

Travail de Bachelor 2017

# Enregistreur de traces, le Datalogger

## Pré-étude

Enregistreur de signaux numériques et analogiques  
haute vitesse avec détecteurs de seuils,  
triggers et équations logiques.

*Auteur :*  
**Blessemaille Sébastien**

*Conseiller :*  
**Bressy Pierre**



# 0.0 Table des matières

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Le mandant, l'entreprise BOBST MEX SA . . . . .	9
1.2	Le contexte . . . . .	9
1.3	La présentation du projet . . . . .	10
1.4	Objectifs secondaires . . . . .	10
1.5	Le cahier des charges du Datalogger . . . . .	10
<b>2</b>	<b>Le matériel</b>	<b>13</b>
2.1	Le microcontrôleur . . . . .	13
2.1.1	Raspberry Pi 2 B . . . . .	13
2.1.2	Raspberry Pi 3 B . . . . .	13
2.1.3	Synthèse . . . . .	14
2.1.4	Conclusion . . . . .	14
2.2	Les périphériques d'entrées et de sorties . . . . .	14
2.2.1	Le port RJ45 Ethernet. . . . .	14
2.2.2	Les ports USB . . . . .	14
2.2.3	Le connecteur jack 3.5 mm . . . . .	14
2.2.4	Le port HDMI . . . . .	15
2.2.5	Le port d'extension GPIO . . . . .	15
2.2.6	Le Wifi . . . . .	15
2.3	Le câblage des GPIO. . . . .	16
2.3.1	Composants utilisés . . . . .	16
<b>3</b>	<b>Le périphérique FTDI</b>	<b>17</b>
3.1	Introduction . . . . .	17
3.2	Le type . . . . .	17
3.3	Caractéristiques des interfaces série . . . . .	18
3.4	Le driver . . . . .	18

3.5	La méthode de connexion . . . . .	18
3.6	Les contrôles de fonctionnement . . . . .	18
3.6.1	Description . . . . .	18
3.6.2	Diagramme de flux . . . . .	19
3.6.3	Le contrôle de flux matériel RTS-CTS . . . . .	20
3.7	Description des tests . . . . .	21
3.7.1	Les cartes de tests utilisées . . . . .	21
3.7.2	TEST_D2XX_RTS_CTS . . . . .	22
3.7.3	TEST_D2XX_RTS_1 . . . . .	23
3.7.4	TEST_D2XX_RTS_0 . . . . .	23
3.7.5	TEST_D2XX_LIVE . . . . .	24
3.7.6	TEST_D2XX_LIVE_VMAX . . . . .	25
3.7.7	Les sources de chez FTDI . . . . .	28
<b>4</b>	<b>Protocoles utilisés</b>	<b>29</b>
4.1	La trame d'envoi . . . . .	29
4.2	La trame de réception . . . . .	29
4.3	La trame de validation . . . . .	30
4.3.1	Résultat théorique attendu . . . . .	30
<b>5</b>	<b>Le traitement des données</b>	<b>33</b>
5.1	L'acquisition . . . . .	33
5.2	Le contrôle de la cohérence des trames . . . . .	33
5.2.1	Description . . . . .	33
5.2.2	Le diagramme de flux . . . . .	34
5.3	L'évaluation des triggers . . . . .	35
5.3.1	Entrée digitale . . . . .	35
5.3.2	Entrée analogique . . . . .	35
5.4	L'évaluation de l'équation . . . . .	35
<b>6</b>	<b>L'environnement de développement</b>	<b>37</b>
6.1	Visual studio . . . . .	37

6.2	QT . . . . .	37
6.3	Mon choix . . . . .	37
6.4	Les phases de développement . . . . .	38
6.5	Les tests unitaires . . . . .	38
6.6	La compilation . . . . .	39
6.6.1	Les choix possibles . . . . .	39
6.6.2	Conclusion . . . . .	39
<b>7</b>	<b>Le rafraîchissement de l'affichage</b>	<b>41</b>
7.1	La persistance rétinienne . . . . .	41
7.2	La fréquence de rafraîchissement d'un écran . . . . .	41
7.3	Le choix de la fréquence de rafraîchissement de l'écran . . . . .	41
<b>8</b>	<b>L'interface graphique</b>	<b>43</b>
8.1	Introduction . . . . .	43
8.2	Le choix des widgets . . . . .	43
8.2.1	Affichage de texte non modifiable par l'utilisateur . . . . .	43
8.2.2	Boutons avec deux ou moins de sélections (type ON/OFF) . . . . .	43
8.2.3	Boutons avec plus de deux sélections . . . . .	43
8.2.4	Curseur horizontal . . . . .	43
8.2.5	Les traces . . . . .	44
<b>9</b>	<b>L'ancienne interface utilisateur</b>	<b>45</b>
9.1	Les différentes pages . . . . .	45
9.1.1	La page principale . . . . .	45
9.1.2	Configuration des différents trigger . . . . .	45
9.1.3	Configuration de l'équation logique . . . . .	47
9.1.4	L'ajustement du pré-trigger . . . . .	47
9.1.5	L'affichage des triggers et de l'équation logique . . . . .	48
9.1.6	L'affichage des traces . . . . .	49
9.1.7	Le menu de debug . . . . .	49
<b>10</b>	<b>Conception d'une Interface Homme Machine "IHM"</b>	<b>51</b>

10.1 Introduction . . . . .	51
10.2 Les bonnes pratiques . . . . .	51
10.3 Méthodologie pour la conception d'IHM . . . . .	52
10.3.1 Les trois domaines de connaissances sur lesquelles la méthodologie se base	52
10.4 Les étapes de la conception . . . . .	53
10.5 Les sources . . . . .	53
<b>11 La nouvelle interface</b>	<b>55</b>
11.1 La page principale . . . . .	55
11.1.1 Aperçu . . . . .	55
11.1.2 Description . . . . .	55
11.1.3 Les widgets pressenti . . . . .	56
11.2 La page de configuration . . . . .	57
11.2.1 Aperçu . . . . .	57
11.2.2 Description . . . . .	57
11.2.3 Description du bouton start . . . . .	58
11.2.4 Le widget pressenti du bouton start . . . . .	58
11.2.5 Description de la moitié de gauche . . . . .	59
11.2.6 Les widget pressentis de la moitié de gauche . . . . .	59
11.2.7 Description de la moitié de droite . . . . .	60
11.2.8 Les widgets pressentis de la moitié de droite . . . . .	60
11.3 La page de visualistion des triggers et de l'équation logique . . . . .	61
11.3.1 Aperçu . . . . .	61
11.3.2 Description . . . . .	61
11.3.3 Les widgets pressentis . . . . .	62
11.4 La page de visualistion . . . . .	63
11.4.1 Description . . . . .	63
11.4.2 Les widgets presentis . . . . .	63
11.5 La page de debug . . . . .	64
11.5.1 Description . . . . .	64
<b>12 Diagramme des classes</b>	<b>65</b>

<b>13 La sauvegarde des données</b>	<b>67</b>
13.1 L'emplacement . . . . .	67
13.2 Le volume . . . . .	67
13.3 Le type de fichier . . . . .	67
13.3.1 Type binaire . . . . .	67
13.3.2 Type CSV . . . . .	67
13.3.3 Type text . . . . .	67
13.4 Conclusion . . . . .	68
<b>14 Les périphériques d'accès et d'interfaces</b>	<b>69</b>
<b>15 Evaluation des risques</b>	<b>71</b>
15.1 Tableau des risques . . . . .	71
15.2 Graphique des risques . . . . .	72
15.3 Synthèse . . . . .	72
15.4 Conclusion . . . . .	72
<b>16 Planning</b>	<b>73</b>
<b>17 Conclusion</b>	<b>77</b>
17.1 Bilan personnel . . . . .	77
17.2 Bilan du projet . . . . .	77
17.3 Message personnel . . . . .	77
<b>A P.V. séance</b>	<b>81</b>
A.1 Historique des entretiens . . . . .	81
A.2 9 mai 2017 . . . . .	82
<b>B Tableau des tâches</b>	<b>83</b>
<b>C Tableau de calculs des masques binaires</b>	<b>87</b>
<b>D Temps de réalisation de la pré-étude</b>	<b>91</b>



## 1.0 Introduction

### 1.1 Le mandant, l'entreprise BOBST MEX SA



Fondée en 1890 à Lausanne (Suisse) par Joseph Bobst, la société BOBST est présente dans plus de 50 pays, possède 12 sites de production dans 8 pays et emploie près de 5 000 personnes dans le monde. La maison mère est en Suisse, à Mex, où travaillent environ 1200 personnes. Elle est le leader mondial en tant que fabricant de machines d'emballage.

Elle est aussi l'un des premiers fournisseurs mondiaux d'équipements et de services destinés aux fabricants d'emballages et d'étiquettes des industries de la boîte pliante, du carton ondulé et des matériaux flexibles.

Le nombre de machines dans le monde peut se compter en millier de toutes générations. La qualité, la fiabilité et le service après-vente sont les forces de l'entreprise et en ont fait sa réputation.

### 1.2 Le contexte

Le Datalogger est un projet de la société existant depuis près de dix ans. Le but était de le faire développer et évoluer sur quelques années avec les apprentis automaticiens et électroniciens. Mais cela n'a jamais abouti.

Les besoins sur le terrain étant toujours d'actualité et mon responsable étant aussi totalement convaincu de son utilité, le démarrage du projet a recommencé il a deux ans environ afin de faire le budget et de trouver des fonds.

Pourquoi faut-il des fonds ? Car il faut faire développer des cartes électroniques d'acquisition et un boîtier industriel afin de protéger le système de l'environnement industriel des fabricants d'emballages très sale et source de perturbation.

Du fait de mes études dans ce domaine, le coût de développement a fondu et le projet a fini par être validé l'année passée. Le prototype de la carte d'acquisition définitive est en cours de fabrication et des tests de fonctionnement sont prévus dans le courant de l'été 2017. Ce projet de carte a été fait en collaboration avec un apprenti électronicien de 4ème année et deux ingénieurs électroniciens.

Grâce à ce développement, notre apprenti a pu acquérir les connaissances et compétences nécessaires afin de développer et de router une seconde carte de sortie 24V et en faire son TPI final. Elle pourra être utilisée comme un périphérique de sortie du Datalogger. Afin de permettre mon projet de Bachelor, une carte de simulation a été développée. Elle envoie les trames normalisées et, par conséquent, me permet de valider l'application.

## 1.3 La présentation du projet

Le Datalogger est un appareil destiné à aider au dépannage des machines. Le but premier est de pouvoir trouver des pannes intermittentes d'une durée de l'ordre de 1Khz.

Son principe est de pouvoir visualiser et enregistrer de signaux analogiques et digitaux en temps réel avec des détecteurs de seuils sélectionnables et réglables.

Sa force est l'équation logique de ces triggers, cette fonction n'est disponible dans aucun appareil existant sur le marché actuellement.

Les utilisateurs principaux seront des techniciens, il devra être intuitif et facile à utiliser.

## 1.4 Objectifs secondaires

1. Faire une sauvegarde de la configuration dans un fichier et pouvoir le rappeler.
2. Permettre une sauvegarde des données sur un support connecté à un port USB. Ce qui permettrait d'augmenter la capacité d'enregistrement.
3. Créer un fichier du type CSV, afin de pouvoir analyser les données avec un autre outil comme Excel.
4. Définir la trame de configuration et pouvoir l'envoyer à la carte d'acquisition. Ceci afin de configurer les registres en fonction des options choisies depuis l'interface.

## 1.5 Le cahier des charges du Datalogger

### Descriptif de l'appareil

Traitement des données envoyées par la carte d'acquisition sans perte.

Si les conditions de trigger sont réunies, sauvegarde des données.

Affichage des signaux sur écran.

Signaux à afficher sélectionnables et échelle verticale réglable pour les analogiques.

### Conditions de trigger

4 entrées sélectionnables parmi les 4 entrées digitales trigger et les 2 premières entrées analogiques.

Pour chacune des entrées sélectionnées, sélection de l'état direct, état inversé, flanc montant ou flanc descendant.

Equation : (A OR/AND/NOT B) OR/AND/NOT (C OR/AND/NOT D)

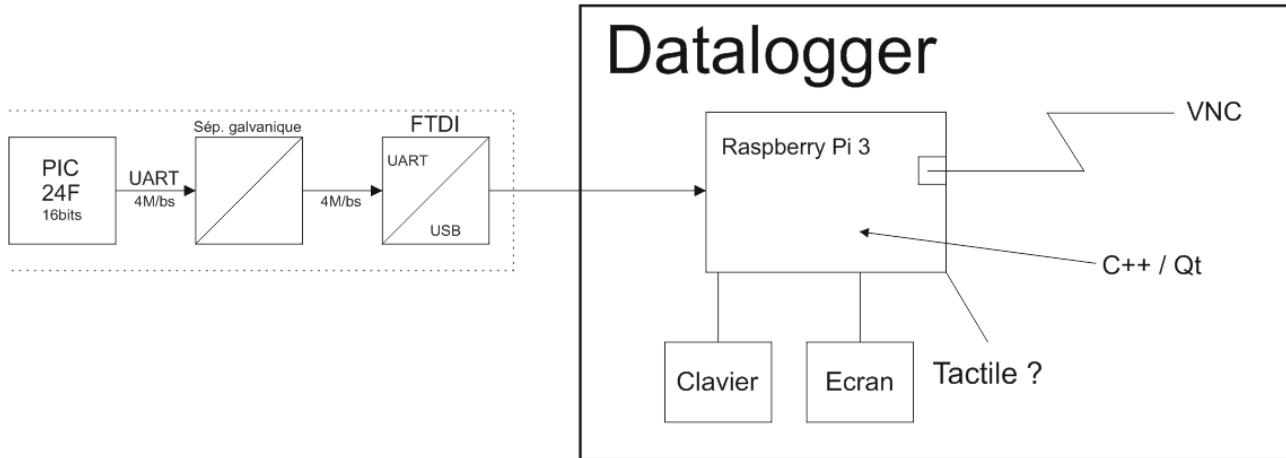
Pré-trigger réglable de 0% à 100%

## Travail à effectuer dans le cadre du Bachelor

- Traitement des données à la volée.
  - Lire la trame reçue par le PIC.
  - Contrôler le compteur de trames et des valeurs si cohérentes.
  - Évaluer les conditions de trigger.
- Traitement des triggers à la volée.
- Visualisation des données en temps réel.
- Affichage graphique des 16 canaux digitaux, 4 canaux analogiques et 3 triggers.
- Implémenter une fonction « roll on ».
- Affichage de valeurs numériques type jeton.
- Bouton d'arrêt hold.
- Enregistrement d'un tour de compteur de trame, 16bit, 65536 valeurs (en RAM).
- Une vitesse d'acquisition minimum de 10 kHz doit être atteinte.
- Évaluer la vitesse d'acquisition maximum à la fin du projet.

## Architecture pressentie

FIGURE 1.1 – Architecture pressentie





## 2.0 Le matériel

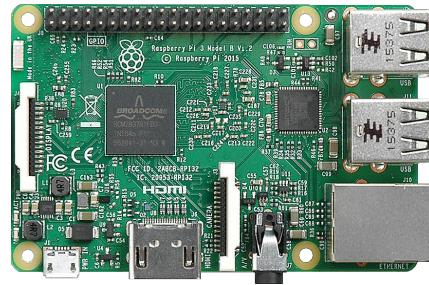
### 2.1 Le microcontrôleur

FIGURE 2.1 – Raspberry Pi 2



source : [www.raspberrypi.org](http://www.raspberrypi.org)

FIGURE 2.2 – Raspberry Pi 3



source : [www.wikimedia.org](http://www.wikimedia.org)

#### 2.1.1 Raspberry Pi 2 B

- CPU<sup>1</sup> 900 MHz quad-core RM Cortex-A7
- Alimentation 2.0A
- 46 CHF

#### 2.1.2 Raspberry Pi 3 B

- CPU<sup>1</sup> 1.2 GHz 64-bit quad-core ARMv8
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth faible consommation
- Alimentation 2.5A
- 48 CHF

- 1GB RAM LPDDR2 900 MHz
- 4 ports USB 2.0
- 40 pins GPIO
- 1 port Full HDMI
- 1 port Ethernet
- Connecteur jack 3.5mm audio et video
- Interface série pour camera (CSI<sup>2</sup>)
- Interface série pour affichage (DSI<sup>3</sup>)
- Slot carte micro SD
- VideoCore IV 3D graphics core

1. Central Processor Unit, Unité de Processeur central  
2. Camera Serial Interface  
3. Display Serial Interface

### 2.1.3 Synthèse

#### Les atouts du RPi2 :

- Moins gourmand en énergie.
- Moins cher à l'achat.

#### Les atouts du RPi3 :

- Wifi embarqué
- Bluetooth 4.1
- Vitesse du processeur de 1.2 GHz

### 2.1.4 Conclusion

*La différence de prix est négligeable par rapport à l'évolution technologique. Je vais choisir le raspberry Pi 3 pour mon projet. J'aurai plus de réserve en puissance de calcul et aussi une meilleure possibilité de connectivité.*

## 2.2 Les périphériques d'entrées et de sorties

### 2.2.1 Le port RJ45 Ethernet.

Il peut être utilisé pour un accès en VNC via une connexion locale LAN.

Cela permettrait un accès sécurisé et crypté à l'application depuis n'importe quelle machine sur le réseau pour autant que l'accès soit autorisé. Une authentification par mot de passe sera demandée.

Dans le futur, l'appareil sera accessible depuis le WEB via une plateforme déjà utilisée sur nos machines. Ceci permettrait d'y accéder depuis le monde entier.

Le transfert de fichiers peut être possible avec le protocole SFTP.

### 2.2.2 Les ports USB

Un port sera dédié à l'acquisition de données, le fichier spécial "USBtty0" de l'OS ne peut pas être utilisé pour l'application.

Ce problème entraîne que la gestion complète de la communication et la configuration du chip FTDI seront faites par l'application.

Les autres ports seront libres d'utilisation pour des périphériques de saisie, comme un clavier, une souris, ou un périphérique de sauvegarde de type clef USB.

### 2.2.3 Le connecteur jack 3.5 mm

Il n'est pas prévu d'utiliser le jack dans ce projet.

## 2.2.4 Le port HDMI

Le port HDMI<sup>4</sup> est utilisé pour la connexion d'un périphérique d'affichage.

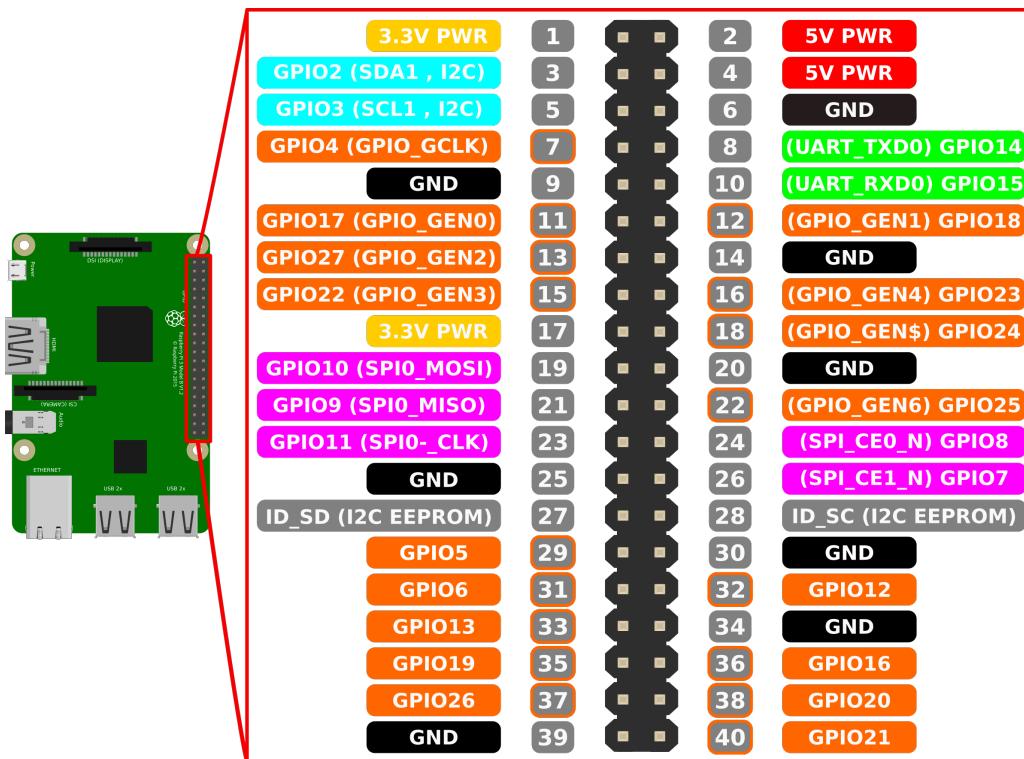
Il va me permettre de visualiser l'application. Cette option peut être utile chez les clients afin de visualiser la machine en temps réel.

## 2.2.5 Le port d'extension GPIO

L'interface GPIO<sup>5</sup> permet une extension des fonctionnalités du Raspberry-Pi. Les broches permettent ainsi au Raspberry-Pi d'avoir un accès au « monde extérieur » ; notamment en agissant en sortie depuis des LED, des afficheurs LCD, des capteurs, ou encore en lisant l'état d'un interrupteur.

Ci-dessous le plan de connexion du port GPIO<sup>5</sup> :

FIGURE 2.3 – Plan des GPIO du RPi3



source : [openclipart.org](http://openclipart.org)

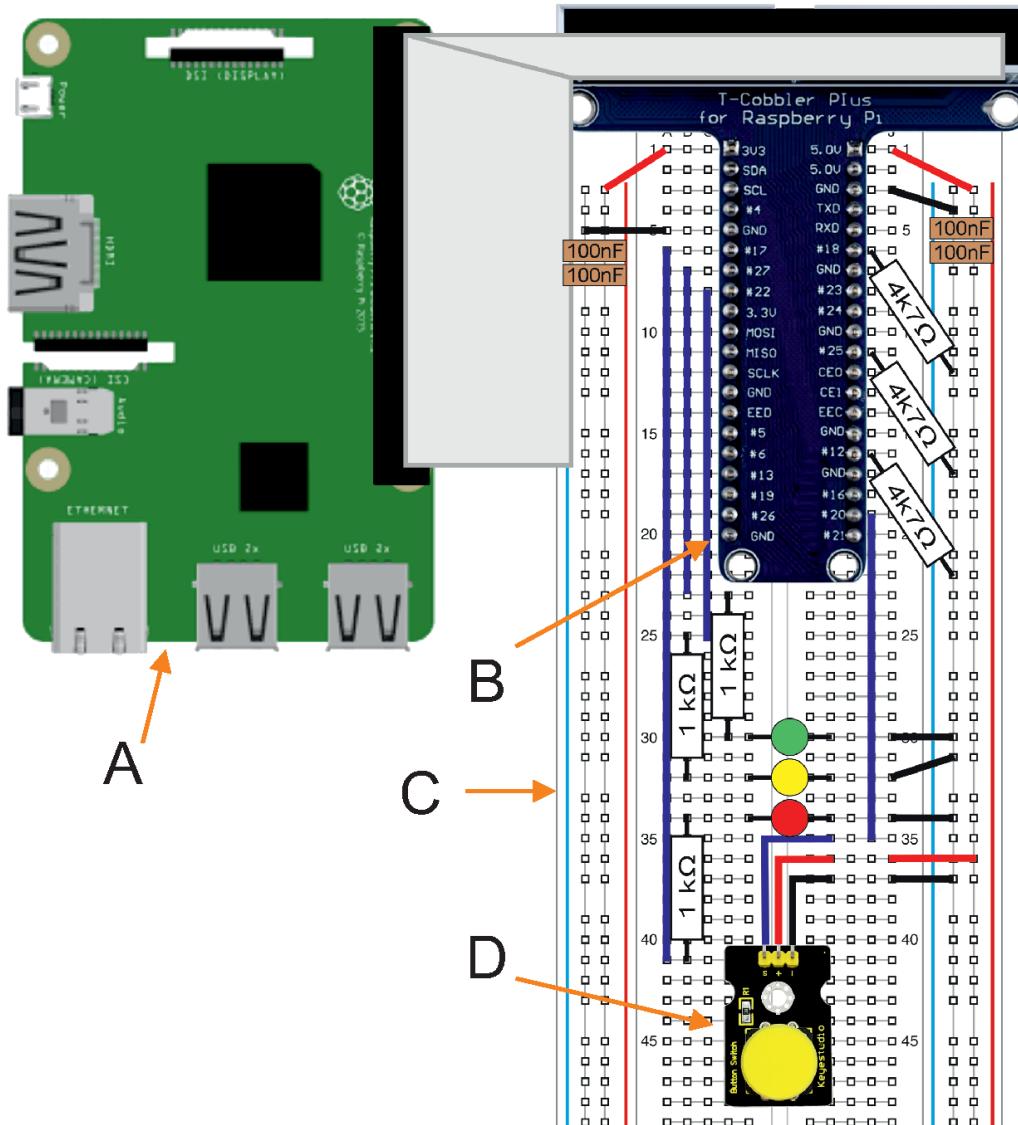
## 2.2.6 Le Wifi

Peut-être utilisé pour se connecter à distance avec des smartphones, tablettes etc. avec une connexion sécurisée en VNC, il est possible de visualiser ou d'interagir avec l'application.

- 4. High-Definition Multimedia Interface
- 5. General Purpose Input/Output

## 2.3 Le câblage des GPIO.

FIGURE 2.4 – Plan de cablage pour l'application



source A : [fritzing.org](http://fritzing.org)

source B : [www.ece.utah.edu](http://www.ece.utah.edu)

source C : [fr.aliexpress.com](http://fr.aliexpress.com)

source D : [www.keyestudio.com](http://www.keyestudio.com)

Le câblage et les éléments ont été ajoutés par mes soins sur l'image

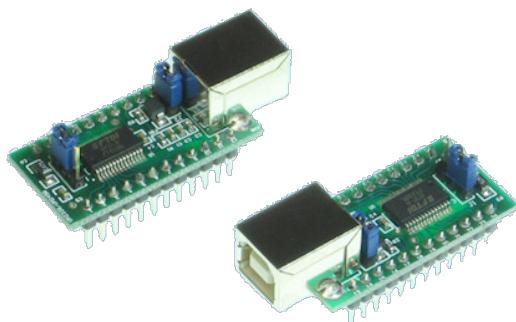
### 2.3.1 Composants utilisés

- Résistance 1K : 1 k $\Omega$  1/4 W 5%
- Résistance 4K7 : 4,7 k $\Omega$  1/4 W 5%
- Condensateur : 100 nF céramique
- Led rouge : HLMP 4700 rouge
- Led verte : HLMP 4700 verte
- Led jaune : HLMP 4700 jaune
- Kit de connexion : Kit de chez Vococal acheté sur Amazon.co.uk

## 3.0 Le périphérique FTDI

### 3.1 Introduction

FIGURE 3.1 – chip FTDI



source : [www.meraprojekt.com.pl/um232r.html](http://www.meraprojekt.com.pl/um232r.html)

Le chip FTDI<sup>1</sup> utilisé est de type UM232R avec un UART IC FT232R

- Vitesse maximale est de 3 Mbaud
- 128 bytes de buffer de réception
- 256 bytes de buffer d'émission
- Utilisable avec le driver d2XX
  - Compatible Windows XP, 7, 10 et plus
  - Compatible Linux 2.4 et plus
  - Compatible Android J2XX

### 3.2 Le type

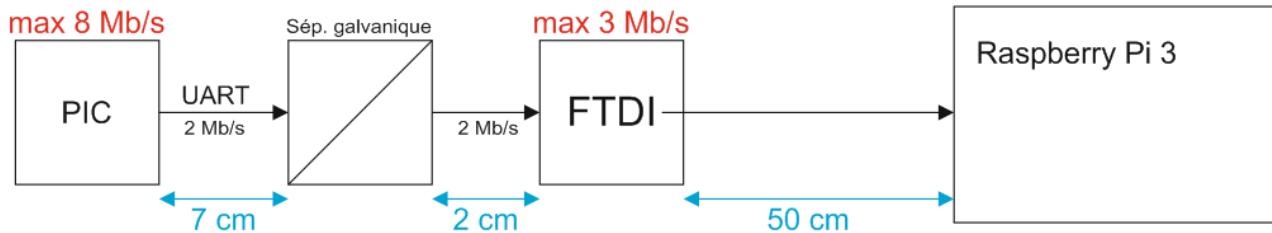
Il sera de type série. Il n'y aura pas de contrôle d'erreur du fait de la vitesse de transmission demandée et, si l'on réfléchit dans l'absolu, les données transmises sont déjà périmées.

Le bit de parité n'a pas été implémenté, car la liaison série entre le PIC<sup>2</sup> et le chip FTDI<sup>1</sup> est soudée et d'une longueur d'environ 5 cm, ce qui représente un faible risque de perturbation.

1. Fournisseur du chip, Future Technology Devices International Ltd.  
2. Microcontrôleur de chez Microchip

### 3.3 Caractéristiques des interfaces série

FIGURE 3.2 – Caractéristiques des interfaces série



- Vitesse : 2 Mb/s
- Nbr de bits par mot : 8
- Bit de parité : non
- Flow control : control hardware
- Nombre de bit stop : 1

### 3.4 Le driver

Le driver utilisé est le D2xx, cela me permet de pouvoir communiquer directement avec le chip et de le configurer correctement selon mon application.

### 3.5 La méthode de connexion

La méthode de connexion est directe, mais l'OS<sup>3</sup> attache le port à un fichier spécial dès la connexion du câble. C'est très pratique dans beaucoup de cas, mais les taux de transfert configurable de base ne sont pas utilisables pour l'application.

Ce fichier empêche l'accès direct au chip et interdit de configurer le chip en direct. Il faut trouver une solution afin de libérer le port et éviter que l'OS<sup>3</sup> ne le prenne plus en charge automatiquement.

### 3.6 Les contrôles de fonctionnement

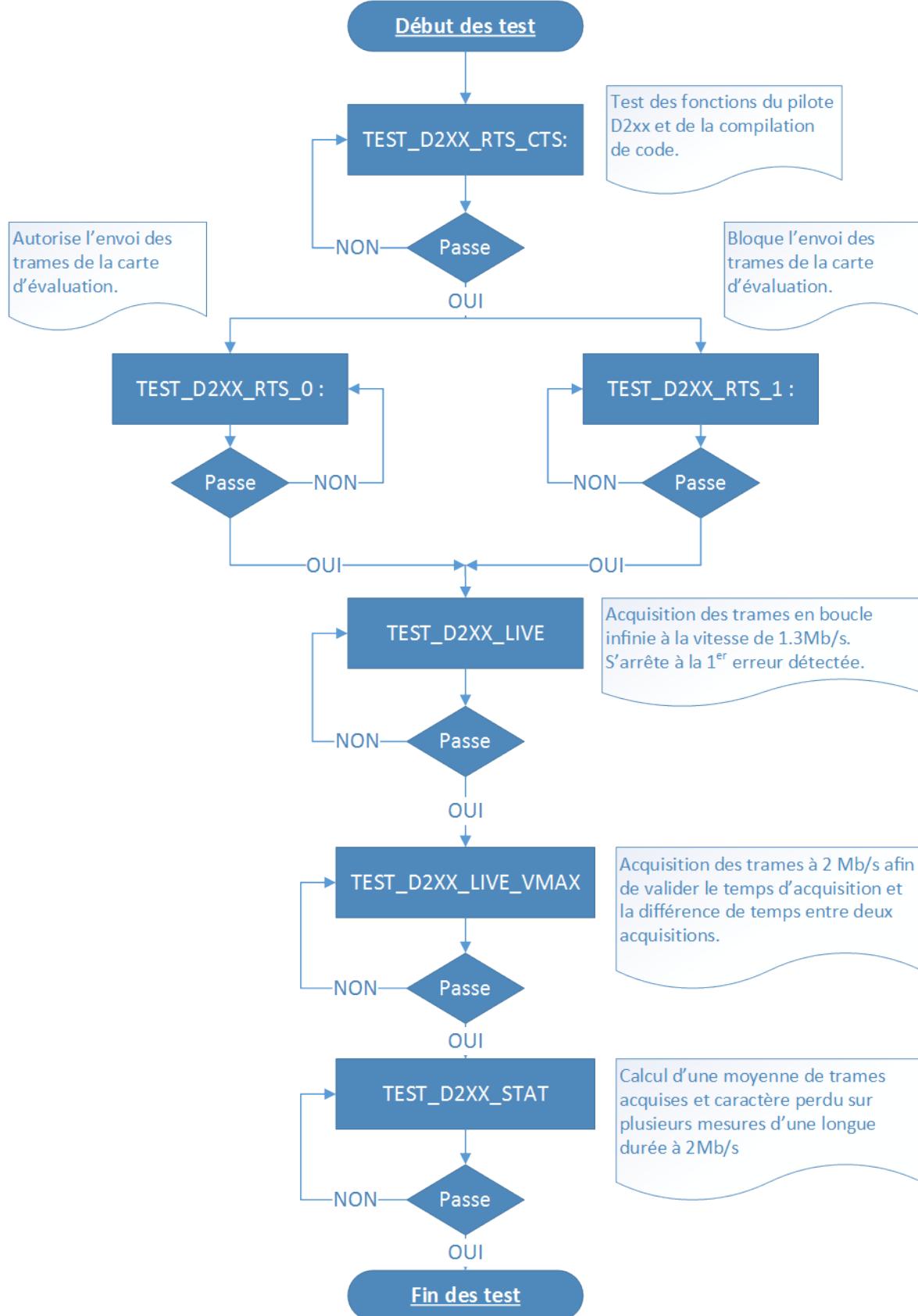
#### 3.6.1 Description

Pour la validation du driver D2XX et la vitesse décrite dans le cahier des charges, j'ai créé divers programmes de tests. Le langage C a été utilisé et compilé en natif sur le Raspberry Pi 3. Selon les besoins des tests, l'affichage des résultats se fait dans la console.

3. Operating System, système d'exploitation.

### 3.6.2 Diagramme de flux

FIGURE 3.3 – diagramme de flux des tests FTDI



### 3.6.3 Le contrôle de flux matériel RTS-CTS

Le contrôle de flux matériel RTS/CTS est un mécanisme qui fait partie des standards RS232.

Il utilise deux broches supplémentaires, RTS (Request to Send) et CTS (Clear to Send).

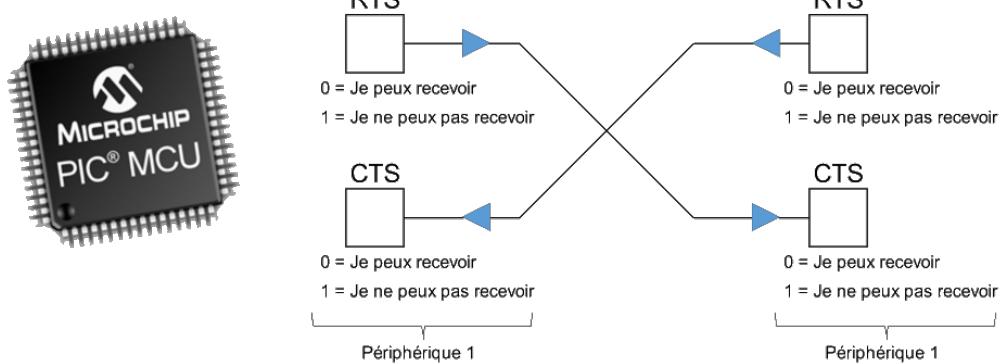
Ces deux lignes permettent au receveur et à l'émetteur d'alerter chaque côté de leur état. Un émetteur alerte sa ligne RTS, qui provoque une interruption sur le receveur,

ex : Puis-je envoyer des données ? Si le receveur est dans une position pour recevoir des données, il va alerter la ligne CTS,

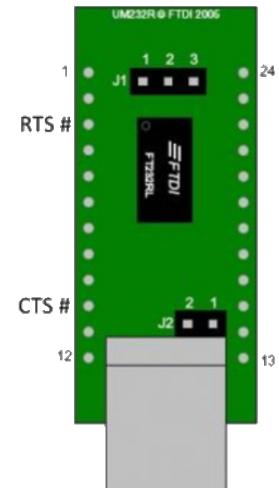
ex : Oui vous pouvez commencer à envoyer.

L'utilisation de ces lignes permet aux pilotes du périphérique qui implémentent un code de contrôle de flux matériel de maintenir une connexion de données fiable entre l'émetteur et le receveur. L'avantage de cette approche est que les données endommagées dans l'émission n'affectent pas le mécanisme de contrôle de flux.

FIGURE 3.4 – fonctionnement RTS CTS



source : [www.microchip.com](http://www.microchip.com)



source : [www.bravekit.com](http://www.bravekit.com)

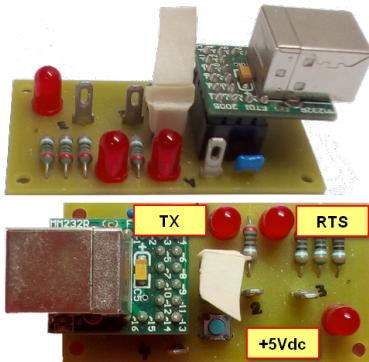
## 3.7 Description des tests

### 3.7.1 Les cartes de tests utilisées

#### Carte de test TX\_RX CLOSE LOOP

Cette carte de test est utilisée afin de valider le fonctionnement d'envoi et de réception en boucle fermée entre le Raspberry et le chip FTDI.

FIGURE 3.5 – Carte de test  
TX\_RX CLOSE LOOP

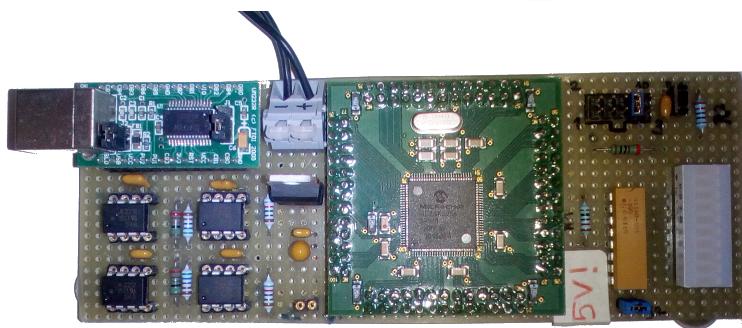


- LED TX : s'allume à chaque bit reçu égal à "0"
- LED RTS : s'allume quand le signal RTS est égal à "0"
- +5Vdc : alimentation 5Vdc présente

#### Carte de test d'envoi de trames PICBRIC\_SIMULATOR

- Cette carte permet de valider le fonctionnement du projet de Bachelor en envoyant la trame de validation.
- La programmation se fait à l'aide d'un programmeur en langage turbo pascal.
- Le soft de base a été développé par les apprentis électroniciens, supervisés par un ingénieur en électronique.
- Les sources sont disponibles si des modifications doivent être effectuées.

FIGURE 3.6 – Carte de test PICBRIC\_SIMULATOR



### 3.7.2 TEST\_D2XX\_RTS\_CTS

#### Description

La carte de test utilisée est la "TX\_RX CLOSE\_LOOP". La série de test ci-dessous consiste à valider le bon fonctionnement du driver D2xx.

Fonction implémentée	souhaité	Résultat
ClrRts()	Allume la led RTS	"REUSSI"
ClrRts()	Eteint la led RTS	"REUSSI"
WriteOneChar()	Ecrit 0xF2 dans le buffer , la led TX clignote	"REUSSI"
WriteReadOneChar()	Ecrit, Lit et affiche à la console le caractère 0xF2	"REUSSI"
WritePurgeReadOneChar()	Ecrit, purge les buffers TX et RX puis attend	"REUSSI"

#### Conclusion

*Tous les tests sont réussis, l'installation du driver, les fonctions et l'utilisation sont corrects.*

### 3.7.3 TEST\_D2XX\_RTS\_1

#### Description

La carte de test utilisée est la " PICBRIC\_SIMULATOR ".

Le but est de pouvoir arrêter l'envoi de trames selon les besoins.

#### Conclusion

Grâce à ce test nous avons pu dépanner la carte de simulation. Il y avait une erreur de configuration de registre dans le PIC. Le signal RTS envoyé par le chip FTDI au PIC arrivait physiquement mais n'était pas géré au niveau soft dans le microcontrôleur.

*Après correction, la fonction est validée.*

### 3.7.4 TEST\_D2XX\_RTS\_0

#### Description

La carte de test utilisée est la " PICBRIC\_SIMULATOR ".

Le but est de pouvoir autoriser l'envoi de trame selon les besoins.

#### Conclusion

*La fonction est validée*

### 3.7.5 TEST\_D2XX\_LIVE

#### Description

La carte de test utilisée est la " PICBRIC\_SIMULATOR ".

Le but est d'estimer le nombre de trames lues et le temps passé jusqu'à la première erreur.

Une trame de test est envoyée en boucle infinie à la vitesse de 1,3 Mb/s dès la mise sous tension de la carte. Un contrôle de cohérence est effectué sur l'entier des valeurs. S'il y a une incohérence, le test s'arrête et affiche le nombre de trames envoyées ainsi que le nombre de caractères perdus.

Le compteur CPT = MSB 1 « 8 | LSB ;

Les tests effectués sont :

1. SI (CPT de la trame précédente < 65535)  
ALORS (CPT = CPT de la trame précédente +1)  
SINON (CPT = 0)  
FIN SI;
2. MSB 1 = MSB 2 = MSB 3 = MSB 4
3. LSB 1 = LSB 2 = LSB 3 = LSB 4

#### Description de la trame de test

FIGURE 3.7 – TEST\_D2XX\_LIVE, trame de test

N°de bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Nbr de bits	8								8								8								8							
Définition	MSB compteur de trame								LSB compteur de trame								MSB compteur de trame								LSB compteur de trame							

N°de bit	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Nbr de bits	8								8								8								8							
Définition	MSB compteur de trame								LSB compteur de trame								MSB compteur de trame								LSB compteur de trame							

## Capture d'écran

FIGURE 3.8 – TEST\_D2XX\_LIVE, capture d'écran de résultat du test

```

trame number : 730547232
cpt error : 5

pi@raspberrypi:/ $ top -d 1
top - 07:33:15 up 12:38, 2 users,  load average: 1.42, 1.59, 1.59
Tasks: 177 total,  2 running, 175 sleeping,  0 stopped,  0 zombie
%Cpu(s): 14.9 us, 21.3 sy,  0.0 ni, 63.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  945520 total,  413604 used,  531916 free,   73300 buffers
KiB Swap: 102396 total,      0 used, 102396 free, 192276 cached Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1581 pi        20   0  55920  21188  17324 R  69.9  2.2 449:08.97 lxterminal
 1645 root      0 -20  45616  1492  1372 S 35.4  0.2 252:32.47 rts_cts_li+
 784 pi        20   0 29144 18248  5244 S 27.6  1.9 157:51.62 Xtightvnc
20392 root      20   0      0      0      0 S  3.9  0.0  0:25.75 kworker/u8+
20213 root      20   0      0      0      0 S  2.0  0.0  0:39.70 kworker/u8+
20716 root      20   0      0      0      0 S  2.0  0.0  0:08.17 kworker/u8+
 1661 pi        20   0  5112  2428  2020 R  1.0  0.3 11:42.70 top
  1 root      20   0 22800 3988  2792 S  0.0  0.4  0:03.81 systemd
  2 root      20   0      0      0      0 S  0.0  0.0  0:00.03 kthreadd
  3 root      20   0      0      0      0 S  0.0  0.0  0:14.11 ksoftirqd/0
  5 root      0 -20      0      0      0 S  0.0  0.0  0:00.00 kworker/0:+
  7 root      20   0      0      0      0 S  0.0  0.0  0:55.97 rcu_sched
  8 root      20   0      0      0      0 S  0.0  0.0  0:00.00 rcu_bh
  9 root      rt  0      0      0      0 S  0.0  0.0  0:00.02 migration/0
 10 root      0 -20      0      0      0 S  0.0  0.0  0:00.00 lru-add-dr+

```

## Conclusion

- Le test a été arrêté après 12h38min de fonctionnement.
- 730'547'232 trames ont été acquises et 5 caractères perdus lors de l'arrêt de l'enregistrement.
- Taux d'erreurs de  $8.5 \times 10^{-10}$ . Bien plus petit que la théorie qui est de  $10^{-6}$ .
- La charge du processeur est de 40%.
- La vitesse 1,3 Mb/s correspond à un envoi de trames à une vitesse de environ 20KHz. C'est le double du cahier des charges.

*La vitesse minimum définie dans le cahier des charges est validée mais sans application*

### 3.7.6 TEST\_D2XX\_LIVE\_VMAX

#### Description

La carte de test utilisée est la "PICBRIC\_SIMULATOR".

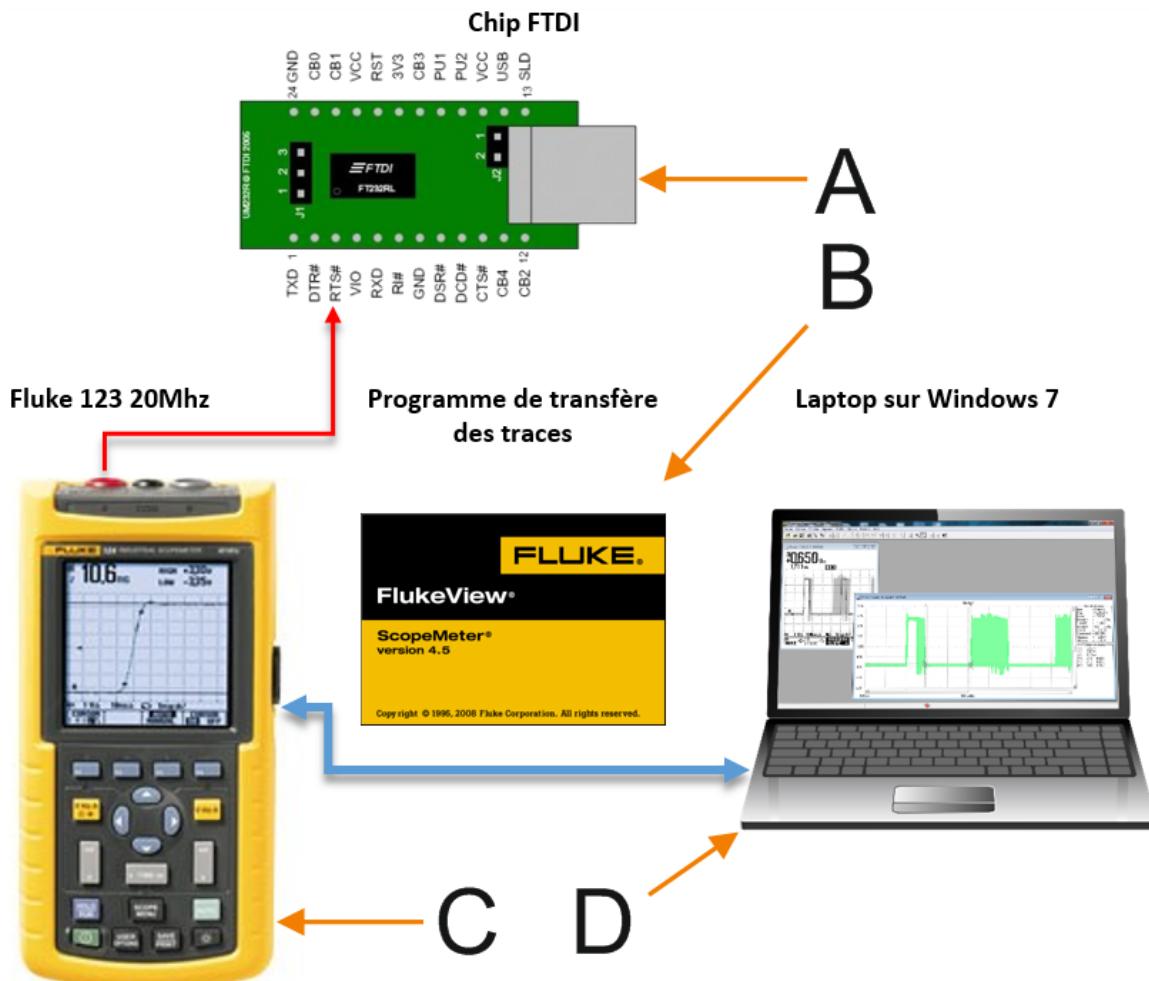
Le but est valider l'acquisition à la vitesse maximale et de mesurer le temps de cycle d'une lecture d'un buffer.

Le principe du test est le même que "TEST\_D2XX\_LIVE" mais à la vitesse de 2 Mb/s. C'est la vitesse maximale paramétrable sur la carte de simulation PICBRIC\_SIMULATOR en accord avec le chip FTDI. Le pas suivant serrait de 4 Mb/s ce qui est supérieur de 1 Mb/s à la spécification du chip.

Plusieurs mesures ont été effectuées à l'oscilloscope en mode enveloppe.

## Technique de mesure

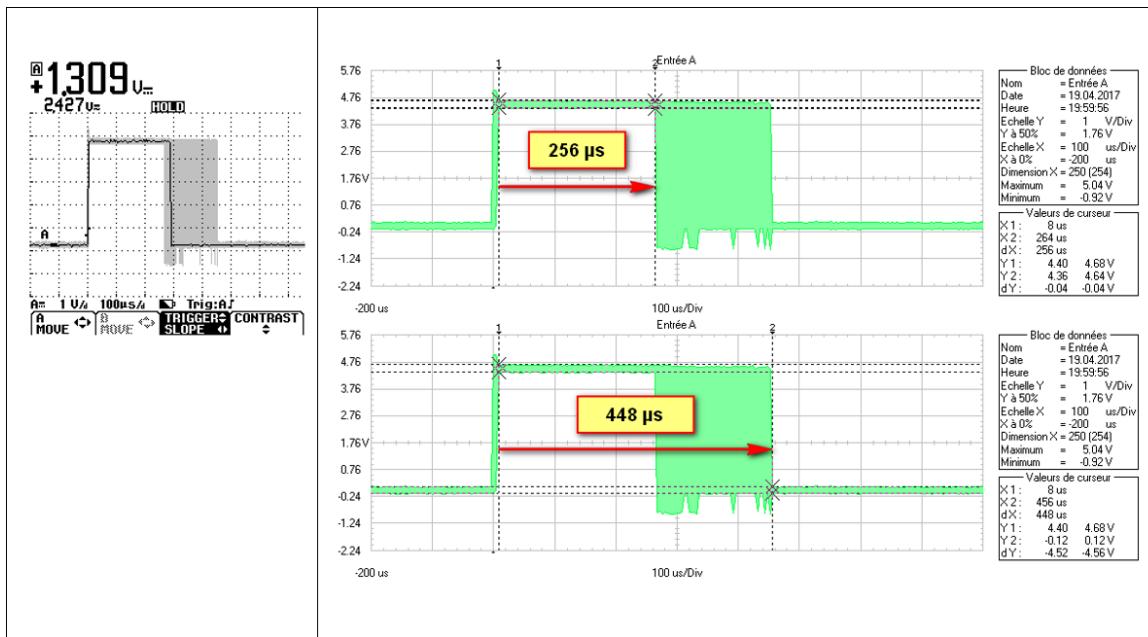
FIGURE 3.9 – TEST\_D2XX\_LIVE, technique de mesure du test



*source A : [www.bravekit.com](http://www.bravekit.com)  
 source B : [www.thietbisonganh.com](http://www.thietbisonganh.com)  
 source C : [en-us.fluke.com](http://en-us.fluke.com)  
 source D : [clipart.pd4pic.com](http://clipart.pd4pic.com)*

## Mesure du temps minimum et maximum d'une acquisition de n trames

FIGURE 3.10 – TEST\_D2XX\_LIVE, temps min et max d'une acquisition



## Valeurs théoriques

Débit :  $2\ 000\ 000 \text{ B s}^{-1}$ .

Temps d'émission par un byte :  $0.5 \mu\text{s}$ .

Temps d'émission **théorique** d'une trame :  $10 \text{ B} * 0.5 \mu\text{s} = 5 \mu\text{s}$ .

## Valeurs mesurée

Le temps d'acquisition varie de  $256 \mu\text{s}$  à  $448 \mu\text{s}$ .

Ces temps mesurés ne changent pas en fonction de la quantité de trame lue par acquisition. Il y a sûrement un buffer qui travaille au niveau du Chip USB du Raspberry.

Je ne peux pas en déduire le temps réel.

Pour valider la vitesse à la fin du projet, d'autres solutions devront être utilisées :

- Implémenter un timer soft qui mesure le temps d'acquisition en millisecondes.
- Utiliser les GPIO<sup>1</sup> pour sortir un signal à l'image de chaque acquisition et mesurable à l'oscilloscope.
- Utiliser un analyseur de trames.

Un analyseur serait disponible dans notre département d'analyses et mesures.

1. General Purpose Input/Output

### 3.7.7 Les sources de chez FTDI

- FTDI home page  
<http://www.ftdichip.com>
- Datasheet UM232R USB  
[http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS\\_UM232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_UM232R.pdf)
- Datasheet FT232R USB UART IC  
[http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf)
- FTDI Drivers Installation Guide for Linux  
[http://www.ftdichip.com/Support/Documents/AppNotes/AN\\_220\\_FTDI\\_Drivers\\_Installation\\_Guide\\_for\\_Linux.pdf](http://www.ftdichip.com/Support/Documents/AppNotes/AN_220_FTDI_Drivers_Installation_Guide_for_Linux.pdf)
- FTDI USB Data Transfer Efficiency  
[http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN\\_103\\_FTDI\\_USB\\_Data\\_Transfer\\_Efficiency\(FT\\_000097\).pdf](http://www.ftdichip.com/Support/Documents/TechnicalNotes/TN_103_FTDI_USB_Data_Transfer_Efficiency(FT_000097).pdf)
- Software Application Development, D2XX Programmer's Guide  
[http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX\\_Programmer's\\_Guide\(FT\\_000071\).pdf](http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer's_Guide(FT_000071).pdf)

## 4.0 Protocoles utilisés

### 4.1 La trame d'envoi

Elle n'a pas été définie pour ce projet car la carte d'acquisition définitive n'est pas finie au niveau du développement.

### 4.2 La trame de réception

La trame de réception a été clairement définie afin de pouvoir valider le projet, elle est de la taille de 64bits et se compose comme ci-dessous.

FIGURE 4.1 – Trame de réception

N°de bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Nbr de bits	8	8	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Définition	MSB compteur de trame	LSB compteur de trame	D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16
N°de bit	32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47	48 49 50 51 52 53 54 55
Nbr de bits	8	8	8
Définition	AI1	AI2	AI3 AI4
			56 57 58 59 60 61 62 63

## 4.3 La trame de validation

La trame de validation est la trame de réception mais remplie avec des données connues. Ceci afin de valider le système.

FIGURE 4.2 – Trame de validation

N°de bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	
Nbr de bits	8	8	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Nom entrée			DI1 DI2 DI3 DI4 DI5 DI6 DI7 DI8 DI9 DI10 DI11 DI12 DI13 DI14 DI15 DI16	
Définition	MSB compteur de trame	LSB compteur de trame	LSB compteur de trame	MSB compteur de trame

N°de bit	32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47	48 49 50 51 52 53 54 55	56 57 58 59 60 61 62 63			
Nom entrée	AI1		AI2		AI3		AI4
Nbr de bits	8		8		8		8
Définition	LSB CPT		LSB CPT >> 1		LSB CPT >> 2		Busy Counter

### 4.3.1 Résultat théorique attendu

- Les entrées digitales de 1 à 16 représentent l'état de chaque bit du compteur de trame.
- AI1 : l'image du LSB du compteur de trame.
- AI2 : AI1 avec un décallage de 1 vers la droite ( » 1).
- AI3 : AI1 avec un décallage de 2 vers la droite ( » 2).
- AI4 : Valeur d'attente entre deux lectures.

Ci-après, la représentation théorique des traces DI1 -> DI6, AI1, AI2, AI3. Les traces DI7 à DI16 continuent dans la même logique.

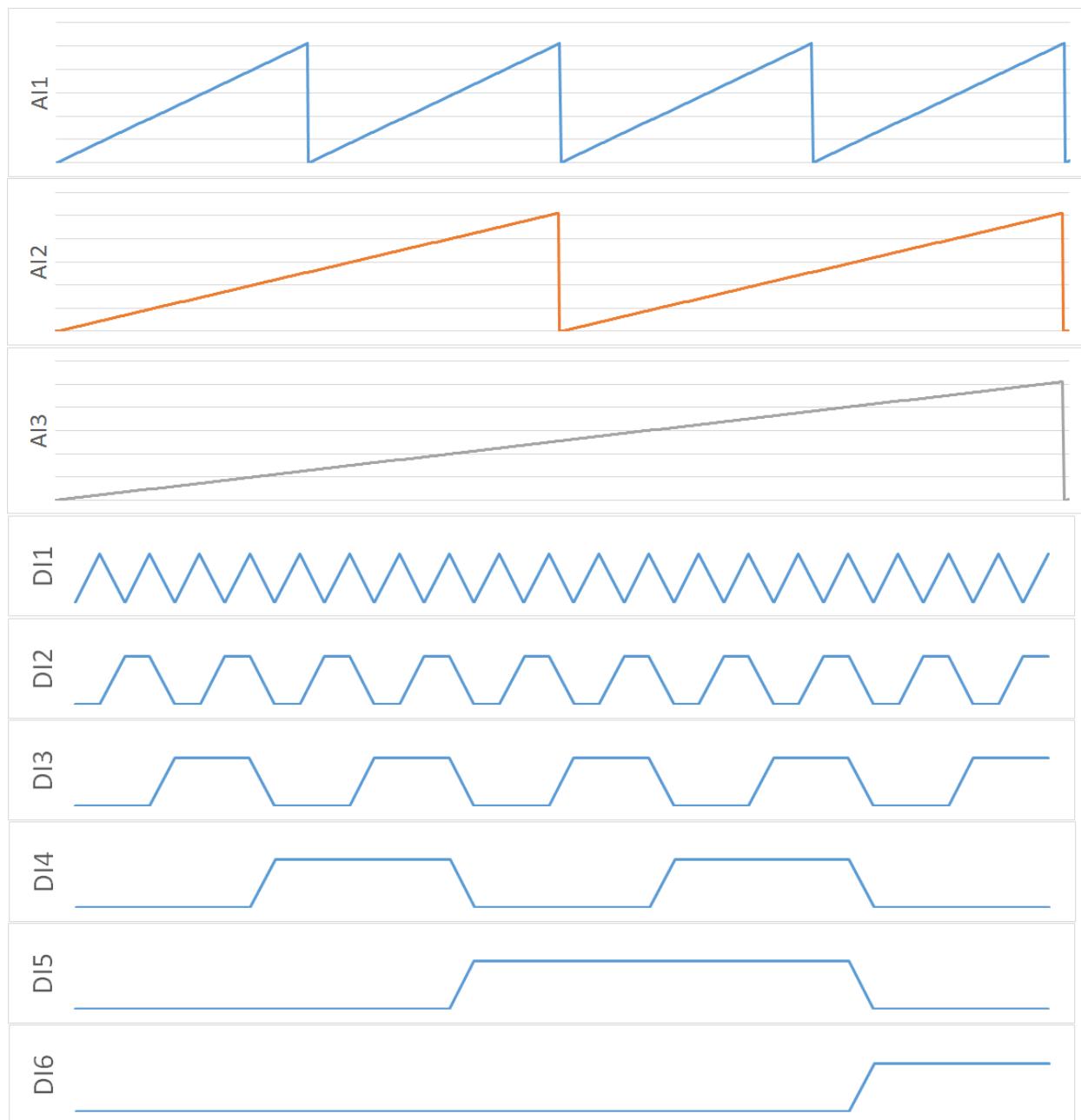


FIGURE 4.3 – Traces de validation



## 5.0 Le traitement des données

### 5.1 L'acquisition

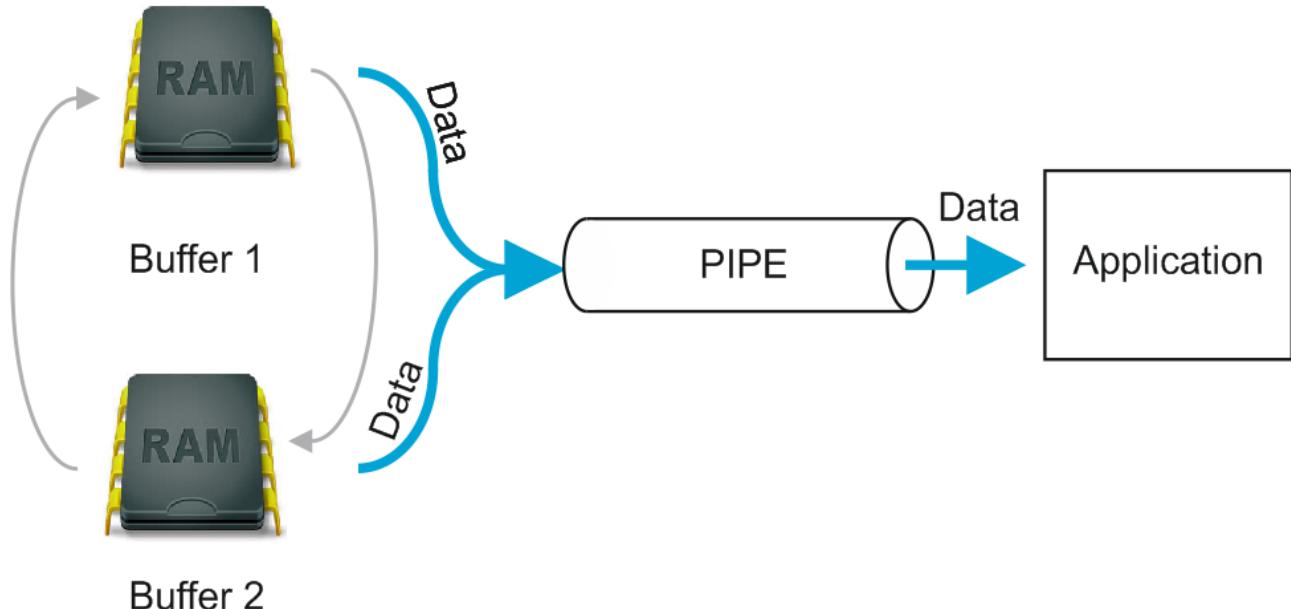
L'acquisition sera faite alternativement dans deux buffers en mémoire RAM par tranche de 1024 bits, ce qui correspond à 16 trames.

Chaque tranche "Data" est envoyée dans un flux "Pipe".

Pour la suite du traitement on prend des paquets d'une trame (64 bits) à la sortie du flux.

Toutes ces valeurs sont arbitraires, elles peuvent changer afin d'obtenir la vitesse d'acquisition la plus rapide possible. Mais restera un multiple de la taille de la trame.

FIGURE 5.1 – Principe d'acquisition des données



source : [www.malekal.com](http://www.malekal.com)

### 5.2 Le contrôle de la cohérence des trames

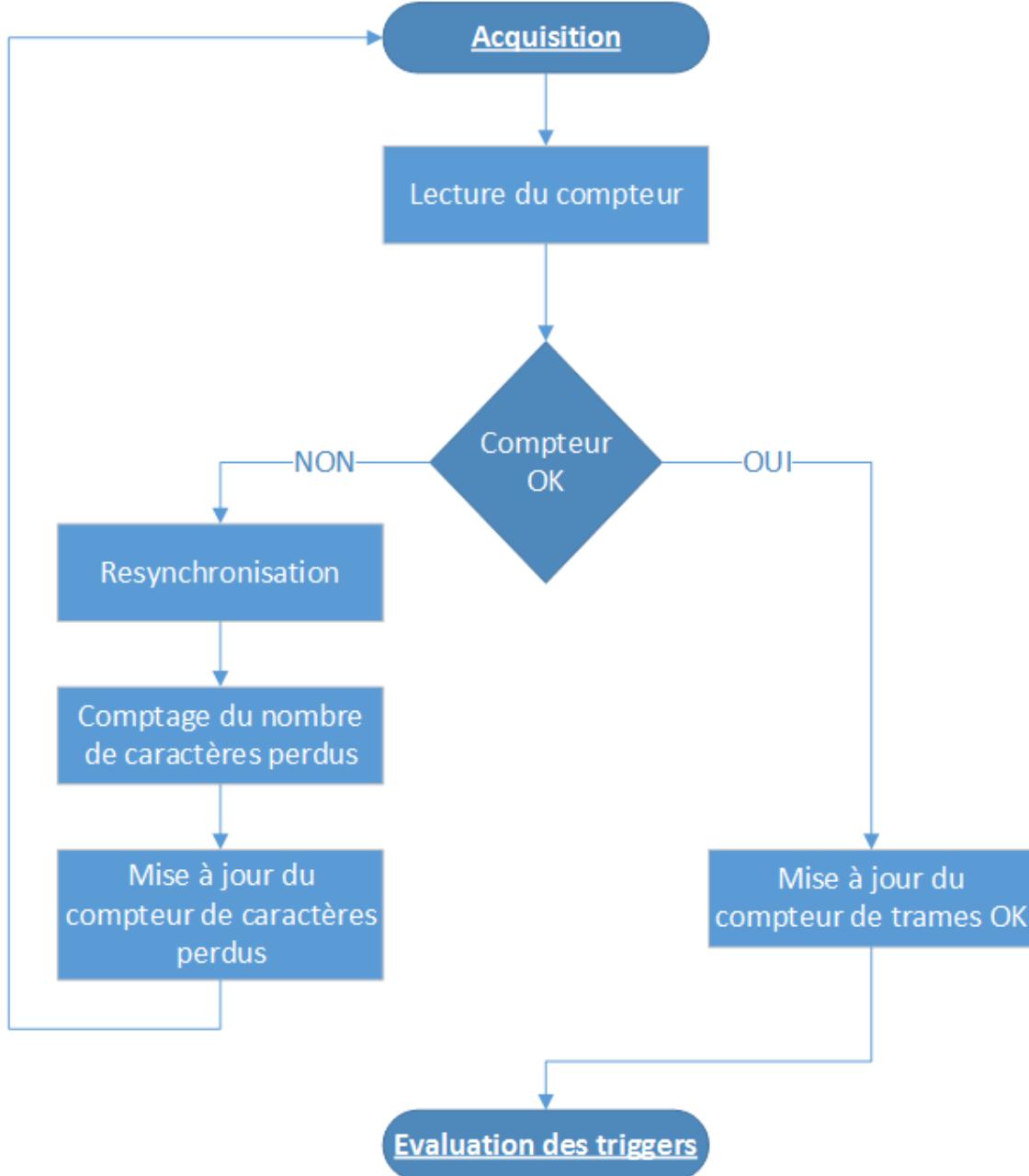
#### 5.2.1 Description

Le contrôle de la cohérence se fait uniquement sur le compteur de trame.

Toutes les autres valeurs sont inconnues et ne suivent aucune logique.

### 5.2.2 Le diagramme de flux

FIGURE 5.2 – Diagramme de flux du contrôle de cohérence



## 5.3 L'évaluation des triggers

### 5.3.1 Entrée digitale

La valeur de seuil est définie dans l'application puis envoyée vers la carte d'acquisition. Le protocole n'étant pas défini, cette option ne va pas être implémentée dans ce projet.

### 5.3.2 Entrée analogique

Le réglage se fait avec l'interface. Une seconde trace affiche la valeur logique du trigger dans le même graphique que la trace analogique.

Logique en pseudocode :

```
Si      (valeur AI < valeur du trigger)
ALORS  (etat_trigger = 0)
SINON   (etat_trigger = 1)
FIN SI;
```

## 5.4 L'évaluation de l'équation

- L'évaluation de l'équation sera faite avec un masque.
- Les masques seront définis et stockés dans un tableau.
- Pour chaque choix possible, un numéro de masque sera choisi et appliqué.
- Une description précise du calcul de masque est attachée en annexe
- Si le résultat de l'équation est "VRAI", alors on sauvegarde les données.



## 6.0 L'environnement de développement

### 6.1 Visual studio

Très rapide en développement et gratuit tant que je suis étudiant. Mais il ne peut être installé que sur un OS Windows pour l'instant.

La compilation pour un autre environnement demande une toolchain<sup>1</sup>, ceci n'est pas trivial à mettre en place.

Il a un outil de développement graphique très bien abouti.

Il a aussi un outil de création de diagramme de classes performant. Je vais l'utiliser pour développer celui de l'application.

### 6.2 QT

Environnement gratuit si je ne fais pas de commerce avec mon application.

Il est multiplateforme, ce qui me permet de l'installer autant sur le Raspberry que sur mon laptop en Windows. Cela rend possible la compilation en natif sur les deux environnements avec les mêmes sources. Cette possibilité m'évite d'avoir à implémenter une toolchain<sup>1</sup>.

Il a un outil de développement graphique assez facile à utiliser et, je pense, suffisant pour l'application à créer.

### 6.3 Mon choix

Mon choix sera l'environnement QT, car on n'est pas obligé d'avoir une toolchain et il est gratuit dans le cadre de mon développement. Le programme sera écrit en C++.

Pour l'industrialisation, une licence devra être achetée.

1. Outil permettant de faire de la compilation croisée entre différents systèmes d'exploitation.

## 6.4 Les phases de développement

Je vois 7 étapes de développement :

1. Installation, câblage et configuration du Raspberry Pi
2. Conception de l'affichage
3. Création du mode de simulation
4. Test et debug en mode simulation
5. Conception de l'acquisition
6. Validation de l'application
7. Test de durée

Les parties de 1 à 4 peuvent être développées en simulation.

Les autres doivent utiliser la carte de test d'envoi de trames PICBRIC\_SIMULATOR.

## 6.5 Les tests unitaires

L'évaluation de l'équation est la partie la plus critique du projet. Un bug n'est pas admissible. Je vais faire cette partie avec les tests unitaires afin de valider chaque masque indépendamment. La validation sera effectuée selon la table de vérité de chaque configuration d'équation possible.

## 6.6 La compilation

### 6.6.1 Les choix possibles

1. En natif<sup>2</sup> sur le Raspberry :

C'est plus long mais c'est une solution qui certifie la compatibilité de l'application.

2. Croisée :

On utilise la puissance de la machine où l'on développe pour la compilation et les fichiers compilés sont directement envoyés sur la cible qui doit être reconnue et connectée.

L'application peut être ensuite directement lancée depuis la cible.

Gain de temps de compilation mais il faut une toochain<sup>1</sup> qui fonctionne pour toutes les librairies utilisées. Peut être très long à mettre en place.

3. A distance :

Le développement se fait en natif<sup>2</sup>. Ensuite les fichiers sont transférés sur une machine distante pour la compilation. La compilation terminée, on récupère les fichiers de l'application. Gain de temps de compilation et permet de développer en natif. Mais demande du temps à être mis en place.

### 6.6.2 Conclusion

Je vais commencer avec la première solution, vu qu'une bonne partie du projet peut être simulée. A chaque fin d'étape, je vais faire l'essai sur le Raspberry afin de valider le fonctionnement.

Je pourrais estimer les différences de temps de compilation, ce qui me guidera tout le long du projet et me permettra d'estimer la meilleure solution à utiliser.

2. On parle de compilation native lorsque l'application est compilée sur une machine à la fois hôte et cible.  
Dans mon cas le Raspberry

1. Outil permettant de faire de la compilation croisée entre différents systèmes d'exploitation.



## 7.0 Le rafraîchissement de l'affichage

### 7.1 La persistance rétinienne

La persistance rétinienne est la capacité de l'œil (et du cerveau) à superposer une image déjà vue aux images que l'on est en train de voir. Elle résulte du temps de traitement biochimique des signaux optiques par la rétine et le cerveau. Elle est plus forte et plus longue si l'image observée est lumineuse.

Il existe deux types de persistance rétinienne :

1. La persistance **positive**, rapide, la couleur de l'image persiste durant environ 50 ms.
2. La persistance **négative**, plus longue due à une exposition prolongée à une forte intensité lumineuse. Une trace sombre de l'image persiste durant plusieurs secondes.

Cette propriété de l'œil est utilisée par le cinéma et la télévision pour donner l'impression d'un mouvement continu à partir d'une séquence d'images.

Ainsi, si l'on fait défiler très rapidement une séquence d'images, au rythme de 24 par seconde, l'œil a en permanence en mémoire les images et ne peut distinguer 2 images successives.

### 7.2 La fréquence de rafraîchissement d'un écran

La fréquence de rafraîchissement est importante afin d'avoir un confort visuel agréable.

Plus cette fréquence est élevée, meilleur est le confort.

La fréquence de rafraîchissement est définie par le nombre d'images s'affichant sur l'écran par seconde. Cette valeur varie généralement entre 50 et 165 Hz (sur certains écrans haut de gamme).

À partir de 60 Hz, l'œil humain observe une image fluide

Pour l'utilisation de la 3D il faudra opter pour un rafraîchissement de l'affichage à 120 Hz. La raison est simple : en 3D, l'écran renvoie deux images, une pour chaque œil. Dès lors, la fréquence de rafraîchissement doit être également doublée.

### 7.3 Le choix de la fréquence de rafraîchissement de l'écran

Le choix sera de 100 Hz, c'est 4 fois la persistance rétinienne et quasiment le double d'un écran standard. Avec cette fréquence, lors de l'acquisition à la vitesse maximum de 10 kHz, on rafraîchit toutes les 100 acquisitions de trames.

Il est possible de faire un rafraîchissement pour tous les points, mais cela serait inutile, car l'écran ne peut pas suivre et la charge du processeur risque d'augmenter jusqu'à la saturation.



## 8.0 L'interface graphique

### 8.1 Introduction

Une ancienne interface graphique a été développée par le département d'analyses et mesures du mandant. Quelques monteurs externes ont aussi été choisis afin de valider le concept.

Je vais m'inspirer de ceci afin d'avoir un design similaire et être proche de la philosophie choisie à l'époque de sa création (il y a environ 10 ans).

Cette interface avait été développée avec le programme LabView.

Je vais partir avec QT Creator comme environnement de développement, le design est plus moderne, actuel et le code est en C++. Je dois trouver un maximum de Widgets<sup>1</sup> existants et semblables afin de gagner du temps de développement.

### 8.2 Le choix des widgets

#### 8.2.1 Affichage de texte non modifiable par l'utilisateur

La classe QLabel sera utilisée. Très simple à implémenter.

#### 8.2.2 Boutons avec deux ou moins de sélections (type ON/OFF)

Avec un maximum de deux choix, la classe QPushButton me semble intuitive, il est possible de tourner en boucle sur les deux choix avec un simple appui. Simple à implémenter et simple à utiliser.

#### 8.2.3 Boutons avec plus de deux sélections

Avec plus de deux choix, la classe QComboBox me semble la plus efficace afin de pouvoir directement sélectionner le bon mode. L'implémentation est un peu plus longue, mais l'utilisation sera plus simple.

#### 8.2.4 Curseur horizontal

Il sera de la classe QSlider, de type horizontal.

1. Élément de base de l'interface graphique d'un logiciel.

### 8.2.5 Les traces

Les traces<sup>2</sup> seront de classe QcustomPlot, c'est une classe open source et dédiée à l'affichage de traces. La configuration risque d'être un point difficile mais néanmoins possible.

Comme 2<sup>ème</sup> alternative, il y a QWT, qui est un librairie libre mais nécessite une installation. Par conséquent il est possible d'avoir des problèmes de compatibilité.

---

2. Représentation du signal sur un graphique.

## 9.0 L'ancienne interface utilisateur

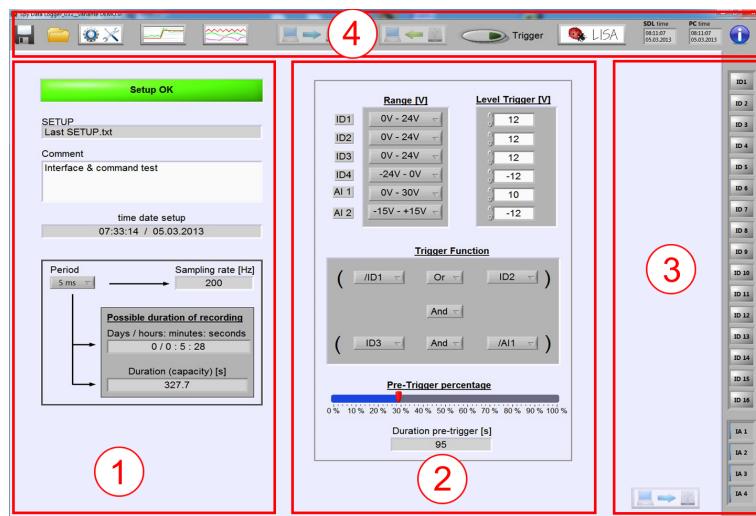
### 9.1 Les différentes pages

Quatre pages de visualisation avaient été développées :

- Page principale, elle permet de configurer l'appareil.
- La visualisation des triggers.
- La visualisation des traces.
- Le menu de développement et de debug.

#### 9.1.1 La page principale

FIGURE 9.1 – Ancienne interface, la page principale



#### Descriptif du fonctionnement

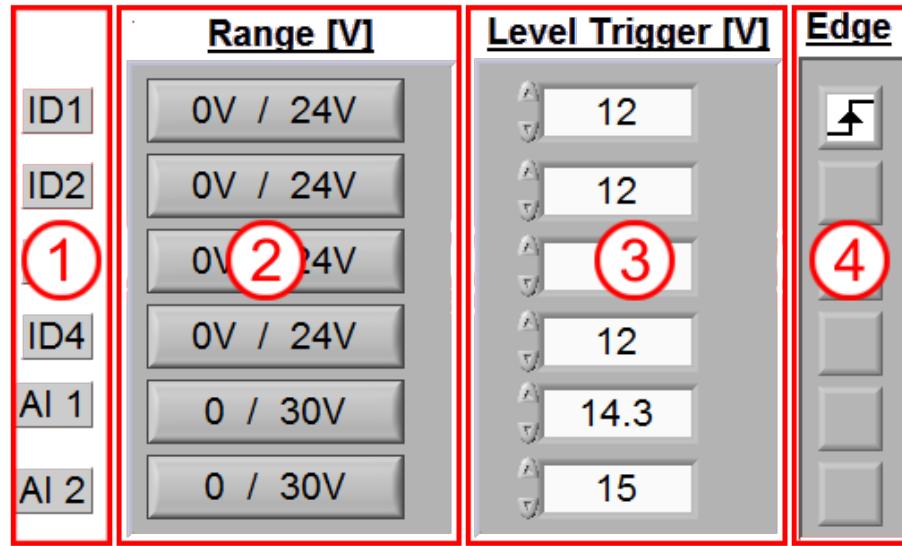
- ① Réglage de la vitesse d'acquisition et calcul du temps d'enregistrement.
- ② La configuration des différents triggers, de l'équation logique, de la sauvegarde et de l'ajustement du pré-trigger. Le descriptif précis décrit plus loin.
- ③ Visualisation de tous les signaux, je ne suis pas sûr de son utilité. À mon avis trop petit pour être utilisable.
- ④ L'onglet de navigation. Je pense rester sur ce design, car en cas d'écran tactile les boutons sont toujours plus pratiques que des menus déroulants.

#### 9.1.2 Configuration des différents trigger

1. Changement d'état d'un signal.

## Descriptif du fonctionnement

FIGURE 9.2 – Ancienne interface, configuration de la fonction de trigger



**1** Nom des différents canaux de triggers utilisables. Ce titre est aussi utilisé afin de pouvoir les sélectionner dans l'équation logique.

- 1<sup>ère</sup> lettre, type de canal : **D** : digital    **A** : analogique
- 2<sup>ème</sup> lettre, défini si c'est une entrée ou sortie : **I** : Input   **O** : Output
- Suivi du numéro de l'entrée.

**2** Sélection des tensions de la plage de mesure de tension.

- Pour les canaux numériques un seul choix possible : 0V ... 24V
- Pour les canaux analogiques deux choix possibles : 0V ... 30V, -15V ... +15V

**3** Sélection du niveau de trigger.

- En **dessous** : La condition de trig **n'est pas remplie**
- **Egal** ou **en-dessus** : La condition de trig **est remplie**

**4** Détection du flanc.

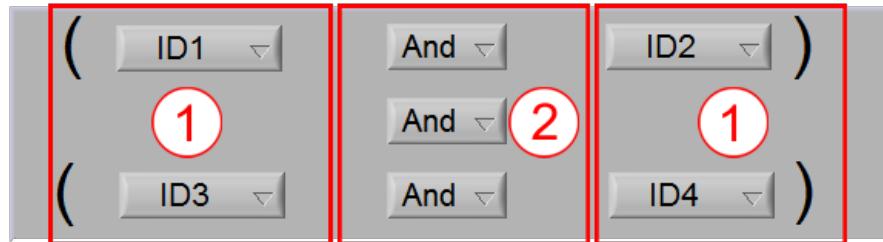
- Flèche montante : flanc<sup>1</sup> montant
- Flèche descendante : flanc<sup>1</sup> descendant

### 9.1.3 Configuration de l'équation logique

#### Descriptif du fonctionnement

Grâce à cette interface, il est possible de choisir l'équation logique pour remplir les conditions de trigger. Si le résultat de l'équation est égal à "1", alors la condition de trigger est vraie et on mémorise les valeurs.

FIGURE 9.3 – Ancienne interface, configuration de l'équation logique



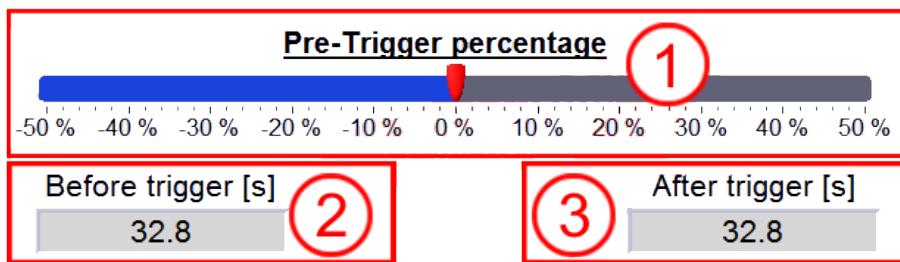
- ① Sélection des canaux de trigger. 7 choix possibles : NONE, DI1, DI2, DI3, DI4, AI1 et AI2.
- ② Sélection de l'opérateur logique : AND, OR, AND NOT, OR NOT.

### 9.1.4 L'ajustement du pré-trigger

#### Descriptif du fonctionnement

Si la condition de trig est vraie, alors la mémorisation des traces se fait en fonction de l'ajustement du pré-trigger choisi.

FIGURE 9.4 – Ancienne interface, l'ajustement du pré-trigger



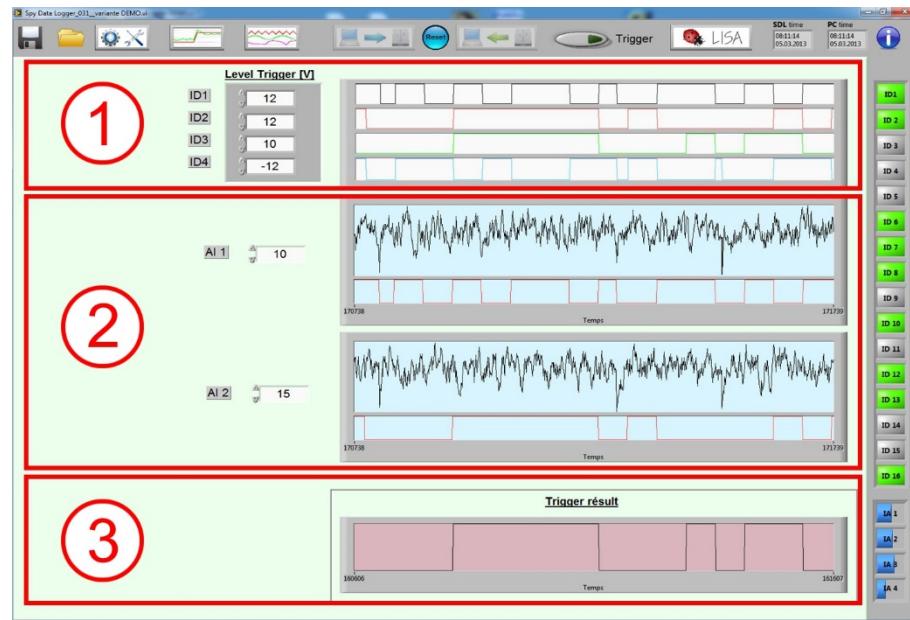
- ① Sélection du pourcentage de mémorisation entre avant et après le trig. Le nombre total de trames est défini à 65'536 (un tour de compteur) selon le cahier des charges ce qui correspond à 100% du bargraphe<sup>2</sup>.
- ② "Before trigger" correspond au temps que l'on garde avant le trig.
- ③ "After trigger" correspond au temps d'enregistrement restant après le trig.

2. Indicateur visuel relatif d'un signal quelconque. Ici en %

### 9.1.5 L'affichage des triggers et de l'équation logique

#### Descriptif du fonctionnement

FIGURE 9.5 – Ancienne interface, l'affichage des triggers et de l'équation logique



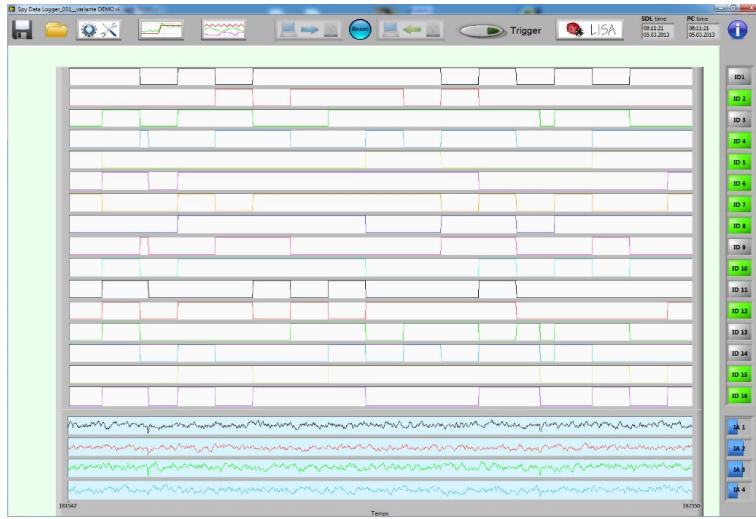
- 1** Contrôle et réglage visuel des triggers sur les traces numériques DI1, DI2, DI3, DI4.
- 2** Contrôle et réglage visuel des triggers sur les traces analogiques AI1, AI2, AI3, AI4.
- 3** Visualisation du résultat de l'équation.

### 9.1.6 L'affichage des traces

#### Descriptif du fonctionnement

Affichage de toutes les traces numériques et analogiques en même temps et en temps réel.

FIGURE 9.6 – Ancienne interface, l'affichage des traces

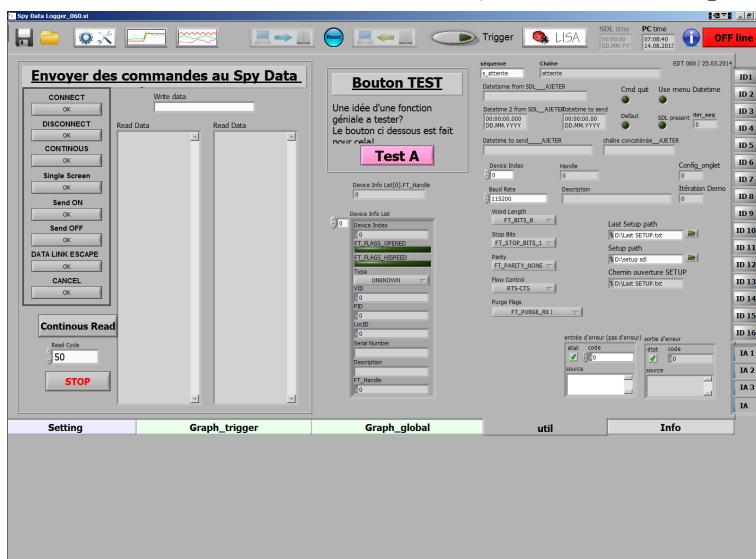


### 9.1.7 Le menu de debug

#### Descriptif du fonctionnement

Ce menu est utilisé pour valider les fonctions indépendamment.

FIGURE 9.7 – Ancienne interface, le menu de debug





# 10.0 Conception d'une Interface Homme Machine "IHM"

## 10.1 Introduction

Une nouvelle interface a été développée pour nos machines de dernière génération. Je vais m'en inspirer afin d'avoir une ligne graphique proche et une bonne compatibilité pour les futurs utilisateurs.

Je vais me baser sur une résolution fixe de HDTV<sup>1</sup> (1280 x 720), elle est compatible avec tous les moniteurs existant depuis 2010, cela m'évite de gérer le redimensionnement automatique dans un premier temps. Cette fonction pourra être implémentée plus tard lorsque le concept sera validé par les utilisateurs, mais les widgets utilisés devront être déjà prêts à l'accepter.

J'ai utilisé le programme "Corel Draw X5" pour dessiner la maquette de la future interface. Grâce à ce programme, j'ai directement travaillé en pixels afin de pouvoir créer toutes les pages nécessaires au développement en format définitif.

## 10.2 Les bonnes pratiques

1. Regrouper les éléments liés entre eux en utilisant des boîtes de groupe.
2. Mettre en majuscules uniquement la première lettre dans les étiquettes.
3. Ne pas mettre de caractère deux-points à la fin des étiquettes des contrôles ou des boîtes de groupe.
4. Eloigner les actions dangereuses des actions sans incidence.
5. Utiliser toujours un QButtonBox (boîte à boutons) pour les boutons 'Ok', 'Annuler', etc...
6. Les onglets ne doivent pas être imbriqués.
7. Les empilements de contrôles de formulaire doivent être évités le plus possible.
8. Eviter d'utiliser des termes techniques et utiliser plutôt des équivalents simples.
9. Utilisez une iconographie cohérente.
10. Placer les longues listes de contrôles dans des boîtes à défilement (scroll). Aucune boîte de dialogue ne doit mesurer plus de 580 pixels de haut et 1000 pixels de large.
11. Séparer les options avancées des fonctions basiques
12. Ne pas ajouter d'option dans le seul objectif d'avoir de nombreuses options.
13. Si un bouton doit ouvrir une nouvelle boîte de dialogue, des points de suspension (...) doivent être ajoutés en suffixe du texte du bouton.

*sources : [www.qgis.org/fr/site/getinvolved/development/qgisdeveloperguide/hig.html](http://www.qgis.org/fr/site/getinvolved/development/qgisdeveloperguide/hig.html)*

1. High-Definition Multimedia Interface

## 10.3 Méthodologie pour la conception d'IHM

Les IHM ne sont pas conçus au petit bonheur la chance, mais nécessitent une méthodologie précise pour aboutir à un résultat de qualité.

Il faut distinguer deux aspects de la conception d'IHM<sup>2</sup>.

- **L'ergonomie.**

Se construit sur la base de l'analyse de l'activité et la connaissance des IHM<sup>2</sup>, s'intéresse à la fonction par rapport aux conditions d'utilisation

- **le design.**

Donne la forme et permet l'industrialisation

### 10.3.1 Les trois domaines de connaissances sur lesquelles la méthodologie se base

- **l'activité ou les usages réels.**

Cette connaissance se base sur l'observation en situation et sur l'analyse des traces de l'activité.  
L'important étant ce que les utilisateurs font et non ce qu'ils disent ou pensent faire

- **le facteur humain.**

Les connaissances sur le facteur humain sont issues de nombreuses sources : médecine, psychologie, neurosciences, cognition. Elles forment une base de connaissances importante sur les capacités physiques et psychologiques des utilisateurs.

- **les IHM<sup>2</sup>.**

Interface Homme-Machine, il doit être simple, intuitif et agréable pour l'utilisateur.

FIGURE 10.1 – Visibilité d'une interface homme machine



source : Règles d'ergonomie et design des interfaces logicielles IHM, par Anne Pellegrin

## 10.4 Les étapes de la conception

### 1. Analyse de l'activité.

La connaissance des usages est nécessaire pour faire des choix aux étapes suivantes. Sans cette première étape, il y a un fort risque de proposer un niveau de complexité inadaptée par rapport à la tâche à faire.

### 2. Choix d'un type d'interface.

Le choix entre les différents types d'interface est assez restreint. Il dépend des possibilités d'interactions (clavier, télécommande,...), des usages et éventuellement des fonctions attendues.

### 3. Conception des guidelines de l'interface utilisateur.

C'est une étape essentielle, mais souvent négligée car elle nécessite une forte expertise et des métas-connaissances (connaissances sur les connaissances). Le guideline sera la référence pour la conception des services.

### 4. Conception des services.

C'est la phase de production. Elle aboutit généralement à la rédaction de Story-boards ou de spécifications détaillées des interfaces.

### 5. Validation.

Tests utilisateurs, Mise en production, Version bêta... Les retours des utilisateurs permettent de corriger et de faire évoluer les services existants.

*Il est important de respecter ces différentes étapes si l'on souhaite obtenir un résultat de qualité. Les trois domaines de connaissances interviennent dans chacune des étapes avec plus ou moins d'importance.*

## 10.5 Les sources

- Bonnes pratiques pour l'IHM  
<https://www.qgis.org/fr/site/getinvolved/development/qgisdevelopersguide/hig.html>
- Méthodologie pour la conception d'IHM  
<http://blocnotes.iergo.fr/concevoir/pour-la-tv/methodologie-pour-la-conception-d% E2%80%99ihm/>
- Règles d'ergonomie et design des interfaces logicielles IHM, par Anne Pellegrin  
<http://en.calameo.com/read/0001310816f2166c1ea6f>

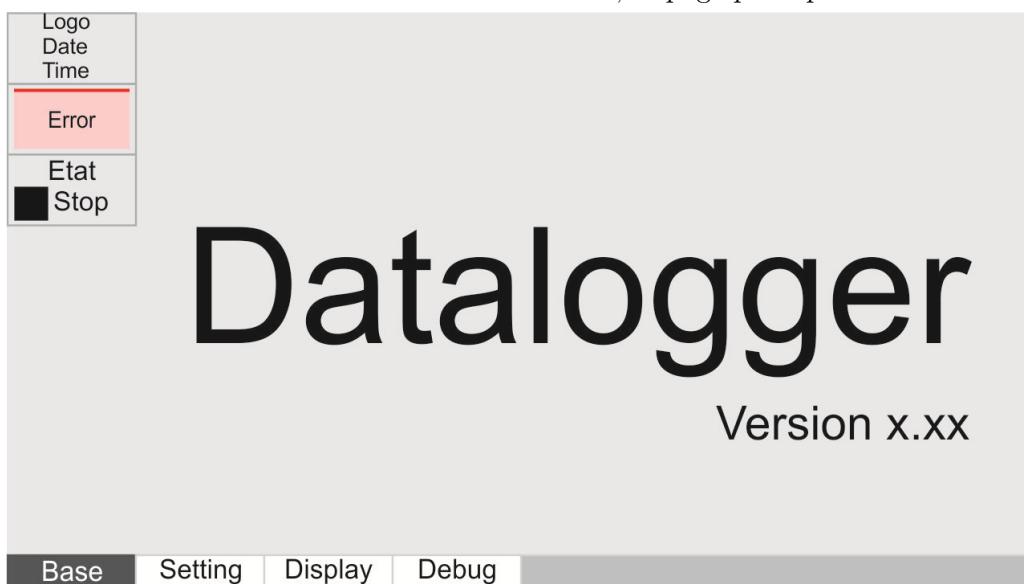


## 11.0 La nouvelle interface

### 11.1 La page principale

#### 11.1.1 Aperçu

FIGURE 11.1 – Nouvelle interface, la page principale



#### 11.1.2 Description

FIGURE 11.2 – Nouvelle interface, la page principale, description

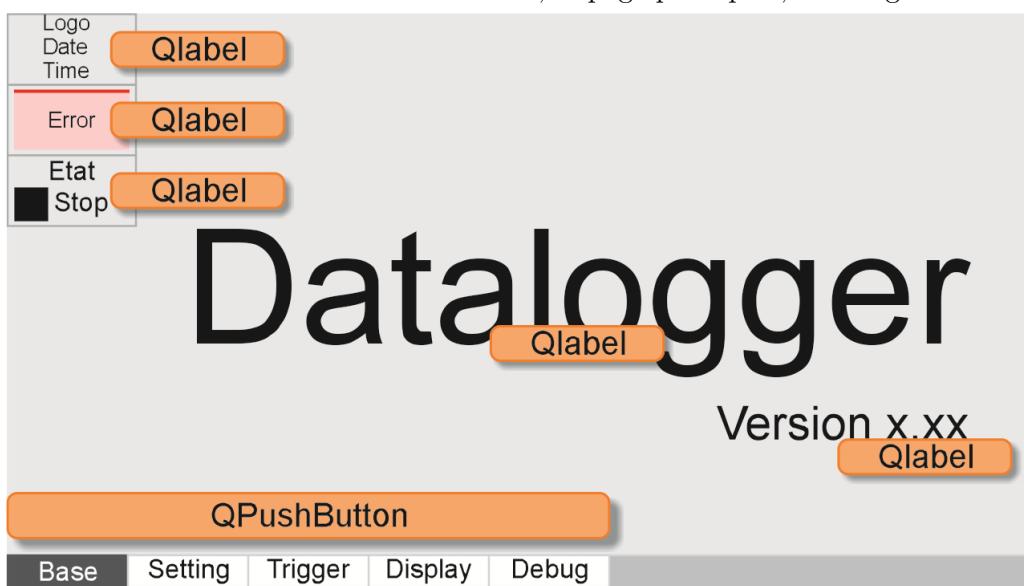


Les 5 états possibles :

- **Stop** : L'appareil est stoppé.
- **Pause** : L'affichage est figé, mais l'acquisition continue.
- **Run trig** : L'appareil est en mode d'analyse avec les triggers actifs.  
La trace se dessine de gauche à droite.
- **On trig** : L'appareil a triggé et sauvé les données. L'acquisition est stoppée.
- **Roll on** : L'appareil est en mode d'analyse sans les triggers.  
La trace se dessine de droite à gauche.

### 11.1.3 Les widgets pressenti

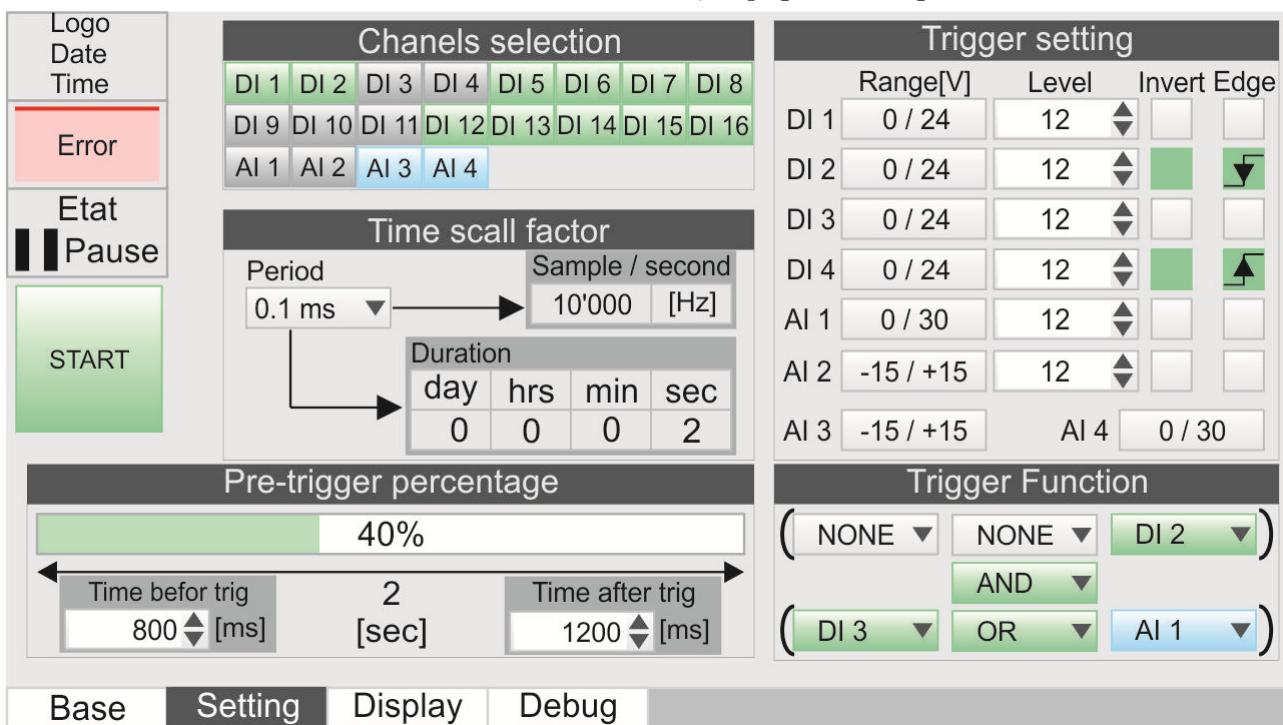
FIGURE 11.3 – Nouvelle interface, la page principale, les widgets



## 11.2 La page de configuration

### 11.2.1 Aperçu

FIGURE 11.4 – Nouvelle interface, la page de configuration



### 11.2.2 Description

Vu la faible résolution choisie, la page est bien remplie. Il me semble que toutes les configurations possibles sont accessibles.

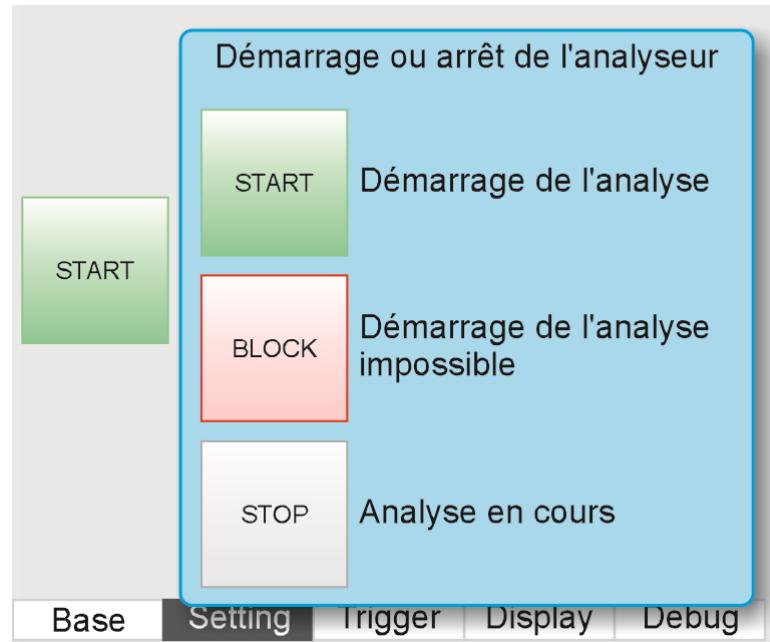
Je verrai à la manipulation si c'est intuitif ou pas. La validation sera effectuée avec plusieurs collègues afin d'avoir une vue extérieure.

Dans cette capture, j'ai mis toutes les options possibles selon le choix.

Si le système est en mode analyse, tous les onglets de configuration seront grisés et le texte sur le bouton changera à "STOP".

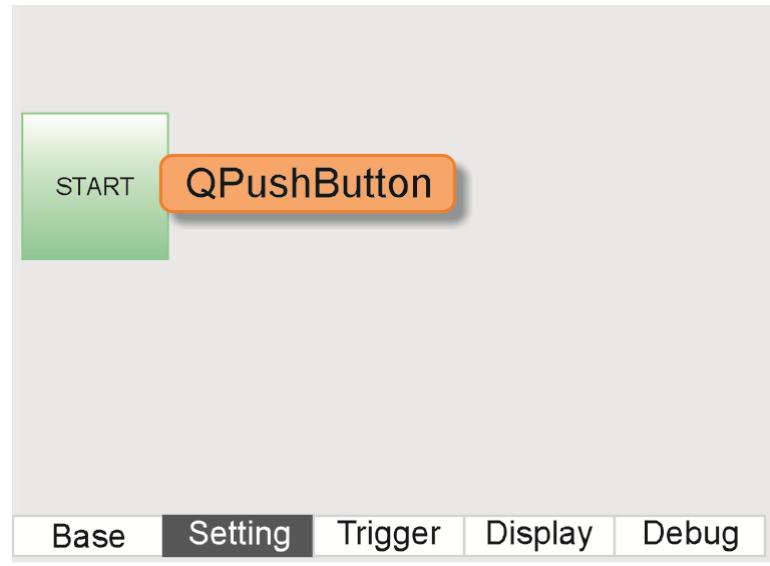
### 11.2.3 Description du bouton start

FIGURE 11.5 – Nouvelle interface, la page de configuration , description du bouton start



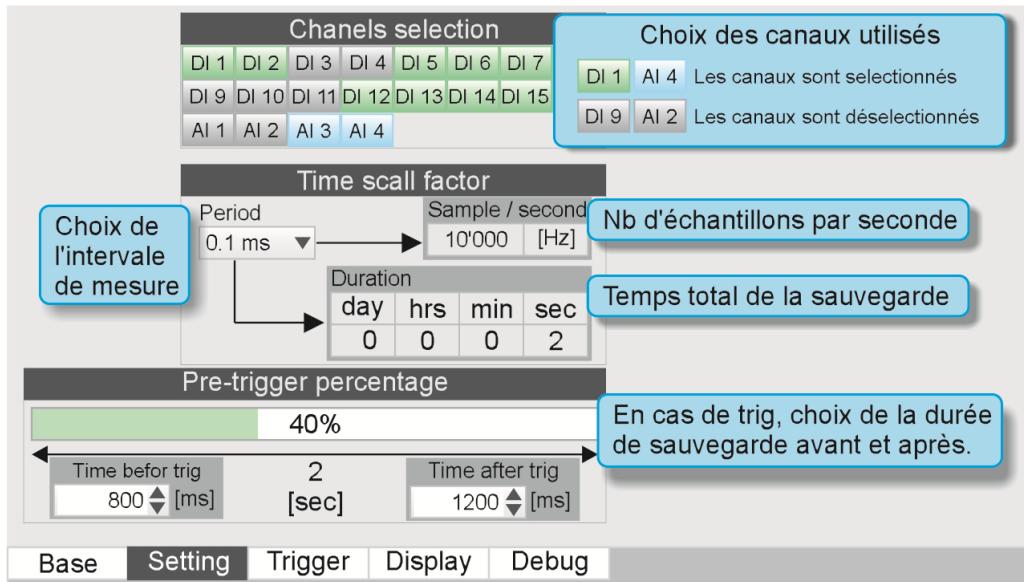
### 11.2.4 Le widget pressenti du bouton start

FIGURE 11.6 – Nouvelle interface, la page de configuration , widgets pressentis du bouton start



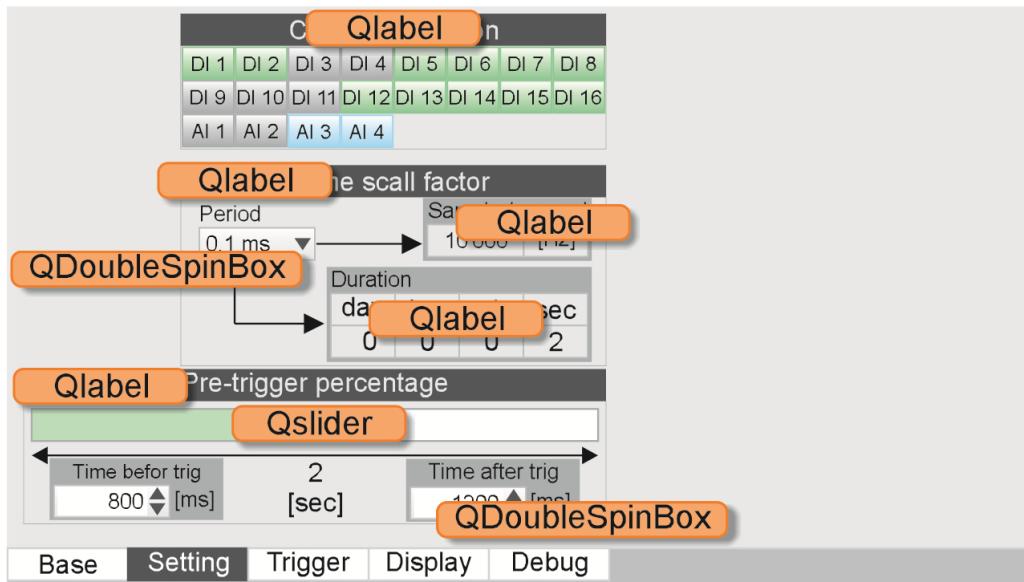
### 11.2.5 Description de la moitié de gauche

FIGURE 11.7 – Nouvelle interface, la page de configuration , description de la moitié de gauche



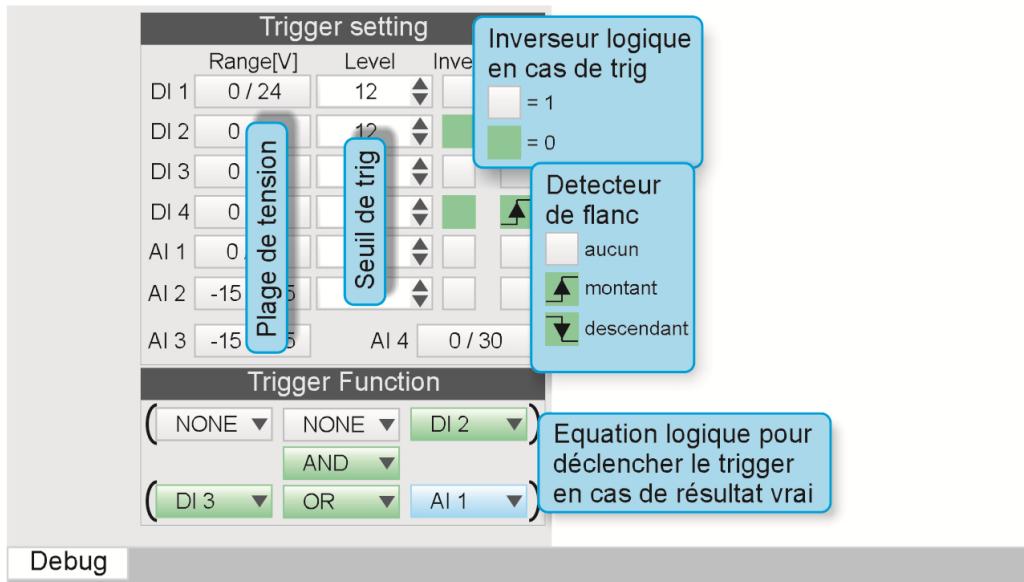
### 11.2.6 Les widget pressentis de la moitié de gauche

FIGURE 11.8 – Nouvelle interface, la page de configuration , widgets presentis de la moitié de gauche



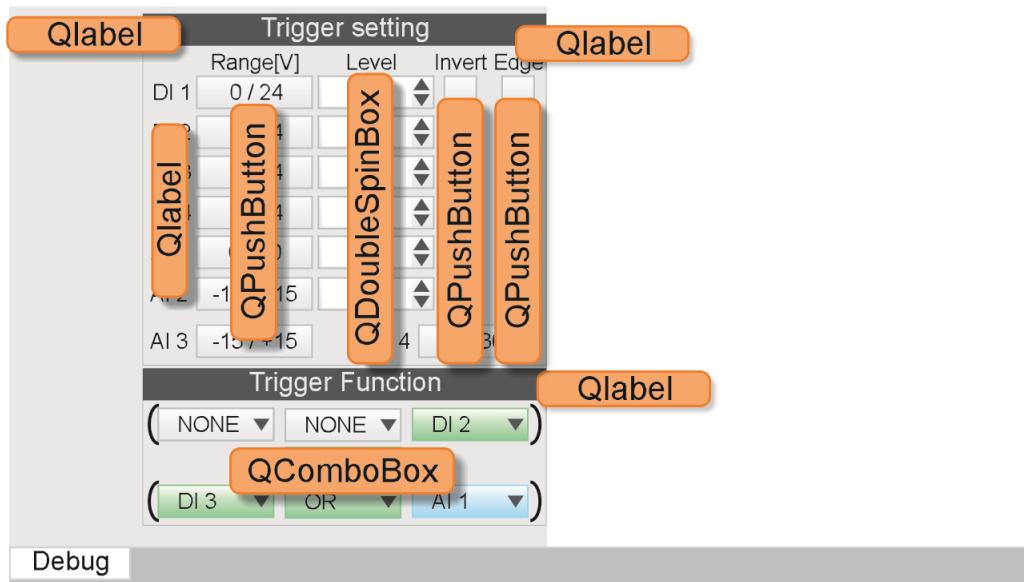
### 11.2.7 Description de la moitié de droite

FIGURE 11.9 – Nouvelle interface, la page de configuration , description de la moitié de droite



### 11.2.8 Les widgets pressentis de la moitié de droite

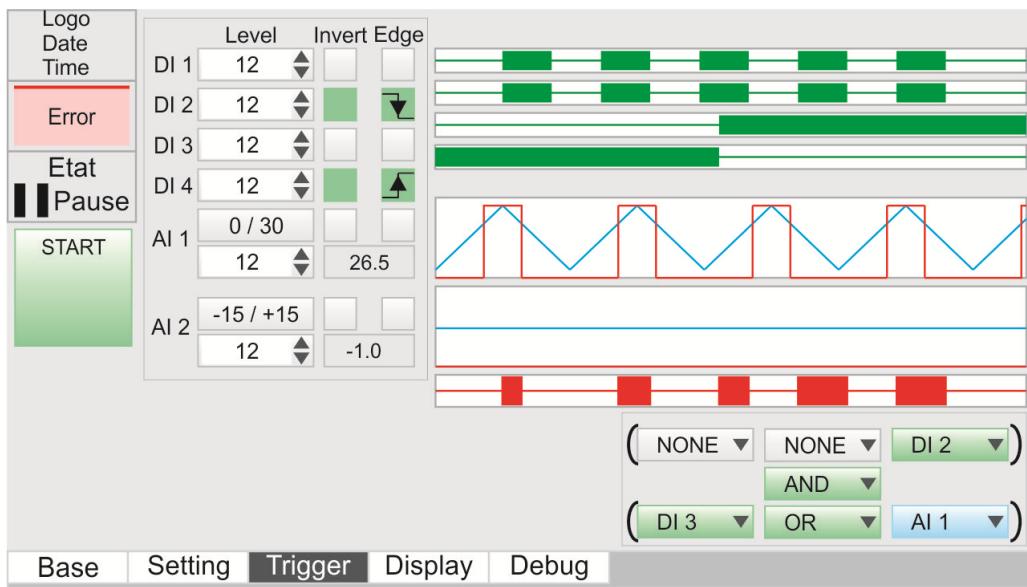
FIGURE 11.10 – Nouvelle interface, la page de configuration , widgets de la moitié de droite



## 11.3 La page de visualisation des triggers et de l'équation logique

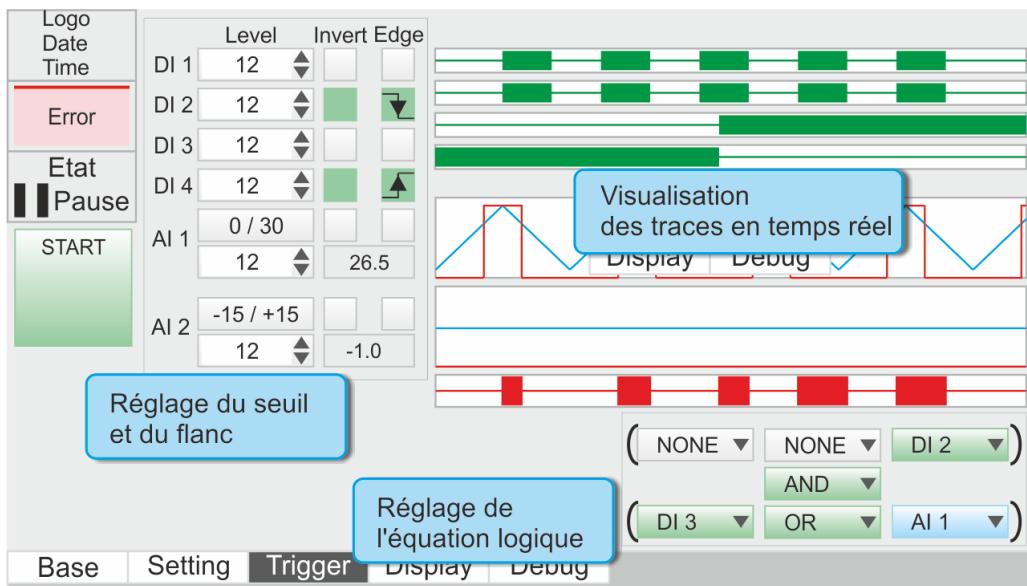
### 11.3.1 Aperçu

FIGURE 11.11 – Nouvelle interface, la page des triggers et de l'équation logique, l'aperçu



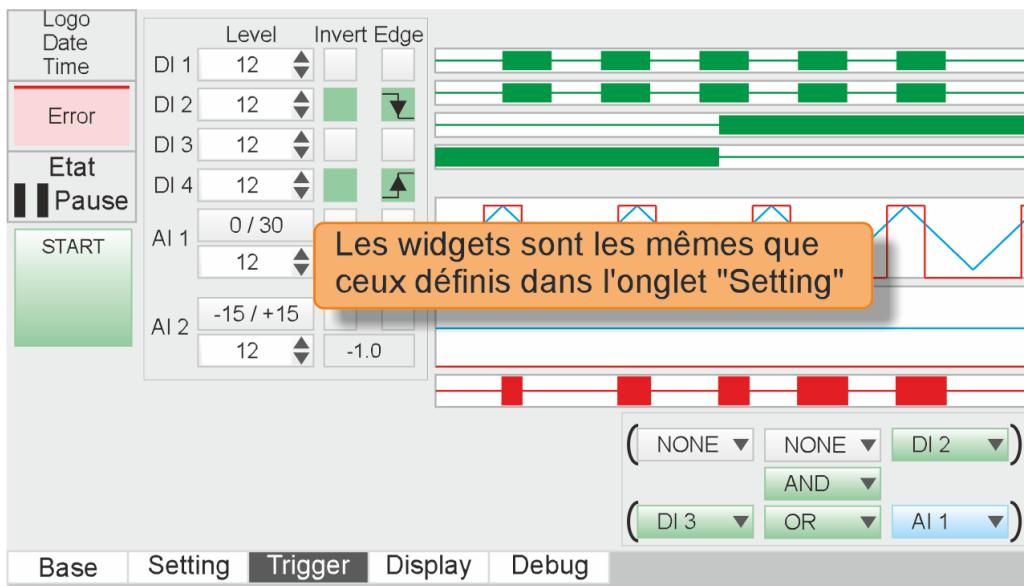
### 11.3.2 Description

FIGURE 11.12 – Nouvelle interface, la page des triggers et de l'équation logique, la description



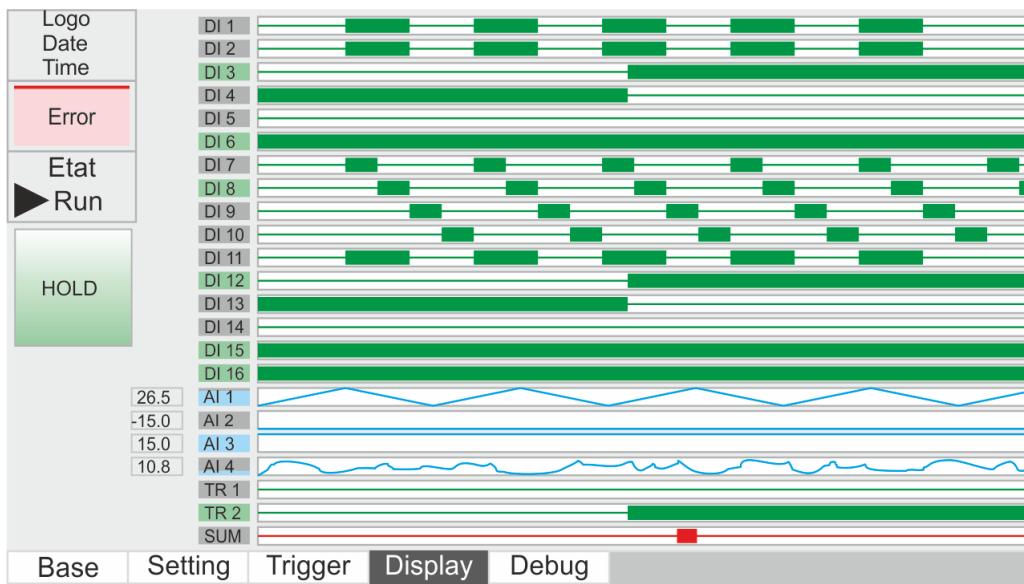
### 11.3.3 Les widgets pressentis

FIGURE 11.13 – Nouvelle interface, la page des triggers et de l'équation logique, les widgets



## 11.4 La page de visualisation

FIGURE 11.14 – Nouvelle interface, la page de visualisation



### 11.4.1 Description

Dans cette page, il est possible de visualiser toutes les traces en même temps. Je ne pense pas que cette option sera beaucoup utilisée, car il faut avoir une très bonne vision analytique pour traiter tous ces signaux en même temps. Mais de mon côté je valide la faisabilité avec la résolution choisie.

Si le technicien a désélectionné des traces, la place libérée sera en priorité pour les traces analogiques.

La visualisation de type jeton, comme dessinée, est très claire.

Les couleurs de signaux sont arbitraires mais je les trouve facile à lire.

Le bouton "HOLD" aura trois couleurs possibles :

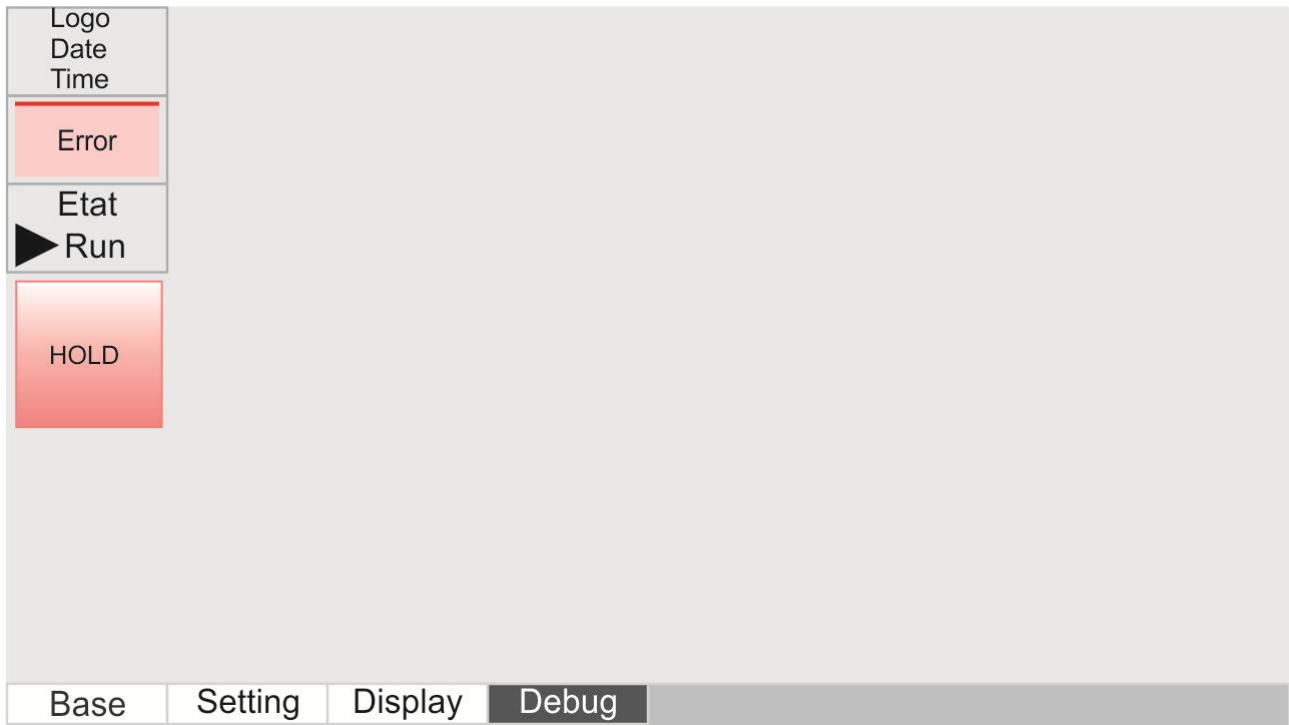
- "Vert" : le système est prêt à être figé, l'état est "Run".
- "Gris" : le système est figé, l'état est "On trig".

### 11.4.2 Les widgets présentis

Toutes les traces seront dérivées de la classe QCustomPlot.

## 11.5 La page de debug

FIGURE 11.15 – Nouvelle interface, La page de debug



### 11.5.1 Description

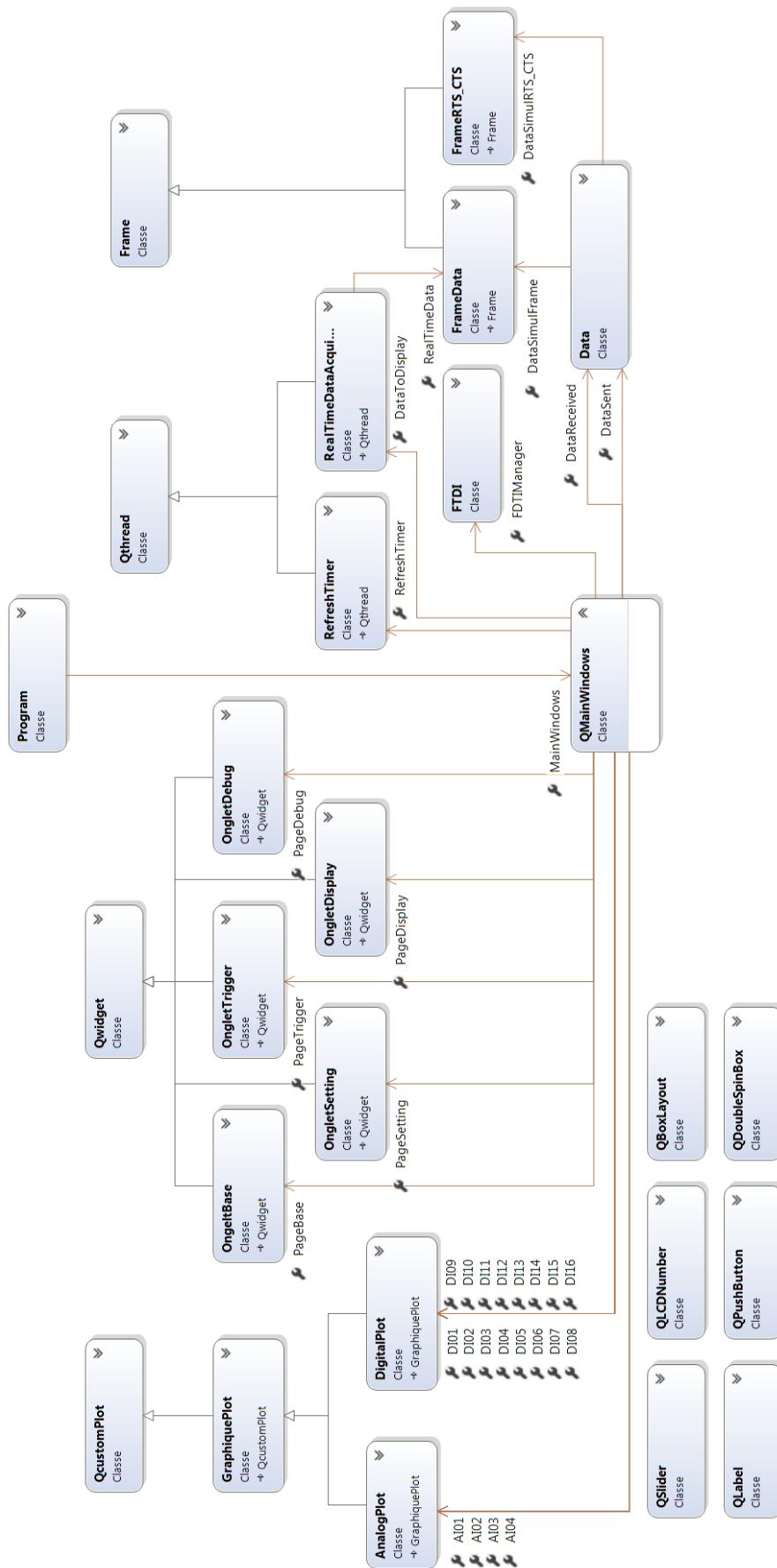
Ce menu sera réservé au contrôle des périphériques en cas de problème. Il évoluera en fonction du développement de mon application.

Dans un premier temps, je pense le laisser accessible à tout le monde, mais dans le futur, je vais sûrement le verrouiller et limiter l'accès avec un mot de passe.

## 12.0 Diagramme des classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

FIGURE 12.1 – Diagramme de Classes



## 13.0 La sauvegarde des données

### 13.1 L'emplacement

L'accès en lecture et écriture est beaucoup plus rapide sur la RAM que sur la Micro SD

Vitesse d'écriture sur la micro SD de classe 10 : 10 Mo/s

Vitesse d'écriture sur la RAM : 7200 Mo/s

### 13.2 Le volume

Il est défini de sauvegarder 65'536 trames de 64 bits, ce qui représente un volume de 4'194'240 bits soit un peu plus de 0.5 Mo. La place sur la RAM est suffisante pour le projet. Il serait aussi possible de sauver les données sur des mémoires externes attachées aux périphériques USB, de type clef ou disque dur externe. Mais l'accès est très lent et cela ne fait pas partie de ce projet.

### 13.3 Le type de fichier

Le fichier de sauvegarde devra pouvoir être lu avec d'autres outils.

#### 13.3.1 Type binaire

Ce type permet de sauvegarder des valeurs par octets, ce qui correspond exactement à un caractère. Il serait simple à implémenter, mais demande un lecteur spécial pour l'afficher.

C'est une solution très simple à coder et permet de sauvegarder les valeurs très rapidement.

#### 13.3.2 Type CSV

Ce type de fichier permet de sauvegarder en format de texte avec des caractères définis pour séparer les valeurs. Il peut être facilement lu par un éditeur standard de type Notepad++ ou importé dans le tableur de type Excel sous Windows.

La mise en forme demande un peu plus de travail en code afin qu'il soit compatible.

#### 13.3.3 Type text

Le type texte est très peu exploitable pour des analyses futures mais il est facile à implémenter

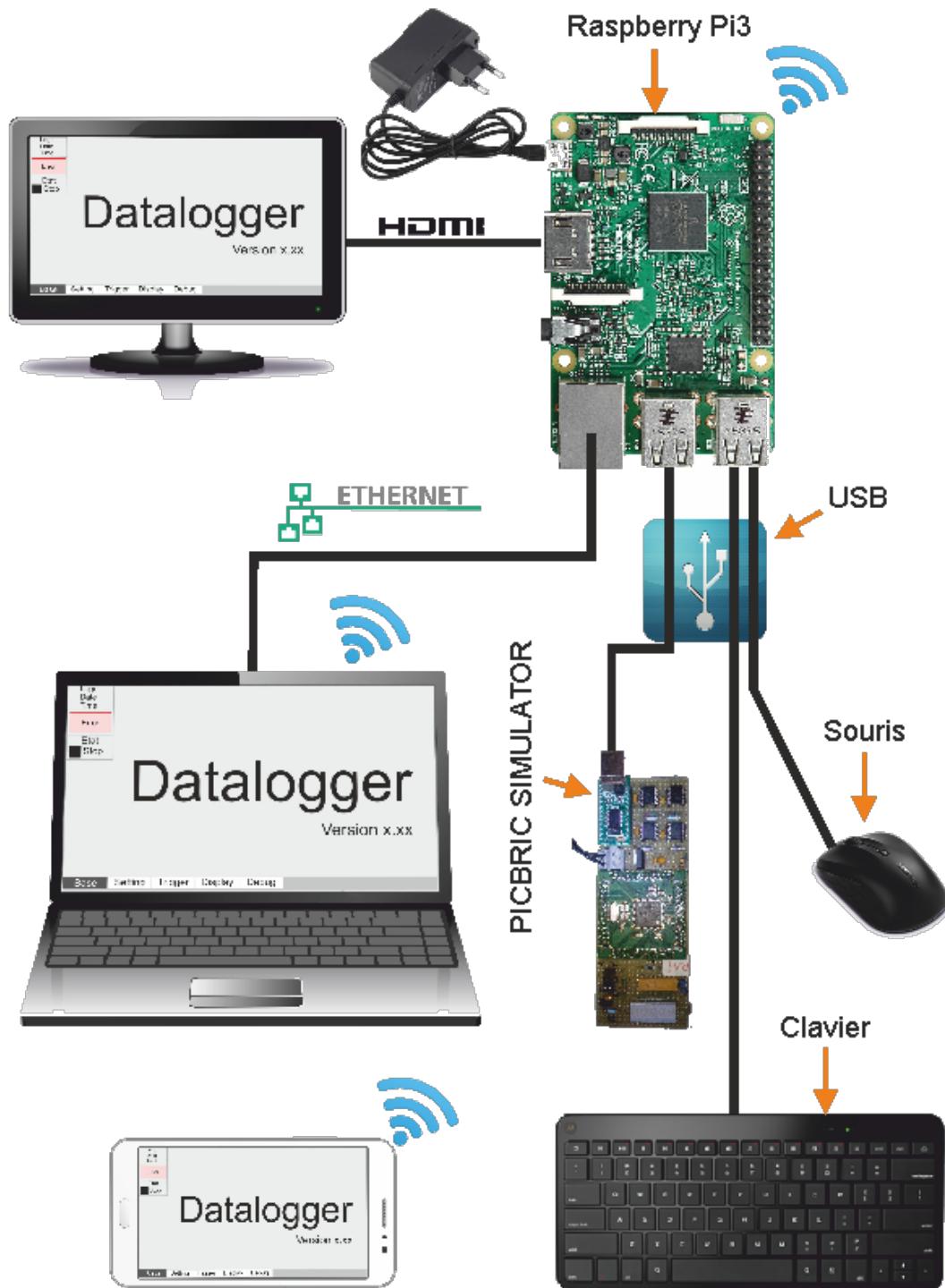
## 13.4 Conclusion

La préférence sera pour un fichier CSV. Afin de pouvoir l'analyser facilement.

L'emplacement sera en RAM, même avec l'application, il y a assez de place pour la sauvegarde.

## 14.0 Les périphériques d'accès et d'interfaces

FIGURE 14.1 – Les périphériques d'interfaces et d'accessibilités



## 15.0 Evaluation des risques

### 15.1 Tableau des risques

FIGURE 15.1 – Tableau des risques

Pondération des valeurs attribuées:	1 à 3 = faible	1-30 = faible
	4 à 7 = moyen	31-66 = moyen
	8 à 10 = important	67 - 100 = fort

n°	Risque	Probabilité	Incidence	Criticité	Implication / Plans d'actions / Commentaires
<b>Risque de l'objectif</b>					
1	Délais	1	10	10	
2	Coûts	1	1	1	
3	Qualité	3	8	24	
<b>Risque de maturité</b>					
4	Complexité trop élevée	6	8	48	Enlever des options non critiques
<b>Risque hiérarchiques</b>					
<b>Risque stratégiques</b>					
5	Changement d'objectifs	1	10	10	Ajuster le planifier et valider la faisabilité
<b>Risque d'outils</b>					
6	Méconnaissance LINUX	8	4	32	Formation possible sur le net
7	Méconnaissance C++	8	7	56	Formation possible sur le net
8	Environnement de développement	3	9	27	Passer sur Visual Studio
9	Pas adaptée	2	10	20	Changer la carte d'acquisition, d'autres sont possible
<b>Risque de pilotage</b>					
10	communication	3	7	21	Ne pas hésiter à convoquer le conseiller
<b>Risque Prévisionnels</b>					
11	Temps sous-estimé	9	9	81	Faire plus d'heures que prévu
<b>Risque Matériels</b>					
12	Carte de simulation défectueuse	4	10	40	En faire fabriquer une de réserve par les apprentis
13	Carte Raspberry défectueuse	2	10	20	Acheter une carte Raspberry Pi 3 en réserve
14	Carte SD défectueuse	1	10	10	Acheter une carte Raspberry Pi 3 en réserve
15	Ordinateur de développement défectueux	3	10	30	Trouver une autre machine
16	GPIO défectueux	1	5	5	Acheter une carte Raspberry Pi 3 en réserve

## 15.2 Graphique des risques

FIGURE 15.2 – Graphique des risques

	Insignifiant	Marginal	Critique	Catastrophique
Fréquent	Indépendant 14 5 13 9	Acceptable 15	Inacceptable	Inacceptable
Probable	Acceptable	Indésirable 8 12 3 10	Acceptable 4	Indésirable 11
Occasionnel	Acceptable	Indésirable	Indésirable	Indésirable 7
Rare	Négligeable 16	Acceptable	Indésirable	Indésirable 6
Improbable	Négligeable	Acceptable	Acceptable	Indésirable
Invraisemblable	Négligeable 2	Négligeable	Négligeable	Acceptable

## 15.3 Synthèse

Le point n°11 est le plus critique. Il est lié au temps planifié. En cas de gros problèmes, le nombre d'heures devra être augmenté ou il faudra diminuer la complexité. C'est cette partie qui je trouve la plus difficile à estimer.

Le point n°12 est géré, 3 cartes sont disponibles mais à configurer.

Pour les autres points dans le rouge, je vais commander le matériel en réserve. Le coût est d'environ 100.-, ce que je trouve très acceptable.

## 15.4 Conclusion

Je pense tout à fait raisonnable malgré le risque. Tous les problèmes pointés ont une solution.

Le plus difficile sera de résoudre les problèmes non planifiés qui vont survenir.

## 16.0 Planning

FIGURE 16.1 – planning page 1 sur 2

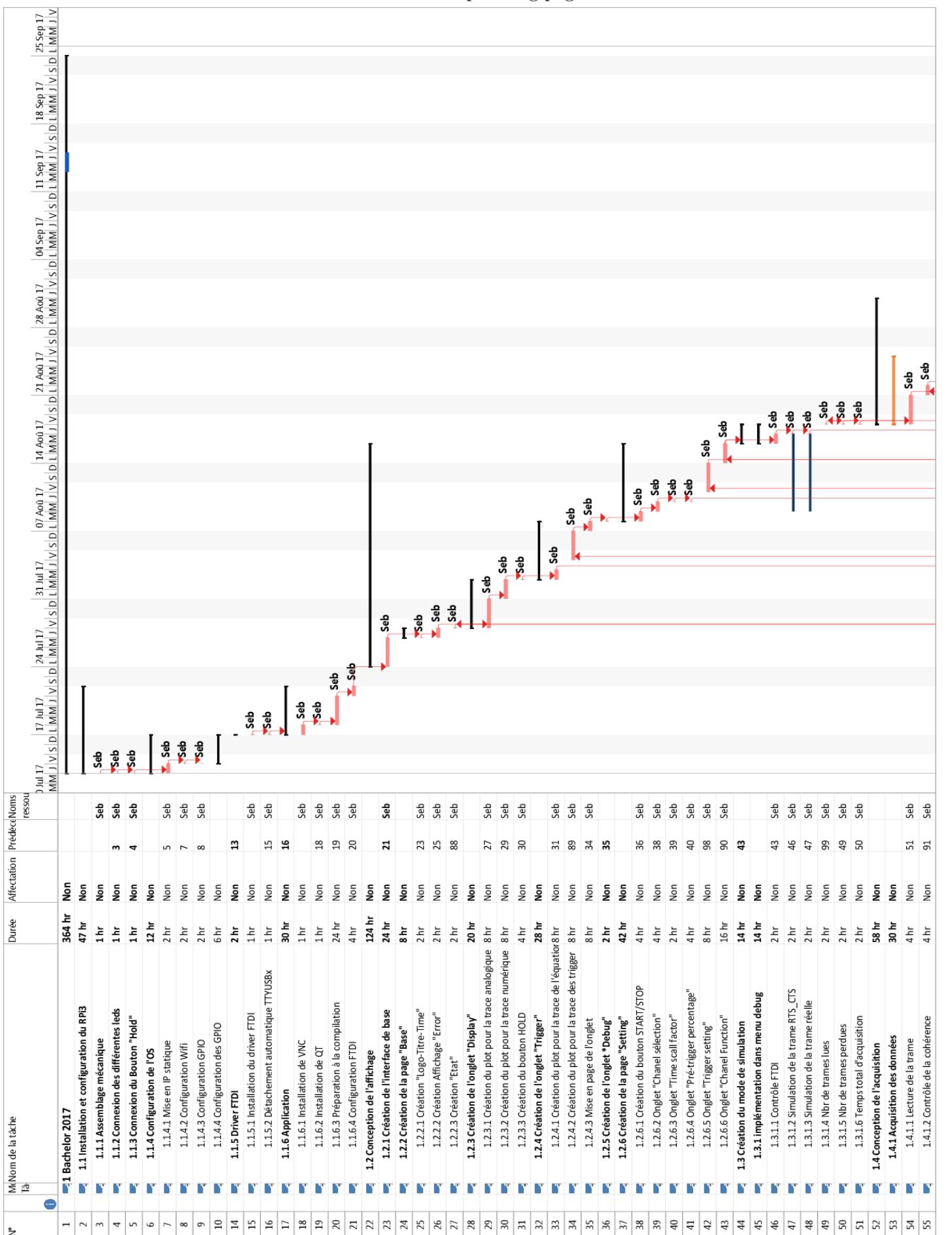
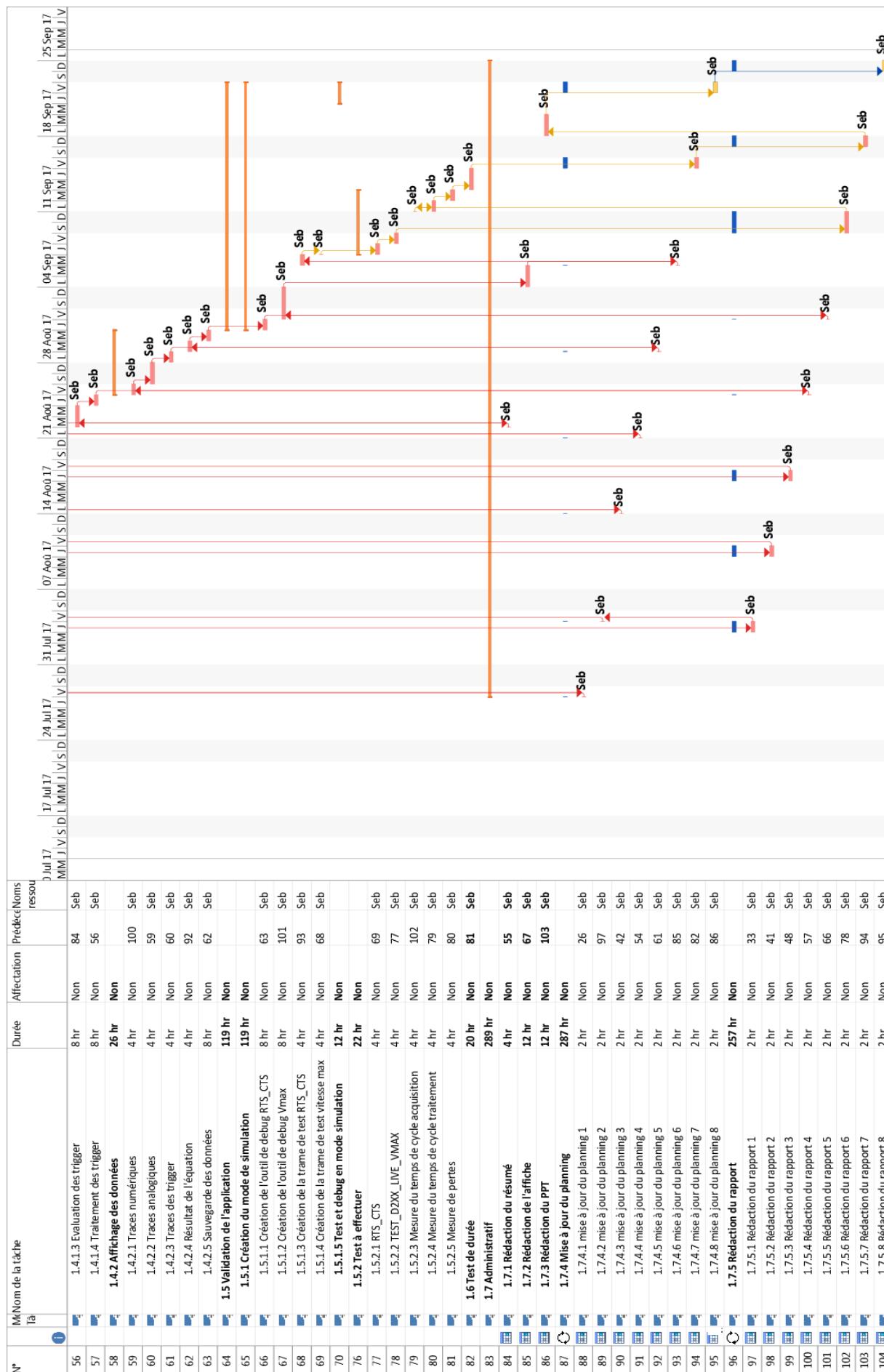


FIGURE 16.2 – planning page 2 sur 2





## 17.0 Conclusion

### 17.1 Bilan personnel

Je pense que mon niveau en orienté objet en C++ est trop faible pour pouvoir faire un code de qualité et optimal. Les cours que j'ai reçus n'ont pas été assez poussés pour ce projet. Je dois monter mon niveau avant juillet de cette année. Il y a beaucoup de cours en ligne possible. Cela va me demander un investissement personnel conséquent mais je pense cela réalisable et surtout très gratifiant lorsque le projet sera terminé.

Je suis aussi un peu faible en ce qui concerne l'environnement Linux, mais cela ne me fait pas trop peur, il y a une grosse communauté pour m'aider. Les connaissances acquises me semblent suffisantes mais je n'ai que peu de pratique à mon actif.

En ce qui concerne les mesures de validation et les tests, j'ai tout en main pour les réaliser. J'ai aussi l'ingénieur en électronique qui me soutient en cas question ou de problème avec la carte d'émulation.

Avec mon caractère pointilleux, il faut que je fasse attention à ne pas faire de la sur-qualité. Il faut que je m'en tienne au cahier des charges comme fil rouge.

### 17.2 Bilan du projet

Je pense le projet réalisable dans le temps imparti.

Le projet part avec le double de la vitesse d'acquisition prévue dans le cahier des charges et une ressource processeur restante de 60%.

Les solutions technologiques sont existantes. Le matériel est facilement remplaçable en cas de panne.

À la fin de ce projet, le Datalogger sera opérationnel pour de la capture de traces.

Ce qui va manquer principalement pour avoir un système industrialisable, c'est l'envoi de commande pour configurer la carte d'acquisition. Ceci sera fait lorsque la carte sera finie et fonctionnelle. Le délai est prévu pour la fin de cet été 2017.

Le but premier du mandant est d'en faire un démonstrateur, afin de pouvoir le proposer, démontrer son utilité et susciter l'intérêt qui créera un besoin. De là, l'industrialisation pourra vraiment commencer.

### 17.3 Message personnel

Je tiens déjà à remercier tous ceux qui me font confiance sur la réalisation de ce projet.

En particulier ma famille, qui a encore un été à souffrir de mon absence. Mon mandant, qui m'a proposé ce travail et mon conseiller qui a accepté ce défi.

Je trouve cela très gratifiant et je peux directement appliquer mes nouvelles connaissances acquises durant ces quatre années d'études.



## 17.3 Table des figures

1.1	Architecture pressentie . . . . .	11
2.1	Raspberry Pi 2 . . . . .	13
2.2	Raspberry Pi 3 . . . . .	13
2.3	Plan des GPIO du RPi3 . . . . .	15
2.4	Plan de cablage pour l'application . . . . .	16
3.1	chip FTDI . . . . .	17
3.2	Caractéristiques des interfaces série . . . . .	18
3.3	diagramme de flux des tests FTDI . . . . .	19
3.4	fonctionnement RTS CTS . . . . .	20
3.5	Carte de test TX_RX_CLOSE_LOOP . . . . .	21
3.6	Carte de test PICBRIC_SIMULATOR . . . . .	21
3.7	TEST_D2XX_LIVE, trame de test . . . . .	24
3.8	TEST_D2XX_LIVE, capture d'écran de résultat du test . . . . .	25
3.9	TEST_D2XX_LIVE, technique de mesure du test . . . . .	26
3.10	TEST_D2XX_LIVE, temps min et max d'une acquisition . . . . .	27
4.1	Trame de réception . . . . .	29
4.2	Trame de validation . . . . .	30
4.3	Traces de validation . . . . .	31
5.1	Principe d'acquisition des données . . . . .	33
5.2	Diagramme de flux du contrôle de cohérence . . . . .	34
9.1	Ancienne interface, la page principale . . . . .	45
9.2	Ancienne interface, configuration de la fonction de trigger . . . . .	46
9.3	Ancienne interface, configuration de l'équation logique . . . . .	47
9.4	Ancienne interface, l'ajustement du pré-trigger . . . . .	47
9.5	Ancienne interface, l'affichage des triggers et de l'équation logique . . . . .	48
9.6	Ancienne interface, l'affichage des traces . . . . .	49
9.7	Ancienne interface, le menu de debug . . . . .	49
10.1	Visibilité d'une interface homme machine . . . . .	52

11.1	Nouvelle interface, la page principale . . . . .	55
11.2	Nouvelle interface, la page principale, description . . . . .	55
11.3	Nouvelle interface, la page principale, les widgets . . . . .	56
11.4	Nouvelle interface, la page de configuration . . . . .	57
11.5	Nouvelle interface, la page de configuration , description du bouton start . . . . .	58
11.6	Nouvelle interface, la page de configuration , widgets pressentis du bouton start . . . . .	58
11.7	Nouvelle interface, la page de configuration , description de la moitié de gauche . . . . .	59
11.8	Nouvelle interface, la page de configuration , widgets presentis de la moitié de gauche . . . . .	59
11.9	Nouvelle interface, la page de configuration , description de la moitié de droite . . . . .	60
11.10	Nouvelle interface, la page de configuration , widgets de la moitié de droite . . . . .	60
11.11	Nouvelle interface, la page des triggers et de l'équation logique, l'aperçu . . . . .	61
11.12	Nouvelle interface, la page des triggers et de l'équation logique, la description . . . . .	61
11.13	Nouvelle interface, la page des triggers et de l'équation logique, les widgets . . . . .	62
11.14	Nouvelle interface, la page de visualistion . . . . .	63
11.15	Nouvelle interface, La page de debug . . . . .	64
12.1	Diagramme de Classes . . . . .	66
14.1	Les périphériques d'interfaces et d'accessibilités . . . . .	70
15.1	Tableau des risques . . . . .	71
15.2	Graphique des risques . . . . .	72
16.1	planning page 1 aur 2 . . . . .	74
16.2	planning page 2 sur 2 . . . . .	75
B.1	Tableau des tâches page 1 aur 2 . . . . .	84
B.2	Tableau des tâches page 2 sur 2 . . . . .	85
C.1	Calcul des masques binaires page 1 sur 3 . . . . .	88
C.2	Calcul des masques binaires page 2 sur 3 . . . . .	89
C.3	Calcul des masques binaires page 3 sur 3 . . . . .	90

## A.0 P.V. séance

### A.1 Historique des entretiens

Date	Emplacement	Durée
9 mai 2017	Cafétéria St-Roch, 1401 Yverdon-les-Bains.	45 minutes

## A.2 9 mai 2017

**Participants :** M. Bressy Pierrs, M. Blessemaille Sébastien

**Durée :** 45 minutes.

**Localisation :** Cafétéria St-Roch, 1401 Yverdon-les-Bains.

**Sujet :** Diverses questions relatives à la rédaction de la pré-étude.

**Points abordés :**

- Titre des chapitres en majuscules ou minuscule : selon mes envies.
- Livraison du document : relié.
- Principe d'acquisition : deux buffers dans une pipe OK.
- Diagramme de classes : à ajouter dans le rapport.
- Étape de la conception : décrire les étapes clés.
- Philosophie de la pré-étude : document assez précis qui devrait permettre à une autre personne de faire le projet
- Détail de la table des matières : selon mon envie.
- Symbolique des puces : selon mon envie.
- User story pour le fonctionnement : pas besoin d'être aussi précis.
- Annexes : de préférence un lien où télécharger les documents.
- Sujet / mandat à mettre dans la pré-étude : non
- Cahier des charges à mettre dans la pré-étude : selon mon envie.
- Mettre les compétences personnelles à acquérir : Oui.
- Planning des RDV : Mettre dans un tableau en annexe.
- Tests effectués : Diagramme de flux et pseudo-code ok, pas de code.

## B.0 Tableau des tâches

## Annexe B. Tableau des tâches

FIGURE B.1 – Tableau des tâches page 1 sur 2

N°	Mod. Tâche	Nom de la tâche	Durée	Affectation	Prédécesseur	Noms ressource	Début	Fin
1		Bachelor 2017						
2		Installation et configuration du RPi3	364 hr	Non			Jeu 13.07.17	Lun 25.09.17
3		Assemblage mécanique	47 hr	Non			Jeu 13.07.17	Ven 21.07.17
4		Connexion des différentes leds	1 hr	Non	3	Seb	Jeu 13.07.17	Jeu 13.07.17
5		Connexion du Bouton "Hold"	1 hr	Non	4	Seb	Jeu 13.07.17	Jeu 13.07.17
6		Configuration de l'OS	12 hr	Non			Jeu 13.07.17	Lun 17.07.17
7		Mise en IP statique	2 hr	Non	5	Seb	Jeu 13.07.17	Jeu 13.07.17
8		Configuration Wifi	2 hr	Non	7	Seb	Ven 14.07.17	Ven 14.07.17
9		Configuration GPIO	2 hr	Non	8	Seb	Ven 14.07.17	Ven 14.07.17
10		Configuration des GPIO	6 hr	Non			Ven 14.07.17	Lun 17.07.17
14		Driver FTDI	2 hr	Non	13		Lun 17.07.17	Lun 17.07.17
15		Installation du driver FTDI	1 hr	Non		Seb	Lun 17.07.17	Lun 17.07.17
16		Détachement automatique TTYUSBx	1 hr	Non	15	Seb	Lun 17.07.17	Lun 17.07.17
17		Application	30 hr	Non	16		Lun 17.07.17	Ven 21.07.17
18		Installation de VNC	1 hr	Non		Seb	Lun 17.07.17	Lun 17.07.17
19		Installation de QT	1 hr	Non	18	Seb	Lun 17.07.17	Lun 17.07.17
20		Préparation à la compilation	24 hr	Non	19	Seb	Mar 18.07.17	Ven 21.07.17
21		Configuration FTDI	4 hr	Non	20	Seb	Ven 21.07.17	Ven 21.07.17
22		Conception de l'affichage	124 hr	Non			Lun 24.07.17	Mer 16.08.17
23		Création de l'interface de base	24 hr	Non	21	Seb	Lun 24.07.17	Jeu 27.07.17
24		Création de la page "Base"	8 hr	Non			Jeu 27.07.17	Ven 28.07.17
25		Création "Logo-Titre-Time"	2 hr	Non	23	Seb	Jeu 27.07.17	Jeu 27.07.17
26		Création Affichage "Error"	2 hr	Non	25	Seb	Jeu 27.07.17	Jeu 27.07.17
27		Création "Etat"	2 hr	Non	88	Seb	Ven 28.07.17	Ven 28.07.17
28		Création de l'onglet "Display"	20 hr	Non			Ven 28.07.17	Mer 02.08.17
29		Création du plot pour la trace analogique	8 hr	Non	27	Seb	Ven 28.07.17	Lun 31.07.17
30		Création du plot pour la trace numérique	8 hr	Non	29	Seb	Lun 31.07.17	Mar 01.08.17
31		Création du bouton HOLD	4 hr	Non	30	Seb	Mar 01.08.17	Mer 02.08.17
32		Création de l'onglet "Trigger"	28 hr	Non			Mer 02.08.17	Mar 08.08.17
33		Création du plot pour la trace de l'équation	8 hr	Non	31	Seb	Mer 02.08.17	Jeu 03.08.17
34		Création du plot pour la trace des trigger	8 hr	Non	89	Seb	Ven 04.08.17	Lun 07.08.17
35		Mise en page de l'onglet	8 hr	Non	34	Seb	Lun 07.08.17	Mar 08.08.17
36		Création de l'onglet "Debug"	2 hr	Non	35		Mar 08.08.17	Mar 08.08.17
37		Création de la page "Setting"	42 hr	Non			Mar 08.08.17	Mer 16.08.17
38		Création du bouton START/STOP	4 hr	Non	36	Seb	Mar 08.08.17	Mer 09.08.17
39		Onglet "Chanel sélection"	4 hr	Non	38	Seb	Mer 09.08.17	Mer 09.08.17
40		Onglet "Time scall factor"	2 hr	Non	39	Seb	Mer 09.08.17	Jeu 10.08.17
41		Onglet "Pré-trigger percentage"	4 hr	Non	40	Seb	Jeu 10.08.17	Jeu 10.08.17
42		Onglet "Trigger setting"	8 hr	Non	98	Seb	Ven 11.08.17	Lun 14.08.17
43		Onglet "Chanel Function"	16 hr	Non	90	Seb	Lun 14.08.17	Mer 16.08.17
44		Création du mode de simulation	14 hr	Non	43		Mer 16.08.17	Ven 18.08.17
45		implémentation dans menu debug	14 hr	Non			Mer 16.08.17	Ven 18.08.17
46		Contrôle FTDI	2 hr	Non	43	Seb	Mer 16.08.17	Mer 16.08.17
47		Simulation de la trame RTS_CTS	2 hr	Non	46	Seb	Jeu 17.08.17	Jeu 17.08.17
48		Simulation de la trame réelle	2 hr	Non	47	Seb	Jeu 17.08.17	Jeu 17.08.17
49		Nbr de trames lues	2 hr	Non	99	Seb	Jeu 17.08.17	Ven 18.08.17
50		Nbr de trames perdues	2 hr	Non	49	Seb	Ven 18.08.17	Ven 18.08.17
51		Temps total d'acquisition	2 hr	Non	50	Seb	Ven 18.08.17	Ven 18.08.17
52		Conception de l'acquisition	58 hr	Non			Ven 18.08.17	Mer 30.08.17
53		Acquisition des données	30 hr	Non			Ven 18.08.17	Jeu 24.08.17
54		Lecture de la trame	4 hr	Non	51	Seb	Ven 18.08.17	Lun 21.08.17
55		Contrôle de la cohérence	4 hr	Non	91	Seb	Lun 21.08.17	Mar 22.08.17
56		Evaluation des trigger	8 hr	Non	84	Seb	Mar 22.08.17	Mer 23.08.17
57		Traitements des trigger	8 hr	Non	56	Seb	Mer 23.08.17	Jeu 24.08.17
58		Affichage des données	26 hr	Non			Ven 25.08.17	Mer 30.08.17
59		Traces numériques	4 hr	Non	100	Seb	Ven 25.08.17	Ven 25.08.17
60		Traces analogiques	4 hr	Non	59	Seb	Ven 25.08.17	Lun 28.08.17
61		Traces des trigger	4 hr	Non	60	Seb	Lun 28.08.17	Lun 28.08.17
62		Résultat de l'équation	4 hr	Non	92	Seb	Mar 29.08.17	Mar 29.08.17
63		Sauvegarde des données	8 hr	Non	62	Seb	Mar 29.08.17	Mer 30.08.17
64		Validation de l'application	119 hr	Non			Jeu 31.08.17	Ven 22.09.17
65		Création du mode de simulation	119 hr	Non			Jeu 31.08.17	Ven 22.09.17
66		Création de l'outil de debug RTS_CTS	8 hr	Non	63	Seb	Jeu 31.08.17	Ven 01.09.17
67		Création de l'outil de debug Vmax	8 hr	Non	101	Seb	Ven 01.09.17	Lun 04.09.17
68		Création de la trame de test RTS_CTS	4 hr	Non	93	Seb	Mer 06.09.17	Jeu 07.09.17
69		Création de la trame de test vitesse max	4 hr	Non	68	Seb	Jeu 07.09.17	Jeu 07.09.17
70		Test et debug en mode simulation	12 hr	Non			Jeu 21.09.17	Ven 22.09.17
76		Test à effectuer	22 hr	Non			Jeu 07.09.17	Mar 12.09.17
77		RTS_CTS	4 hr	Non	69	Seb	Jeu 07.09.17	Ven 08.09.17
78		TEST_D2XX_LIVE_VMAX	4 hr	Non	77	Seb	Ven 08.09.17	Ven 08.09.17
79		Mesure du temps de cycle acquisition	4 hr	Non	102	Seb	Lun 11.09.17	Lun 11.09.17
80		Mesure du temps de cycle traitement	4 hr	Non	79	Seb	Lun 11.09.17	Mar 12.09.17
81		Mesure de pertes	4 hr	Non	80	Seb	Mar 12.09.17	Mar 12.09.17
82		Test de durée	20 hr	Non	81	Seb	Mar 12.09.17	Ven 15.09.17
83		Administratif	289 hr	Non			Ven 28.07.17	Lun 25.09.17
84		Rédaction du résumé	4 hr	Non	55	Seb	Mar 22.08.17	Mar 22.08.17
85		Rédaction de l'affiche	12 hr	Non	67	Seb	Lun 04.09.17	Mer 06.09.17
86		Rédaction du PPT	12 hr	Non	103	Seb	Lun 18.09.17	Mar 19.09.17
87	↻	Mise à jour du planning	287 hr	Non			Ven 28.07.17	Ven 22.09.17
88	🕒	Mise à jour du planning 1	2 hr	Non	26	Seb	Ven 28.07.17	Ven 28.07.17

N°	Mod. Tâche	Nom de la tâche	Durée	Affectation	Prédecesseur	Noms ressources	Début	Fin
89		Mise à jour du planning 2	2 hr	Non	97	Seb	Jeu 03.08.17	Ven 04.08.17
90		Mise à jour du planning 3	2 hr	Non	42	Seb	Lun 14.08.17	Lun 14.08.17
91		Mise à jour du planning 4	2 hr	Non	54	Seb	Lun 21.08.17	Lun 21.08.17
92		Mise à jour du planning 5	2 hr	Non	61	Seb	Mar 29.08.17	Mar 29.08.17
93		Mise à jour du planning 6	2 hr	Non	85	Seb	Mer 06.09.17	Mer 06.09.17
94		Mise à jour du planning 7	2 hr	Non	82	Seb	Ven 15.09.17	Ven 15.09.17
95		Mise à jour du planning 8	2 hr	Non	86	Seb	Ven 22.09.17	Ven 22.09.17
96	↻	<b>Rédaction du rapport</b>	<b>257 hr</b>	<b>Non</b>			<b>Jeu 03.08.17</b>	<b>Lun 25.09.17</b>
97		Rédaction du rapport 1	2 hr	Non	33	Seb	Jeu 03.08.17	Jeu 03.08.17
98		Rédaction du rapport 2	2 hr	Non	41	Seb	Jeu 10.08.17	Jeu 10.08.17
99		Rédaction du rapport 3	2 hr	Non	48	Seb	Jeu 17.08.17	Jeu 17.08.17
100		Rédaction du rapport 4	2 hr	Non	57	Seb	Ven 25.08.17	Ven 25.08.17
101		Rédaction du rapport 5	2 hr	Non	66	Seb	Ven 01.09.17	Ven 01.09.17
102		Rédaction du rapport 6	2 hr	Non	78	Seb	Ven 08.09.17	Lun 11.09.17
103		Rédaction du rapport 7	2 hr	Non	94	Seb	Dim 17.09.17	Lun 18.09.17
104		Rédaction du rapport 8	2 hr	Non	95	Seb	Dim 24.09.17	Lun 25.09.17

FIGURE B.2 – Tableau des tâches page 2 sur 2



## C.0 Tableau de calculs des masques binaires

FIGURE C.1 – Calcul des masques binaires page 1 sur 3

FIGURE C.2 – Calcul des masques binaires page 2 sur 3

		(A ET B)		(A OU B)		(A ET / B)		(A OU / B)		(A ET / B)		(A OU / B)		(A ET / B)	
D	C	B	A	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)	(C ET D)
0	0	0	0	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX
1	0	0	1	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX
2	0	1	0	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX	FAUX
3	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
4	0	1	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
5	0	1	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
6	0	1	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
7	0	1	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
8	1	0	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
9	1	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
10	1	0	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
11	1	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
12	1	1	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
13	1	1	0	FAUX	VRAI	VRAI	VRAI								
14	1	1	1	FAUX	VRAI	VRAI	Faux								
15	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
= 15				> 12				= 13				= 3			
VAL EQ				VRAI				= 7				= 7			
VAL EQ				VAL EQ				35941				35953			
		(A ET B)		(A OU B)		(A ET / B)		(A OU / B)		(A ET B)		(A OU / B)		(A ET / B)	
D	C	B	A	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)
0	0	0	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
1	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
2	0	1	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
3	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
4	0	1	0	FAUX	VRAI	VRAI	VRAI								
5	0	1	1	FAUX	VRAI	VRAI	VRAI								
6	0	1	1	FAUX	VRAI	VRAI	VRAI								
7	0	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
8	1	0	0	FAUX	VRAI	VRAI	VRAI								
9	1	0	1	FAUX	VRAI	VRAI	VRAI								
10	1	0	0	FAUX	VRAI	VRAI	VRAI								
11	1	0	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
12	1	1	0	FAUX	VRAI	VRAI	VRAI								
13	1	1	1	FAUX	VRAI	VRAI	VRAI								
14	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
15	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
= 7				> 4 ET < 8				= 5				> 3 ET < 6			
VAL EQ				VRAI				= 11				= 15			
VAL EQ				37987				37993				38001			
		(A ET B)		(A OU B)		(A ET / B)		(A OU / B)		(A ET / B)		(A OU / B)		(A ET / B)	
D	C	B	A	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)
0	0	0	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
1	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
2	0	1	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
3	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
4	0	1	0	FAUX	VRAI	VRAI	VRAI								
5	0	1	1	FAUX	VRAI	VRAI	VRAI								
6	0	1	1	FAUX	VRAI	VRAI	VRAI								
7	0	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
8	1	0	0	FAUX	VRAI	VRAI	VRAI								
9	1	0	1	FAUX	VRAI	VRAI	VRAI								
10	1	0	0	FAUX	VRAI	VRAI	VRAI								
11	1	0	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
12	1	1	0	FAUX	VRAI	VRAI	VRAI								
13	1	1	1	FAUX	VRAI	VRAI	VRAI								
14	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
15	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
= 7				> 4 ET < 8				= 5				> 3 ET < 6			
VAL EQ				VRAI				= 11				= 13			
VAL EQ				37987				37993				38001			
		(A ET B)		(A OU B)		(A ET / B)		(A OU / B)		(A ET / B)		(A OU / B)		(A ET / B)	
D	C	B	A	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)	(C OUD)
0	0	0	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	VRAI	VRAI
1	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
2	0	1	0	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
3	0	0	1	FAUX	FAUX	FAUX	FAUX	VRAI	FAUX	FAUX	FAUX	VRAI	FAUX	Faux	VRAI
4	0	1	1	FAUX	VRAI	VRAI	VRAI								
5	0	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
6	0	1	1	FAUX	VRAI	VRAI	VRAI								
7	1	0	0	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
8	1	0	1	FAUX	VRAI	VRAI	VRAI								
9	1	0	0	FAUX	VRAI	VRAI	VRAI								
10	1	0	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
11	1	0	1	FAUX	VRAI	VRAI	VRAI								
12	1	1	0	FAUX	VRAI	VRAI	VRAI								
13	1	1	1	FAUX	VRAI	VRAI	VRAI								
14	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
15	1	1	1	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
= 7				> 8 ET < 12											

FIGURE C.3 – Calcul des masques binaires page 3 sur 3

## D.0 Temps de réalisation de la pré-étude

Description	Nombre d'heures
Validation de la carte de simulation	20 h
Rédaction d'une partie du document sous Word	30 h
Prise en main de Latex	10 h
Passage du document de Word à Latex	8 h
Rédaction du document sous Latex	16 h
Mise en page et finitions	8 h
<b>Total</b>	<b>92 h</b>