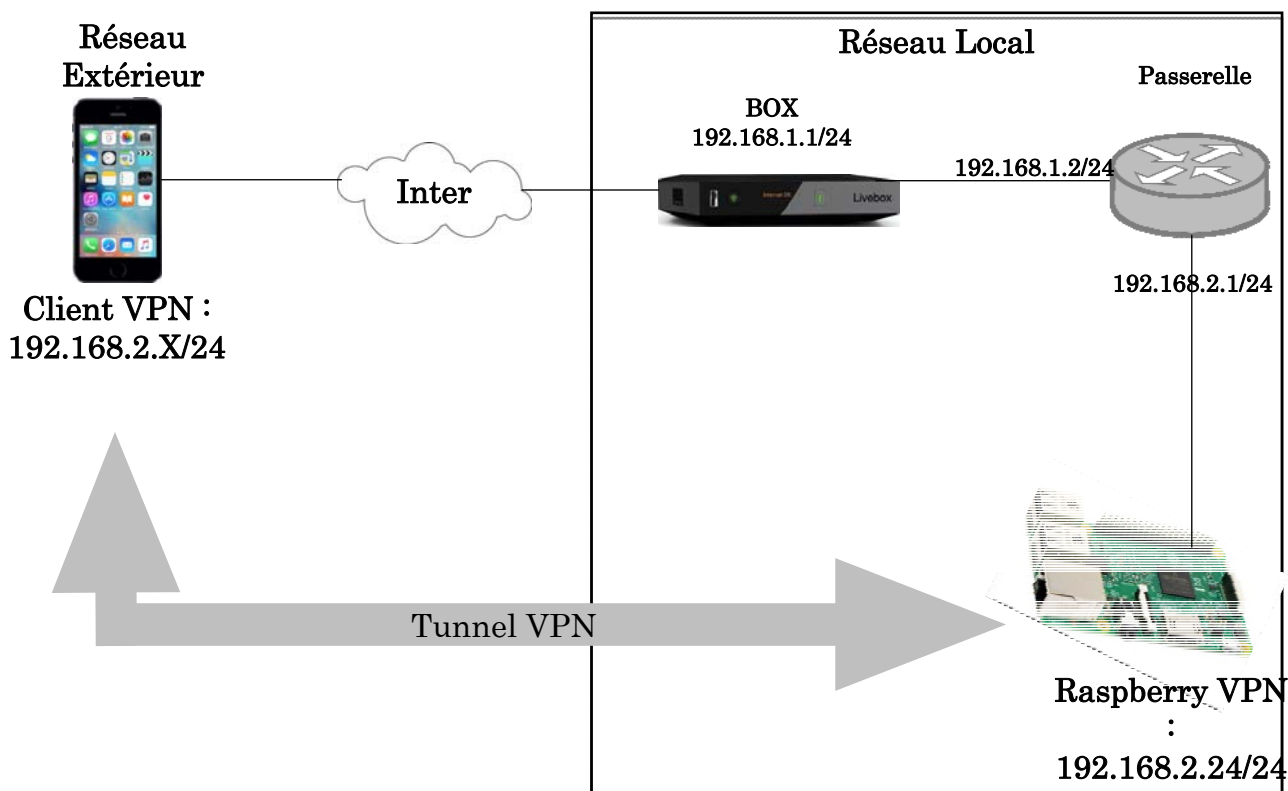
Le VPN peut se résumer à une connexion entre un ordinateur distant et un ordinateur local à travers un « tunnel » sécurisé qui fait en sorte que l'ordinateur distant est considéré comme un ordinateur du réseau local.



b) Principe de fonctionnement global :

Le serveur VPN Virtual Private Network permet donc d'accéder à un réseau interne depuis un autre réseau. C'est pour cette raison qu'il est très utilisé dans le domaine du télétravail. On peut par exemple accéder depuis notre maison à un serveur normalement uniquement accessible depuis notre entreprise.

Architecture désirée dans le cadre du projet :



Un VPN repose sur un ou des protocoles, appelé protocoles de tunnelisation (ou tunneling). Ce sont des protocoles permettant aux données passant entre deux réseaux physiques d'être sécurisées par des algorithmes de chiffrement. On utilise d'ailleurs le terme de « tunnel » pour mettre l'accent sur le fait qu'entre l'entrée et la sortie d'un VPN les données sont chiffrées et protégées. Lorsqu'un VPN est établi entre deux réseaux physiques, l'élément qui permet de chiffrer et de déchiffrer les données du côté client (ou utilisateur) est nommé « Client VPN ». On appelle « Serveur VPN » l'élément qui chiffre et qui déchiffre les données du côté de l'organisation.



c) Présentation OpenVPN :

OpenVPN permet la création d'une liaison chiffrée entre une machine et un serveur hébergé sur Internet (par exemple chez un fournisseur d'accès se trouvant en France ou à l'étranger).

Tous les accès à Internet seront alors vus à partir de l'adresse IP de ce serveur VPN et non plus par celle de la machine.



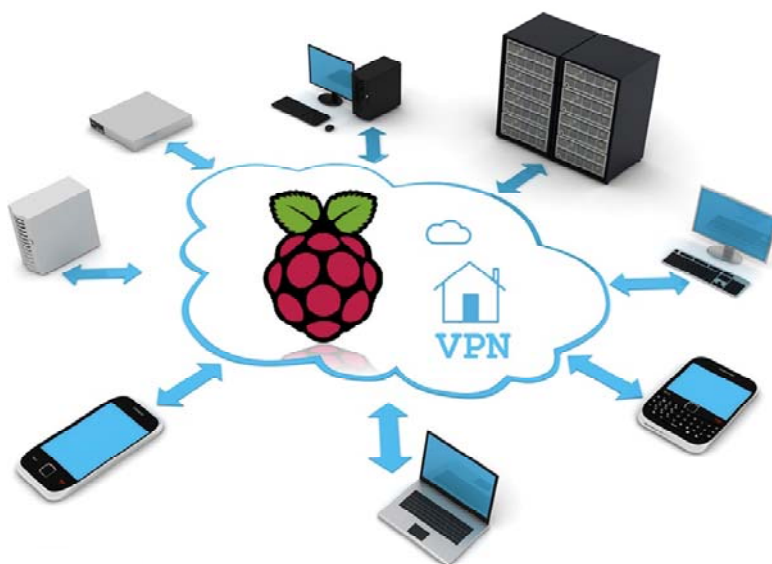
OpenVPN n'est pas un VPN IPsec. C'est un VPN SSL se basant sur la création d'un tunnel IP UDP ou TCP (dans notre cas il sera en UDP) authentifié et chiffré avec la bibliothèque OpenSSL.

OpenVPN cumule de nombreux avantages car il est open source et fiable, multi-plates-formes (Windows, Linux Mac OS X ...), très répandu il est capable de tourner sur n'importe quel port en écoute côté serveur (80 ou 443 compris) et surtout il est gratuit au niveau logiciel.

Par contre, il a un inconvénient, les protocoles qu'il utilise ne sont pas pris en charge nativement par les systèmes d'exploitations, et il est donc nécessaire d'installer son client (un logiciel qui envoie des données sur un serveur) sur votre ordinateur.

d) Mise en œuvre :

D'un côté nous aurons donc la Raspberry faisant office de serveur et de l'autre côté nous pourrons nous connecter avec un smartphone ou avec une autre machine à notre application.



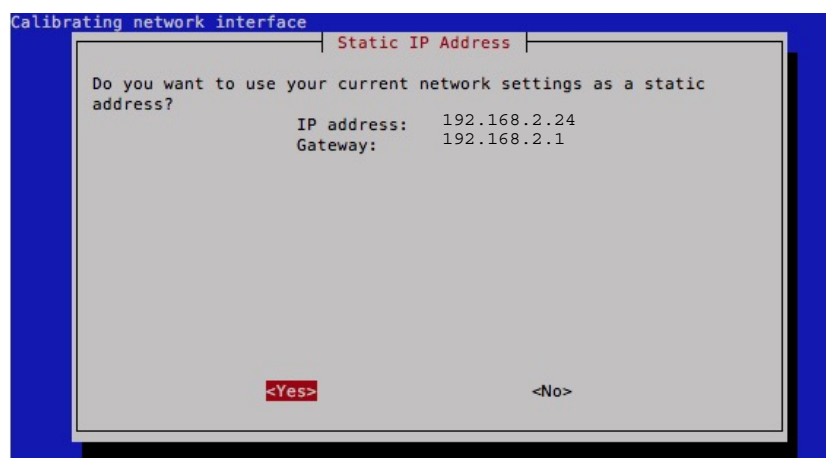
e) Installation PiVPN :

PiVPN est un script découvert sur internet qui va automatiser l'installation d'OpenVPN sur la raspberry. Il va aussi me permettre de créer autant de profils clients que je veux.

Ce script est le moyen le plus simple et le plus rapide de configurer un serveur OpenVPN sur une raspberry. Il propose de surcroît une configuration extrêmement sûre.

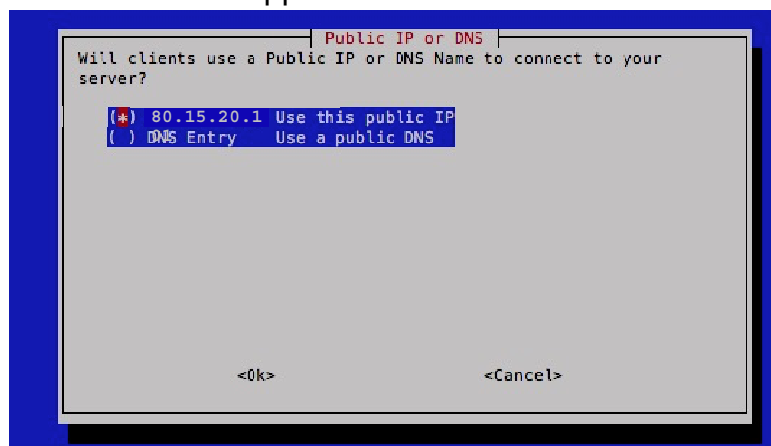
f) Paramétrage au niveau de l'installation :

Lors de l'installation de PiVPN, plusieurs informations m'ont été demandées.



J'ai choisis d'utilisé de l'UDP sur le port 1194 par défaut avec un niveau de chiffrement de 2048-bit.

UDP est un protocole stateless (*sans état*), on peut le comparer au courrier. On place le message à envoyer dans une enveloppe qui contient toutes les informations nécessaires au routage, l'adresse IP et le port (les coordonnées du destinataire), puis envoie de l'enveloppe.

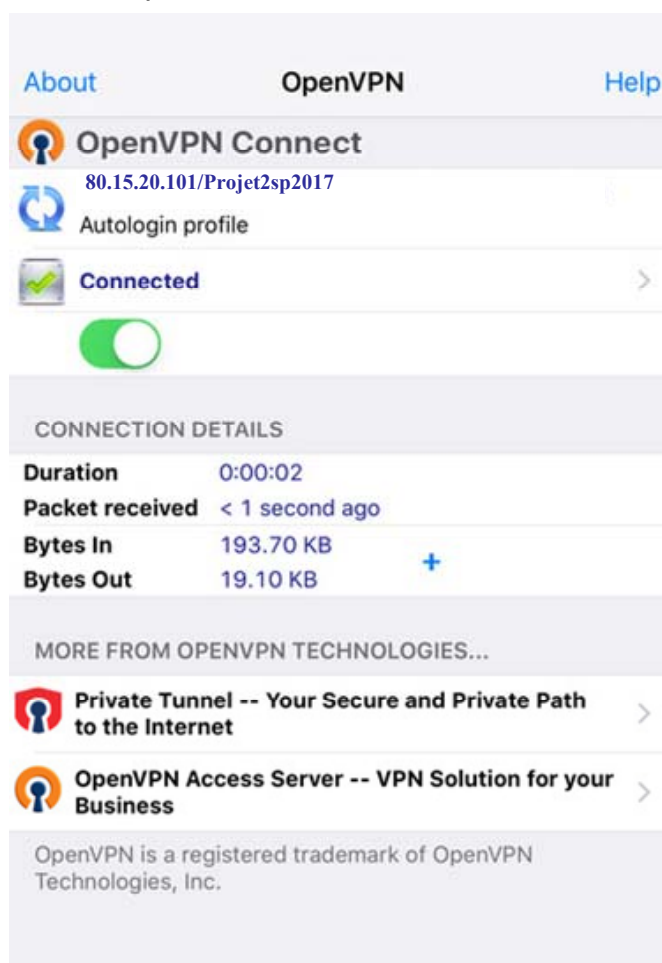


Ici, l'IP Public de notre box nous demande d'être confirmé.

Pour finir la configuration, j'ai du créer un nouveau profil de connexion à mon serveur OpenVPN grâce à la commande "pivpn add". J'ai donner un nom (Projet2sp2017) et un mot de passe, ainsi une clé est générée et un fichier lab.ovpn est placé dans le dossier /home/pi/ovpn.

g) Test depuis le client "OpenVPN Connect":

Après avoir créé le client, j'ai du transférer le fichier Projet2sp2017.ovpn sur mon Iphone grâce à Itunes. Une fois transféré j'ai pu me connecter à mon serveur virtuel. Pour pouvoir se connecter il faut être sur un réseau différent de notre serveur. Il faut notamment s'assurer que le WiFi est désactivé.



Lorsque nous sommes connectés, un icône VPN s'affiche en haut de notre écran. J'ai donc pu ensuite me connecter à ma raspberry, et afficher mon application d'aide au placement optimal en attendant d'avoir l'application web de supervision des cuves.





IV) Conclusion :

1) Bilan :

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète du métier.

En effet, la prise d'initiative, le respect des délais et le travail en équipe seront des aspects essentiels de notre futur métier et nous avons eu la chance de découvrir cet environnement.

Ce projet m'a aussi permis de découvrir de nouvelles façons de travailler notamment avec la raspberry ou encore avec de nouveaux langages ou plateforme de développement notamment node.js.



2) Pistes d'améliorations :

Aujourd'hui l'application de placement sur ordinateur ressemble à ça :



Pour améliorer mon application j'ai pensé à adapter l'affichage de façon plus dynamique car l'affichage change selon le types de machines (tablettes, iPhone, Ordinateur).

V) Annexes :

Journal de bord :

| Date | Intervenant | Objet |
|--------------|-------------|---|
| • 09/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Prise de contact de Node.js • Mise en fonctionnalité de la Raspberry en SSH • Installation de node.js avec HomeBrew |
| • 10/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Formation OpenClassrooms Node.js <ul style="list-style-type: none"> - Jusqu'aux événements • Etat de l'application <ul style="list-style-type: none"> - J'ai fusionné les codes il gère donc à la fois les pages et les paramètres |
| • 12/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Formation OpenClassrooms Node.js • Diagramme de Gantt |
| • 16/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Formation OpenClassrooms Node.js • Réflexion sur la généralisation HTML/CSS |
| • 17/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Réflexion sur la généralisation HTML/CSS <ul style="list-style-type: none"> - Sur la barre de réception RSSI |
| • 19/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Formation OpenClassrooms Node.js <ul style="list-style-type: none"> - Sur les Socket • Etude des codes source fourni |
| • 23/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Etude des codes source fourni |
| • 24/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Réunion avec le client pour fixer les objectifs |
| • 26/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Mise en place de l'application RSSI <ul style="list-style-type: none"> - Serveur • Tutoriel de Mise en Oeuvre de l'application Web <ul style="list-style-type: none"> - Jusqu'a affichage des graphiques |
| • 30/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Copie des codes source et de la bdd sur la Raspberry • Test du code sur la Raspberry |
| • 31/01/2017 | • IR 3 | <ul style="list-style-type: none"> • Première mise en fonctionnalité de l'application • Récupération de la vdd sur la Raspberry |
| • 02/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Installation de git sur Ordinateur et Raspberry • Amélioration du code html et css |
| • 06/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Commencer dossier de la revue 1 |
| • 07/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Campagne de Mesure <ul style="list-style-type: none"> - Relever le niveau RSSI du module Bluetooth, en faire une moyenne pour voir l'évolution sur 4 éloignements différents |
| • 14/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Campagne de mesure |
| • 20/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Campagne de mesure <ul style="list-style-type: none"> - Champ libre |
| • 27/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Ajout de deux nouveaux périphériques BLE |
| • 28/02/2017 | • IR 3 | <ul style="list-style-type: none"> • Dossier Revue 1 |
| • 2/03/2017 | • IR 3 | <ul style="list-style-type: none"> • Dossier Revue 1 |



| <i>Date</i> | <i>Intervenant</i> | <i>Objet</i> |
|--------------|--------------------|--|
| • 13/03/2017 | • IR 3 | • Prise de contact des VPN |
| • 14/03/2017 | • IR 3 | • Formation OpenClassrooms OpenVPN |
| • 16/03/2017 | • IR 3 | • Formation OpenClassrooms OpenVPN |
| • 20/03/2017 | • IR 3 | • Prise de contact avec Pug |
| • 21/03/2017 | • IR 3 | • Recherche d'informations sur le moteur de template "pug" |
| • 23/03/2017 | • IR 3 | • Amélioration graphique de l'application RSSI |
| • 27/03/2017 | • IR 3 | • Amélioration graphique de l'application RSSI |
| • 28/03/2017 | • IR 3 | • Intégrer dans l'IHM une information avec pug
• Problème d'intégration |
| • 03/04/2017 | • IR 3 | • Ajout d'un affichage de valeur en chiffre RSSI |
| • 04/04/2017 | • IR 3 | • Faire fonctionner une partie du Pug.
• Ajout de début de code sur BitBucket |
| • 24/04/2017 | • IR 3 | • Réunion pour la partie commune du dossier
• Formation et mise en place VPN |
| • 25/04/2017 | • IR 3 | • Mise en place VPN
• Préparation Dossier Revue 2 |
| • 27/04/2017 | • IR 3 | • Mise en place VPN
• Préparation Dossier Revue 2 |

