

# Filière Systèmes industriels

Orientation Infotronics

# Diplôme 2014

*Romain Franzen*


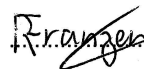
*Monitoring environnemental*

-  *Professeur*  
François Corthay
-  *Expert*  
Eric Travaglini
-  *Date de la remise du rapport*  
11.07.2014

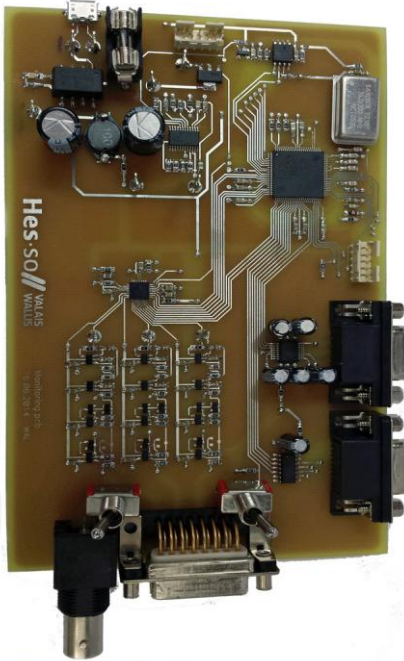
SI	TV
X	X

<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / Studienjahr 2013/14	No TD / Nr. DA it/2014/63
Mandant / Auftraggeber <input type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input checked="" type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>	Etudiant / Student <b>Romain Franzen</b>  Professeur / Dozent <b>François Corthay</b>	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja <sup>1</sup> <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) <b>Eric Travaglini</b> CREALP   Rue de l'Industrie 45   1950 Sion	

Titre / Titel  <b>Monitoring environnemental</b>
Description / Beschreibung  Le Centre de Recherche sur l'Environnement ALPin (CREALP) est intéressé à faire l'acquisition de données sur le terrain et à transmettre les informations correspondantes à distance. La fréquence d'échantillonnage est de l'ordre de grandeur de 10 kHz. La transmission se fait par modem GPRS. Afin de limiter la quantité de données à transmettre, un travail de mise en forme est à effectuer : redressement du signal, filtrage, calcul d'enveloppe, seuillage et finalement comptage d'événements.  Le but du travail de diplôme est de réaliser un circuit imprimé servant à l'acquisition du signal d'un géophone, au compactage de l'information et à l'émission des données via un modem GSM. Le circuit doit être suffisamment adaptable pour permettre l'évolution de l'algorithme de traitement du signal.  Ce développement pourra servir par la suite pour d'autres capteurs comme des hydrophones ou des pluviomètres acoustiques.  Objectifs / Ziele – Réalisation d'une carte FPGA à basse consommation – Développement d'un algorithme de traitement de l'information – Réalisation d'un démonstrateur.

Signature ou visa / Unterschrift oder Visum  Responsable de l'orientation Leiter der Vertiefungsrichtung:   <sup>1</sup> Etudiant / Student : 	Délais / Termine  Attribution du thème / Ausgabe des Auftrags: 12.05.2014  Remise du rapport / Abgabe des Schlussberichts: 11.07.2014, 12:00  Expositions / Ausstellungen der Diplomarbeiten: 27 – 29.08.2014  Défense orale / Mündliche Verfechtung: Semaine   Woche 36
---	--

<sup>1</sup> Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.  
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.



## Monitoring Environnemental

Diplômant/e Romain Franzen

### Objectif du projet

L'organisation CREALP s'intéresse aux dangers naturels, comme les laves torrentielles ou le charriage. Un prototype qui traite les mesures prises par un géophone va être créé. Il doit être autonome et doit fonctionner pendant plusieurs jours.

### Méthodes | Expériences | Résultats

Nous avons créé une carte électronique Low Power, permettant l'acquisition de données émises par un géophone qui peut mesurer dans trois directions possibles. Pour y arriver, nous nous sommes basés sur des fichiers fournis par le CREALP. A partir de ces documents, des courbes ont été dessinées et analysées. Des algorithmes ont ensuite été testés sur ces courbes. Ces algorithmes ont pour but de réduire la quantité de données et ils seront implémentés dans l'unité centrale de traitement.

Quant à la carte, elle est divisée en plusieurs parties : une première, dont le rôle est l'acquisition et le traitement des données, une seconde possédant un convertisseur analogique-digital qui convertit les signaux du géophone en nombres exploitables, d'une mémoire locale et enfin, de port permettant l'envoi de données à un serveur via un module de télécommunication.

L'unité centrale, le filtre analogique, les algorithmes ainsi que la communication série et I2C fonctionnent, mais le convertisseur AD ne convertit pas encore les tensions.

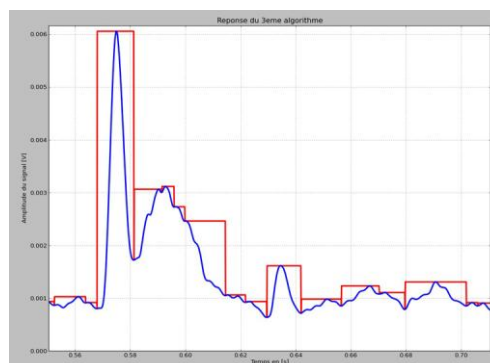
Travail de diplôme  
| édition 2014 |

Filière  
*Systèmes industriels*

Domaine d'application  
*Infotronic*

Professeur responsable  
*François Corthay*  
*Francois.corthay@hevs.ch*

Partenaire  
*Centre de recherche sur  
l'environnement alpin  
Rue de l'industrie 45  
CH-1951 Sion*


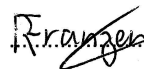


La courbe du dessous représente le signal original. La courbe rectangulaire quant à elle, symbolise la réponse à l'algorithme des valeurs maximums.

SI	TV
X	X

<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / Studienjahr 2013/14	No TD / Nr. DA it/2014/63
Mandant / Auftraggeber <input type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input checked="" type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>	Etudiant / Student <b>Romain Franzen</b>  Professeur / Dozent <b>François Corthay</b>	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja <sup>1</sup> <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) <b>Eric Travaglini</b> CREALP   Rue de l'Industrie 45   1950 Sion	

Titre / Titel  <b>Monitoring environnemental</b>
Description / Beschreibung  Le Centre de Recherche sur l'Environnement ALPin (CREALP) est intéressé à faire l'acquisition de données sur le terrain et à transmettre les informations correspondantes à distance. La fréquence d'échantillonnage est de l'ordre de grandeur de 10 kHz. La transmission se fait par modem GPRS. Afin de limiter la quantité de données à transmettre, un travail de mise en forme est à effectuer : redressement du signal, filtrage, calcul d'enveloppe, seuillage et finalement comptage d'événements.  Le but du travail de diplôme est de réaliser un circuit imprimé servant à l'acquisition du signal d'un géophone, au compactage de l'information et à l'émission des données via un modem GSM. Le circuit doit être suffisamment adaptable pour permettre l'évolution de l'algorithme de traitement du signal.  Ce développement pourra servir par la suite pour d'autres capteurs comme des hydrophones ou des pluviomètres acoustiques.  Objectifs / Ziele – Réalisation d'une carte FPGA à basse consommation – Développement d'un algorithme de traitement de l'information – Réalisation d'un démonstrateur.

Signature ou visa / Unterschrift oder Visum  Responsable de l'orientation Leiter der Vertiefungsrichtung:   <sup>1</sup> Etudiant / Student : 	Délais / Termine  Attribution du thème / Ausgabe des Auftrags: 12.05.2014  Remise du rapport / Abgabe des Schlussberichts: 11.07.2014, 12:00  Expositions / Ausstellungen der Diplomarbeiten: 27 – 29.08.2014  Défense orale / Mündliche Verfechtung: Semaine   Woche 36
---	--

<sup>1</sup> Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.  
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.

# Table des matières

<b>1</b>	<b>Lexique</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
2.1	Contexte . . . . .	5
2.2	Cahier des charges et objectifs . . . . .	6
<b>3</b>	<b>Matériel et Méthode</b>	<b>7</b>
3.1	Matériels et logiciels utilisés . . . . .	7
3.2	Les géophones . . . . .	7
3.2.1	Fonctionnement d'un géophone . . . . .	7
3.2.2	Géophone du CREALP . . . . .	8
3.2.3	Tests réalisés sur le géophone . . . . .	8
3.2.4	Traitement du signal . . . . .	8
3.3	Liste du Matériel . . . . .	10
3.3.1	Parties principales . . . . .	10
3.3.2	Alimentation . . . . .	11
3.3.3	Batterie . . . . .	11
3.3.4	Régulateur ou Convertisseur . . . . .	11
3.3.5	Connecteur d'alimentation 5V . . . . .	11
3.3.6	Convertisseur DC-DC 5V-3.3V . . . . .	11
3.3.7	Régulateur 3.3V - 1.5V . . . . .	12
3.3.8	Acquisition des données . . . . .	12
3.3.9	Convertisseur AD . . . . .	12
3.3.10	Diode Zener . . . . .	12
3.3.11	Traitement des données . . . . .	12
3.3.12	FPGA . . . . .	13
3.3.13	Connecteur JTAG . . . . .	14
3.3.14	Manuel Reset . . . . .	14
3.3.15	Oscillateur . . . . .	14
3.3.16	Mémoire . . . . .	14
3.3.17	Transmission . . . . .	15
3.3.18	Transmetteur et récepteur RS-232 . . . . .	15
3.3.19	Transmetteur et récepteur RS-422 . . . . .	15
3.3.20	Connexion SUB-D 15 . . . . .	15
3.3.21	Transistor . . . . .	15
3.4	Méthode de calcul / Algorithme . . . . .	16
<b>4</b>	<b>Développement</b>	<b>17</b>
4.1	Puissance de calcul . . . . .	17
4.2	Matériel et schéma électrique . . . . .	17
4.2.1	Alimentation . . . . .	18
4.2.2	Convertisseur DC-DC . . . . .	19
4.2.3	Alimentation FPGA Régulateur 3.3V vers 1.5V : TPS70402 . . . . .	19
4.2.4	FPGA . . . . .	20
4.2.5	Reset MAX811 . . . . .	22
4.2.6	EEPROM . . . . .	23
4.2.7	Convertisseur AD : . . . . .	24
4.2.8	Diode Zener . . . . .	25

4.2.9	TTL et CMOS . . . . .	26
4.2.10	Transmetteur et récepteur RS-232 ICL3221E . . . . .	27
4.2.11	Driver RS-422 . . . . .	28
4.2.12	Connexion SUB-D 15 . . . . .	29
4.2.13	Mosfet . . . . .	29
4.2.14	Switchs . . . . .	30
4.2.15	Résistance de 0 Ohm . . . . .	32
4.2.16	Divers petits matériels . . . . .	32
4.2.17	Résumé du matériel à acheter . . . . .	32
4.2.18	Carte électronique . . . . .	33
4.3	Filtre digital-analogique . . . . .	33
4.4	Premier traitement du signal . . . . .	33
4.5	Diviseur de tension . . . . .	34
4.6	Redresseur . . . . .	35
4.6.1	Redresseur à partir d'additionneurs et d'inverseurs . . . . .	35
4.6.2	Redresseur à partir d'amplificateurs opérationnels et de diodes . . . . .	36
4.6.3	Choix du redresseur . . . . .	38
4.7	Filtre passe-bas analogique . . . . .	38
4.7.1	Filtre passe-bas passif . . . . .	39
4.7.2	Filtre passe-bas actif . . . . .	39
4.7.3	Choix du filtre passe-bas. . . . .	42
4.8	Configuration pour la lecture et l'écriture avec l'EEPROM . . . . .	42
4.8.1	Lecture . . . . .	42
4.8.2	Ecriture . . . . .	43
4.9	Algorithmes . . . . .	43
4.9.1	Algorithme : Variation d'amplitude . . . . .	44
4.9.2	Schéma de l'algorithme . . . . .	45
4.9.3	Tests effectués avec l'algorithme . . . . .	48
4.9.4	Algorithme : Moyenne des maximums . . . . .	48
4.9.5	Schéma du second algorithme . . . . .	49
4.9.6	Tests effectués avec l'algorithme . . . . .	49
4.9.7	Algorithme : Valeurs maximums . . . . .	50
4.9.8	Schéma du troisième algorithme . . . . .	50
4.9.9	Tests effectués avec l'algorithme . . . . .	51
4.9.10	Comparaison des trois algorithmes . . . . .	52
4.9.11	Algorithme : Détection de seuils . . . . .	55
4.10	Implémentation VHDL . . . . .	56
4.10.1	Les étapes . . . . .	56
4.10.2	Synchrone ou asynchrone . . . . .	57
4.10.3	Horloge . . . . .	57
4.10.4	Configuration pour les tests . . . . .	58
4.10.5	Protocole I2C . . . . .	58
4.10.6	Protocole I2S . . . . .	59
4.10.7	RS-232 . . . . .	61
4.10.8	Convertisseur AD . . . . .	62
4.10.9	Algorithmes : . . . . .	64
4.10.10	Mémoire : . . . . .	65
4.11	Taux d'occupation dans la FPGA . . . . .	66
4.12	Codage des données . . . . .	67

4.13	Tests effectués sur la carte . . . . .	67
4.13.1	Test du signal réel . . . . .	68
4.14	Calcul de la consommation de courant totale . . . . .	70
<b>5</b>	<b>Discussion</b>	<b>72</b>
5.1	Partie Matériel . . . . .	72
5.2	Partie Algorithmique . . . . .	72
5.3	Partie logicielle . . . . .	73
5.4	Améliorations futures . . . . .	74
<b>6</b>	<b>Conclusion</b>	<b>76</b>
<b>7</b>	<b>Date et signature</b>	<b>76</b>
<b>8</b>	<b>Référence</b>	<b>77</b>
8.1	Datasheets . . . . .	79
	<b>Appendices</b>	<b>81</b>

## Table des figures

1	Géophone DT20DX [2] . . . . .	5
2	Schéma des parties du projet. . . . .	6
3	Géophone avec la masse magnétique . . . . .	7
4	Tension minimale de sortie du géophone Dynamic Technologies DT20DX 4.5Hz . . . . .	9
5	Schéma électrique des principaux composants principaux. . . . .	10
6	Schéma avec la diode Zener pour la limitation de la tension . . . . .	12
7	Graphique de consommation d'une FPGA Igloo par rapport à deux autres FPGA. . . . .	13
8	Graphique de consommation d'une FPGA Igloo par rapport à deux autres FPGA en fonction de la fréquence. . . . .	14
9	Schéma du circuit. . . . .	18
10	Schéma du bloc alimentation . . . . .	18
11	FPGA Igloo Nano de MICROSEMI. . . . .	21
12	Simulation du circuit contenant la diode Zener. . . . .	26
13	CMOS et TTL . . . . .	27
14	Schéma du switch qui permet de sélectionner le géophone désiré. . . . .	31
15	Schéma du switch qui permet de sélectionner la sortie du géophone qui va être redirigée. . . . .	31
16	Position des switches pour déterminer les fonctions que l'on désire. . . . .	31
17	Carte électronique du projet. . . . .	33
18	Filtre passe-bas de type CLC . . . . .	33
19	Bloc du traitement du signal avant l'échantillonnage. . . . .	34
20	Problématique si nous ne filtrons pas le signal passe-bas avant d'échantillon- ner.Source [15] . . . . .	34
21	Diviseur de tension. . . . .	35
22	Additionneur et inverseur . . . . .	36
23	Réponse de l'additionneur et de l'inverseur . . . . .	36
24	Schéma du redresseur double alternance . . . . .	37
25	Réponse du redresseur double alternance . . . . .	38
26	Filtre passe-bas du 3ème ordre selon Sallen-Key . . . . .	40
27	Réponse en fréquence du filtre passe-bas de 3ème ordre avec une fréquence de coupure de 100Hz. . . . .	42
28	Lecture dans l'EEPROM. (Source : Datasheet de l'EEPROM). . . . .	43
29	Ecriture dans l'EEPROM. (Source : Datasheet de l'EEPROM). . . . .	43
30	Machine d'état de l'algorithme . . . . .	45
31	Courbe simplifiée . . . . .	46
32	Demi-triangle . . . . .	47
33	Courbe perturbatrice en orange . . . . .	47
34	Signal de départ et échantillonnage après filtrage . . . . .	48
35	Schéma du second algorithme . . . . .	49
36	Signal traité avec le second algorithme. . . . .	50
37	Schéma du troisième algorithme . . . . .	51
38	Signal traité avec le troisième algorithme. . . . .	52
39	Signal avec une fréquence de coupure à 5Hz. . . . .	53
40	Signal avec une fréquence de coupure à 1Hz. . . . .	54
41	Algorithme de détection des seuils. . . . .	55



42	Blocs de l'architecture à l'intérieur de la FPGA. . . . .	56
43	Simulation I2C . . . . .	58
44	Trame I2C simulée. . . . .	59
45	Trame I2C mesurée par un oscilloscope. . . . .	59
46	Test LRCK : 31250kHz. . . . .	60
47	Test BCK : 2MHz. . . . .	60
48	Simulation de la communication RS-232 . . . . .	61
49	Simulation de la communication RS-232 . . . . .	61
50	Blocs pour la configuration du convertisseur AD. . . . .	62
51	Blocs pour l'acquisition des données. . . . .	62
52	Simulation de la communication avec le convertisseur AD. Valeurs de 16bits récupérées. . . . .	63
53	Blocs pour le traitement des données. . . . .	64
54	Simulation de l'algorithme des seuils. . . . .	65
55	Blocs internes au bloc mémoire. . . . .	66
56	Signal sinusoïdal redressé. . . . .	69
57	Signal sinusoïdal après avoir été filtré passe-bas. . . . .	69
58	Signal sinusoïdal à après avoir été filtré passe-bas. . . . .	70
59	Autre possibilité pour la détection de maximums. . . . .	73
60	Divers type de géophones, nous prenons le LGT-4.5. . . . .	82
61	Sortie par rapport à la fréquence du LGT-4.5 . . . . .	83
62	Interface Sub-D 15 . . . . .	88
63	Liste des FGPA IGLOO . . . . .	90
64	Block I2C . . . . .	95
65	Block I2C . . . . .	96
66	Signal récupéré . . . . .	97
67	Signal redressé . . . . .	98
68	Signal filtré passe-bas 3ème ordre à 100Hz . . . . .	99
69	Réponse de l'algorithme pour les maximums . . . . .	100
70	Réponse de la moyenne uniquement . . . . .	101
71	Niveau logique de dépassement d'amplitude . . . . .	102
72	Ensemble de courbes . . . . .	103
73	Filtrage des valeurs maximales et sous-échantillonnage . . . . .	104
74	Filtre passe-bas du 3ème ordre selon Sallen-Key . . . . .	106
75	Gamme de tensions pour le filtre passe-bas . . . . .	107
76	Réponse en fréquence du filtre, dans les basses fréquences. . . . .	107
77	Réponse en fréquence du filtre sur une grande plage de fréquence. . . . .	108
78	Réponse en fréquence du filtre passe-bas de 3ème ordre avec une fréquence de coupure de 100Hz. . . . .	110
79	Tension maximale de sortie du géophone Dynamic Technologies DT20DX 4.5Hz . . . . .	112
80	Tension minimale de sortie du géophone Dynamic Technologies DT20DX 4.5Hz . . . . .	112
81	Schéma du convertisseur AD : PCM3794 . . . . .	113
82	Gamme de tension pour le filtre passe-bas . . . . .	115
83	Réponse en fréquence du filtre, dans les basses fréquences. . . . .	115
84	Réponse en fréquence du filtre sur une grande plage de fréquence. . . . .	116

## 1 Lexique

Différents termes spécifiques au projet sont présents dans ce rapport et ces derniers sont expliqués dans cette section.

- Echantillonnage : Prendre la valeur d'un signal à un instant  $t$  et à ses multiples de  $k$ .
- GSM : Global System for Mobile Communication
- CMOS : Complementary Metal Oxide Semiconductor
- TTL : Transistor-Transistor Logic
- Sps : Sample per Second, c'est le nombre de "symboles" par seconde. Chaque nombre est appelé "sample".
- SNR : C'est le rapport signal sur bruit. C'est un indicateur de qualité de la transmission d'une information.
- I2C : Inter Integrated Circuit, Protocol de communication
- SPI : Serial Peripheral Interface, Protocol de communication
- UART : Universal Asynchronous Receiver Transmitter
- I2S : Integrated Interchip Sound
- FPGA : Field-Programmable Gate Array
- CPU : Central Processing Unit
- LDO : Low drop out regulator
- FIFO : First In First Out
- EEPROM : Electrically-Erasable Programmable Read-Only Memory
- MOSFET : Metal Oxide Semiconductor Field Effect Transistor
- BJT : Bipolar Junction Transistor

## 2 Introduction

La montagne est un espace plein de vie, qui ne cesse de changer. De nouveaux cours d'eau se créent, d'autres disparaissent. La nature est en changement constant mais il existe de nombreux dangers ; que ce soit des éboulements, des avalanches, des barrages qui cèdent sous la pression, des tremblements de terres, des tempêtes et leurs conséquences peuvent avoir des effets parfois dévastateurs.

Le CREALP (Centre de Recherche sur l'environnement ALPin) est une fondation basée à Sion et créée en 1968 par le canton du Valais et la commune de Sion. Cet organisme de recherche s'intéresse aux dangers naturels comme les laves torrentielles, les crues, les glissements de terrain, les avalanches, aux ressources en eau, à la gestion des nappes phréatiques. Dans une dynamique de recherche le CREALP est en perpétuel évolution et sans cesse à la recherche de nouveaux procédés et méthodologies. Dans la prévisions des laves torrentielles, ou pour la mesures de charriage, des géophones sont utilisées pour mesurer les vibrations émises dans le sol par les différents phénomènes. A la frontière entre le monde académique, institutionnel et fédéral, le CREALP a fait appel à la HES pour répondre à des problématiques techniques liées à la fréquence d'acquisition de ce type d'appareil. L'interaction de ces deux organismes permet ainsi de « merger » les compétences académiques, la formation et les problématiques appliquées.

Dans notre projet, nous allons plus particulièrement nous intéresser au géophone. C'est un appareil comportant une masse qui est entourée d'une bobine. Plus cette masse bouge, plus les signaux qui sortent de l'appareil sont importants.



FIGURE 1 – Géophone DT20DX [2]

### 2.1 Contexte

Le projet s'articulera autour des géophones qui seront potentiellement utilisés au sein de plans d'alarme. Ces appareils ont une fréquence d'acquisition très importante, et leur utilisation nécessite un post traitement gourmand en calcul et qui est généralement effectué par des ordinateurs. Ceci nécessite alors un équipement important ce qui limite son utilisation sur le terrain. Le but de ce projet sera donc de développer un prototype facilement déplaçable, robuste et qui puisse rester plusieurs semaines sans aucune intervention humaine.

Le but de l'appareil étant d'effectuer à moindre coût énergétique le post traitement qui consiste à transmettre une indication du signal d'origine. Ceci permet une transmission

plus facile.

Ce travail consiste donc à étudier un signal, l'analyser au moyen de diverses opérations mathématiques et de pouvoir en tirer des informations précises et pertinentes. Une carte prototype sera créée afin de pouvoir récupérer les données d'un géophone. Cette carte doit pouvoir échantillonner un signal à une fréquence minimum de 10kHz, soit 10'000 points par seconde. Puis elle doit analyser le signal reçu et enfin elle doit pouvoir transmettre via un module GSM des données à un serveur délocalisé.

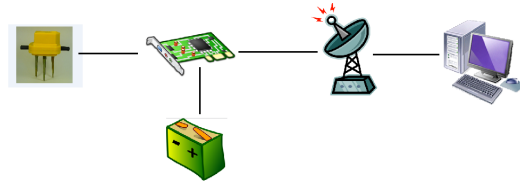


FIGURE 2 – Schéma des parties du projet.

Un des points importants de ce projet est de limiter la consommation totale du système. En effet, en milieu montagnard, la ressource énergétique est un facteur limitant.

## 2.2 Cahier des charges et objectifs

Le cahier des charges fourni par le CREALP nous impose que les mesures doivent être échantillonnées à plus de 10kHz. La consommation totale du système doit être relativement faible ne dépassant pas 50mW. Le système étant déposé à l'extérieur, il sera soumis à des conditions climatiques très diverses, il faut le protéger grâce à une boîte hermétique. Les données seront stockées puis transférées par GSM à un serveur.

Les objectifs de ce travail sont :

- La réalisation d'une carte FPGA basse consommation.
- Le développement d'un algorithme de traitement de l'information.
- La réalisation d'un démonstrateur.

Nous allons ensuite nous intéresser au géophone : Découvrir son fonctionnement, réaliser quelques tests afin de pouvoir récolter quelques informations utiles. Nous nous intéresserons ensuite au matériel qui sera utilisé pour créer le prototype.

### 3 Matériel et Méthode

Notre but est de créer une carte permettant d'acquérir des données provenant d'un géophone. Nous allons tout d'abord comprendre le fonctionnement des géophones. Puis nous établirons la liste des composants dont nous aurons besoin pour pouvoir créer notre carte d'acquisition.

Nous parlerons dans cette section du matériel que nous avons utilisé. Les géophones fournis par le CREALP et les éléments plus spécifiques à la partie ingénierie, que ce soit le prototype ou les algorithmes du traitement du signal.

#### 3.1 Matériels et logiciels utilisés

Pour ce projet, nous avons utilisé plusieurs logiciels.

- Traitement du signal en python : Spyder 2.2.0
- Schématique électronique : P-CAD 2006
- Simulation : HDL Designer Version 2012.2
- Chargement sur la FPGA : Libero V11.0

Nous utilisons pour les mesures les appareils suivants :

- Oscilloscope : Agilent Technologie, InfiniiVision, MSO-X 3012A
- Alimentation : DC Power, TPS-4000
- Multimètre : Fluke 183

#### 3.2 Les géophones

Un géophone est un capteur dont le but est de mesurer des mouvements oscillatoires.

##### 3.2.1 Fonctionnement d'un géophone

Composé d'une bobine et d'une masse magnétique, la mise en mouvement génère un courant électrique. La tension émise est proportionnelle au mouvement qui agite le géophone.

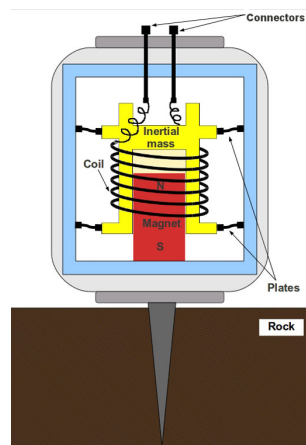


FIGURE 3 – Géophone avec la masse magnétique

Cette figure représente la coupe d'un géophone. (Source[5]).

*La réponse en fréquence d'un géophone est celle d'un oscillateur harmonique, qui se situe souvent aux alentours des 10Hz. Source [6]*

### 3.2.2 Géophone du CREALP

Le CREALP a mis à disposition un géophone afin de pouvoir faire des mesures. Ce dernier est un Dynamic Technologies DT20DX[3] 4.5 Hz Vertical Sensor. La résistance de la spire interne est de  $395[\Omega]$ . Cette spire génère une tension de sortie de  $23.4V/m/s$ .

Dans une vision d'avenir, le CREALP pense investir dans un nouveau géophone : Baoding Longet Equipments 4.5Hz-3Component LGT Sensor. La résistance de la spire est de  $375[\Omega]$ . Sa tension de sortie est de  $28.8V/m/s$ . Ce géophone mesure les déplacements dans les trois directions, X, Y et Z.

### 3.2.3 Tests réalisés sur le géophone

Quelques tests ont été réalisés sur un géophone afin de pouvoir, par la suite, traiter les signaux et pouvoir dimensionner au mieux les composants électroniques dont nous aurons besoin. Ceci est résumé dans le tableau suivant. Nous avons branché le géophone à un oscilloscope et nous avons secoué manuellement ce dernier plus ou moins rapidement. Les tensions générées par le géophone sont dans le tableau suivant.

<i>Tensions minimales et maximales</i>	
Tension maximale	16V
Tension minimale	20mV

Les figures des courbes enregistrées grâce à un oscilloscope peuvent être consultées dans l'annexe 19.

### 3.2.4 Traitement du signal

Grâce à un oscilloscope nous avons enregistré un signal émis par un géophone. Le signal généré par les géophones sont des signaux oscillatoires. Chaque secousse dans le sol émet une série de vibrations que l'on peut distinguer sur la figure 4. Les échelles sont selon les X : 1seconde/décade et en Y : 50mV/décade.

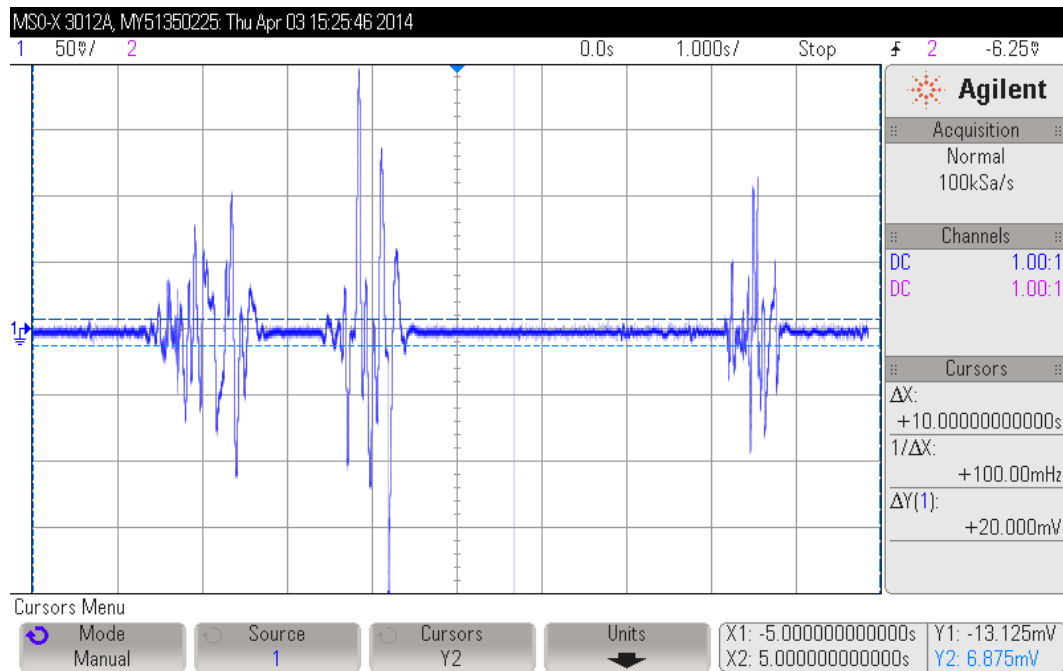


FIGURE 4 – Tension minimale de sortie du géophone Dynamic Technologies DT20DX 4.5Hz

On remarque des pics plus ou moins hauts. Ces pics indiquent que les vibrations étaient plus importantes à ce moment-là. Notre but est de pouvoir détecter ces pics et enregistrer leurs amplitudes.

On peut également remarquer que le signal oscille très rapidement dans le temps. La résolution est de 1 seconde / division. Il faut donc échantillonner à une fréquence plus élevée afin de ne perdre aucune information.

Selon le cahier des charges, la fréquence d'échantillonnage est de 10kHz. Nous prenons un convertisseur analogique-digital à 16 bits. Avec 16 bits, nous avons une bonne précision sur le signal. Ceci implique :

<i>Caractéristiques du signal</i>	
Échantillonnage	10kHz
Période	100 micro secondes
Taille données par secondes	20kbytes
Transmission	120kbytes / minute

A raison de 16 bits de données et d'une fréquence d'échantillonnage de 10kHz, nous avons :

Résumé des quantités de données	
<i>Nombres de valeurs</i>	<i>Quantité de données</i>
1/1	20KBytes / seconde
	1.2MBytes / minute
	72 MBytes / heure
	1,728GBytes / jour
1/100	200 Bytes / seconde
	12 KBytes / minute
	720 KBytes / heure
	17,28 MBytes / jour

TABLE 1 – Résumé des différents matériels

Nous constatons que la quantité de données diminue grandement si nous n'en prenons qu'une sur 100.

### 3.3 Liste du Matériel

Afin de pouvoir traiter les signaux vus précédemment, nous devons créer une carte robuste qui consomme peu d'énergie. Pour se faire, nous allons dresser la liste des composants dont nous aurons besoin pour ce projet.

#### 3.3.1 Parties principales

Voici un schéma des différentes parties que nous aurons :

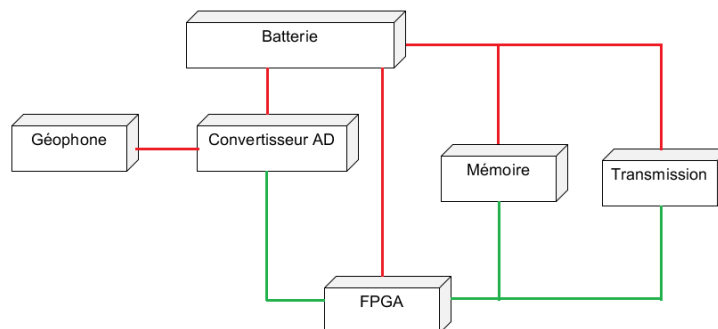


FIGURE 5 – Schéma électrique des principaux composants principaux.

En rouge nous avons la partie électrique. En vert, la partie données. Nous avons besoin de différents niveaux de tensions suivant les composants. De plus nous avons besoin d'un convertisseur analogique digital pour la conversion des signaux analogiques en nombres et une unité de traitement des données. Des batteries permettent d'alimenter la carte ainsi que ses composants. Nous allons passer en revue chacun de ces blocs et nous allons ainsi déterminer les composants dont nous aurons besoin.



### 3.3.2 Alimentation

Nous nous intéressons tout d'abord à la partie alimentation du système. Ce dernier sera autonome et il aura besoin de courant pour fonctionner. Nous aurons besoin de convertisseur DC-DC ou de régulateur afin de pouvoir générer les tensions demandées par les composants. Un connecteur d'alimentation sera également nécessaire.

### 3.3.3 Batterie

Nous avons observé plusieurs types de batteries. Les plus performantes et celles qui paraissent de meilleures qualités possèdent une capacité variant entre 14'000 et 20'000mAh. Si nous voulons tenir une année sans rien toucher, avec la batterie de 20'000mAh, il est obligatoire que le circuit consomme au maximum : 2,2mA.

En effet,  $\frac{20'000mAh}{365jours*24heures} = 2,2mA$ .

### 3.3.4 Régulateur ou Convertisseur

Un LDO ou un DC-DC :

LDO : Low drop-out : est un régulateur DC linéaire qui peut opérer sur une très petite différence entre l'entrée et la sortie du régulateur. Les avantages sont une faible tension de fonctionnement, une plus grande efficacité de fonctionnement et une tension de sortie qui reste quasi constante.

Les convertisseurs DC-DC peuvent proposer une plus grande chute de tension que les régulateurs, ils consomment moins d'énergie par contre la tension à la sortie peut osciller plus que dans un régulateur.

### 3.3.5 Connecteur d'alimentation 5V

Afin de pouvoir connecter la batterie à la carte, il faut un connecteur d'alimentation. Nous avons choisi un micro-usb (disponible chez Mouser[11] au prix de CHF 1.50.-) car ce type de connecteur est très répandu sur le marché et est compatible avec nos batteries.

### 3.3.6 Convertisseur DC-DC 5V-3.3V

Nous aurons besoin d'abaisser la tension de 5V à 3.3V afin de pouvoir alimenter la FPGA, les différents amplificateurs opérationnels, la mémoire, le convertisseur Analogique-digital ainsi que les drivers pour la communication série.

Nous avons regardé les différents régulateurs que nous pouvions trouver pour passer de 5V à 3.3V. Or les résultats proposés par Farnell sont deux convertisseurs et un régulateur. Or ce régulateur est entre 5 et 10 fois plus chère qu'un convertisseur. De plus, il peut y avoir de fort courant, aux environ de 100mA et les régulateurs ne supportent pas de tel courant.

### 3.3.7 Régulateur 3.3V - 1.5V

Afin de pouvoir alimenter le cœur de la FPGA, il est obligatoire de générer une tension de 1.5V. Cette tension est prise depuis celle de 3.3V qui elle est déjà réduite. Un régulateur a été choisi car il permet de réguler la tension et donc d'absorber les variations de tension. Ainsi la tension sera toujours la même pour le cœur de la FPGA.

### 3.3.8 Acquisition des données

Nous passons ensuite à la partie conversion. En effet, nous recevons les données d'un géophone sous forme de tensions. Nous devons donc les échantillonner afin de pouvoir par la suite les traiter avec une unité arithmétique.

### 3.3.9 Convertisseur AD

Le géophone nous fournit des signaux sous forme de tensions. Afin de pouvoir traiter ces signaux nous devons les transformer en nombres digitaux. Ceci est le but du convertisseur Analogique - Digital. Le cahier des charges nous impose une fréquence d'échantillonnage de 10kHz au minimum. Or un modèle de géophone comprend 3 canaux, il faudra donc échantillonner trois fois plus vite si les canaux sont en série, ou à 10kHz si les canaux sont en parallèles.

De plus la consommation doit être réduite afin de pouvoir tenir le plus longtemps avec une batterie sans devoir la recharger.

### 3.3.10 Diode Zener

Le convertisseur AD ne peut recevoir en entrée une tension supérieure à 4V, il faut donc la limiter, or le géophone peut générer une tension maximale de 20V. Une protection à l'aide de diodes Zener permet de protéger le convertisseur contre les surtensions.

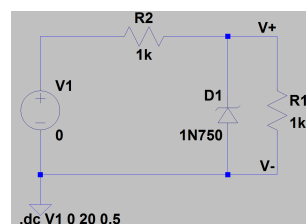


FIGURE 6 – Schéma avec la diode Zener pour la limitation de la tension

### 3.3.11 Traitement des données

Après que le convertisseur Analogique-Digital ai échantillonné les tensions, il va fournir des chiffres permettant de quantifier ces tensions. Nous avons donc besoin d'une unité permettant de décoder et de traiter ces données. Nous devons ensuite également les enregistrer car nous ne pouvons pas forcément les transmettre à un serveur tout de suite. Nous allons

parler plus en détail de l'unité de calcul ainsi que de l'unité de stockage.

### 3.3.12 FPGA

Nous avons choisi une FPGA. Etant donné que nous cherchons une faible consommation de courant (moins d'un milliampère), nous nous sommes tourné vers l'entreprise Microsemi. Il est facile d'ajouter des composants. Il a aussi moins de contraintes sur les broches que pour un processeur. Un processeur a souvent des broches spécifiques à certaines fonctions, ceci peut être un paramètre limitant. Mais surtout, elle traite les données en parallèle. Il est donc possible de pouvoir enregistrer des données d'un côté, les traiter d'un autre et enfin toujours en même temps, les envoyer. Ce qui n'est pas faisable avec un microprocesseur.

Nous avons besoin d'une FPGA qui consomme peu de courant. Les FPGAs qui ont des consommations relativement basses sont celles de MICROSEMI, la gamme IGLOO [7]. On y trouve des IGLOO Nano possédant entre 10'000 et 250'000 gates.

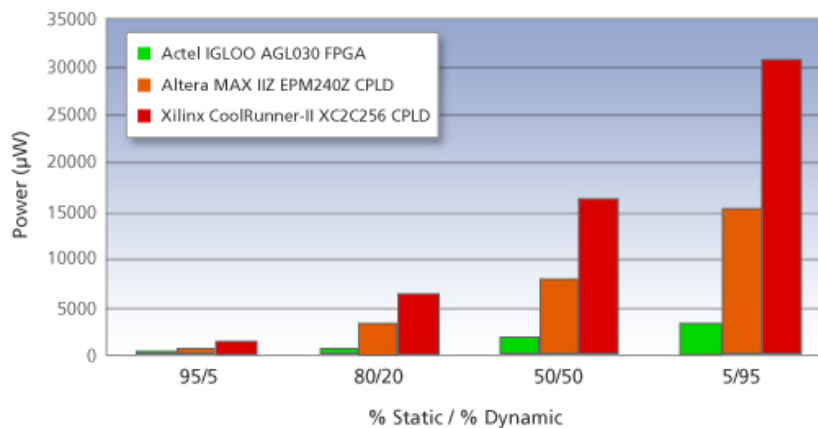


FIGURE 7 – Graphique de consommation d'une FPGA Igloo par rapport à deux autres FPGA.

Ce graphique [8] montre la puissance totale pour 3 cas :

- Première colonne : 5% dynamique et 95% statique
- Deuxième colonne : 50% dynamique et 50% statique
- Troisième colonne : 95% dynamique et 5% statique

Le graphique[12] suivant montre la puissance consommée en fonction de la fréquence.

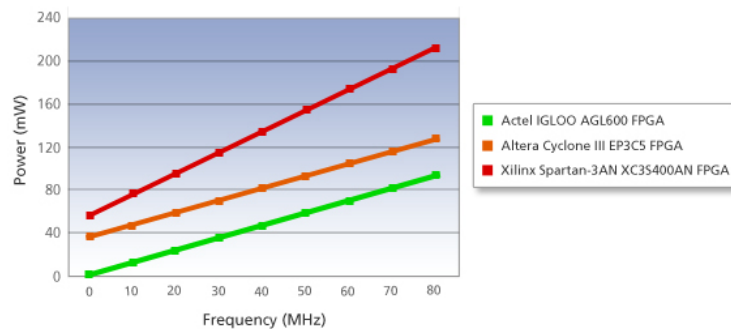


FIGURE 8 – Graphique de consommation d’une FPGA Igloo par rapport à deux autres FPGA en fonction de la fréquence.

### 3.3.13 Connecteur JTAG

Afin de pouvoir programmer la FPGA, nous aurons besoin d’un connecteur JTAG avec 10 broches, qui est également la norme à la HEI Valais c’est donc pour cette raison que ce dernier a été utilisé.

### 3.3.14 Manuel Reset

Pour pouvoir mettre à zéro les composants, nous allons utiliser un Reset. Ce dernier a pour but d’enclencher les différents composants en même temps. De plus, la tension d’alimentation doit atteindre un certain seuil avant que les composants ne démarrent. Cela permet également de réinitialiser les composants au cas où la tension passerait en dessous d’un certain seuil. On évite ainsi des problèmes lorsque la tension s’affaiblit.

### 3.3.15 Oscillateur

Nous avons besoin d’un oscillateur qui servira de clock externe pour tous les composants, ceci afin qu’ils soient synchrones. Cet oscillateur a été trouvé à l’atelier de la HEI. C’est un oscillateur DIP 14 avec une fréquence de 16MHz. Nous avons choisis 16MHz car nous avons besoin d’une certaine rapidité de calcul mais nous ne devons pas monter trop haut. Plus la fréquence d’oscillation est élevée et plus la consommation de courant est importante. C’est une puissance de 10, car d’autres oscillateurs avaient des fréquences de l’ordre de 18432Hz. Il est donc plus aisé de pouvoir diviser des fréquences comme 16MHz pour en générer d’autres et obtenir des fréquences sans chiffres après la virgule.

### 3.3.16 Mémoire

Afin de pouvoir enregistrer les données, nous avons besoin d’une mémoire. Nous avons tout d’abord choisi une carte SD, or cette dernière est trop gourmande en énergie. Il a donc fallu trouver une autre mémoire. Nous nous sommes penchés sur les EEPROM. Elles sont beaucoup moins énergivores et plus petites. Par contre leur capacité est également plus faible.

### 3.3.17 Transmission

Après avoir récupéré les données, les avoir traitées, il n'est pas très utile de garder ces données en mémoire, le but est de les transmettre à un serveur. Le CREALP possède un module GSM, qui va être utilisé pour transmettre les informations à un ordinateur. Ce module est pourvu d'un connecteur de type RS-232. C'est donc pour cette raison que nous allons opter pour ce type de communication.

### 3.3.18 Transmetteur et récepteur RS-232

Afin de pouvoir transmettre nos données au module GSM ou lors des tests à un ordinateur, nous aurons besoin d'un module de transmission. Le CREALP possède des transmetteur GSM doté d'un port série. C'est pourquoi nous choisissons la transmission RS-232. Nous avons besoin d'un driver RS-232 afin de pouvoir envoyer et recevoir les informations. La carte de prototypage est alimentée en 3.3V, or la technologie RS-232 peut monter jusqu'à 25V. Il faut donc abaisser cette tension. C'est le but du driver RS-232.

Avec ceci nous aurons besoin d'un connecteur RS-232 afin de pouvoir connecter le câble.

### 3.3.19 Transmetteur et récepteur RS-422

Lorsque nous avons un géophone sur le terrain, il est possible que certaines données soient erronées, si, par exemple, un animal passe à proximité du géophone. Pour pallier à ce problème, il faudrait disposer de plusieurs géophones couplés les uns aux autres. Ceci permet de pouvoir comparer les données. Si tous les géophones ont le même type de données à un instant, c'est qu'il est probable qu'une secousse sismique en soit à l'origine. Dans le cas contraire, il peut s'agir d'un animal.

Le bus RS-422 ressemble fortement au RS-232. Il permet de transmettre des données sur de plus longues distances. Ceci est possible grâce au fait que les lignes sont différentielles.

Afin de pouvoir adapter ces niveaux de tension, nous avons besoin de drivers RS-422.

### 3.3.20 Connexion SUB-D 15

Pour relier les nouveaux géophones (de type Baoding Longet Equipments 4.5 Hz 3 ) à la carte, il est nécessaire d'avoir un interface SUB-D(D-subminiature) 15 ou connecteur VGA afin de pouvoir récupérer les données.

### 3.3.21 Transistor

Afin de pouvoir activer ou désactiver le module GSM qui consomme beaucoup de courant, environ 300mA, il faut trouver un moyen de l'activer et de le désactiver. Ceci sera fait grâce à un MOSFET. Un MOSFET est commandé en tension et permet de laisser ou non passer du courant. Un BJT sera commandé en courant, ce qui est plus difficile à mettre en oeuvre. C'est pourquoi le MOSFET sera privilégié.

### 3.4 Méthode de calcul / Algorithme

Nous recevons de la part du convertisseur AD de nombreuses valeurs correspondant aux tensions générées par le géophone. Or la quantité de valeurs est très importante, plus de 10'000 par seconde. Il est donc impératif de traiter ces données grâce à un algorithme.

Les tensions des géophones sont converties en nombre afin de pouvoir être quantifiée. Ce signal est de type continu, totalement aléatoire et non-prédictible.

Nous débutons notre analyse en observant le signal. Nous constatons les éléments suivants :

Le signal n'est pas périodique, il est composé d'amplitudes positives et négatives. Ces amplitudes peuvent varier fortement et de façon spontanées.

Nous cherchons à détecter les pics d'amplitude, le fait que l'amplitude soit positive ou négative n'a aucune importance, nous la cherchons en absolu. Les amplitudes seront ensuite comparées à des seuils. Puis le résultat sera stocké et envoyé via le port RS-232.

Puis nous cherchons à connaître les fréquences possibles du signal en utilisant la FFT. Cette fonction est la transformée rapide de Fourier. Elle permet de trouver les fréquences d'un signal (toutes les composantes). Nous avons trouver des fréquences qui se répète aux alentours des 10Hz, mais sinon il y avait de nombreuses fréquences. Il est donc difficile de pouvoir sortir une information précise à partir de cette analyse.

Connaissant cela, nous avons pu mettre en place des algorithmes qui se basent sur ces informations.

Nous passons maintenant à la partie qui concerne les résultats. Nous parlerons des différents composants et la raison de leur sélection, ainsi que des différentes parties du circuit. Nous terminerons avec les algorithmes et le code VHDL.

## 4 Développement

Nous allons maintenant choisir et expliquer les différents choix concernant les composants qui seront implémentés sur la carte électronique. Avant toute chose, nous devons connaître dans les grandes lignes la puissance de calcul que doit avoir notre carte pour que le tout puisse fonctionner correctement.

### 4.1 Puissance de calcul

Nous devons effectuer du traitement de signal ainsi que l'acquisition de données. Ceci doit se faire dans un certain "laps" de temps.

Le cahier des charges indique que nous devons échantillonner à au moins 10kHz. Le convertisseur AD sérialise les données et les envoie à la FPGA. Il est donc nécessaire d'augmenter la vitesse de l'oscillateur afin de pouvoir transmettre toutes les données. Donc, nous aurons une fréquence minimale d'horloge de :  $\text{Nombre\_de\_bits} * \text{fréquence\_échantillonnage}$ .

En tenant compte de ces différents éléments, nous pouvons déduire les informations suivantes :

Matériel : Résumé		
<i>Composant</i>	<i>Fréquence</i>	<i>Fréquence minimal</i>
PCM3794	10kHz	3 canaux * 10kHz = <b>30kHz</b>
EEPROM	400kHz	-
Oscillateur	1-150MHz	16MHz

TABLE 2 – Résumé des différents matériels

Nous devons échantillonner à une fréquence trois fois supérieure à celle mentionnée dans le cahier des charges. Le fait est que le convertisseur AD transmet les données de chacun des canaux un après l'autre. Si nous voulons avoir un échantillon par canal chaque 100us, nous devons échantillonner à 33us chaque canal.

### 4.2 Matériel et schéma électrique

Nous allons maintenant expliquer brièvement le choix de certains composants ainsi que leur configuration dans les schémas électriques. La conception du schéma électrique consiste à relier tous les modules entre eux afin de pouvoir réaliser les fonctions voulues. Nous avons tout d'abord pris les composants disponibles à l'atelier de la HEI et commandés ceux qui ne l'étaient pas.

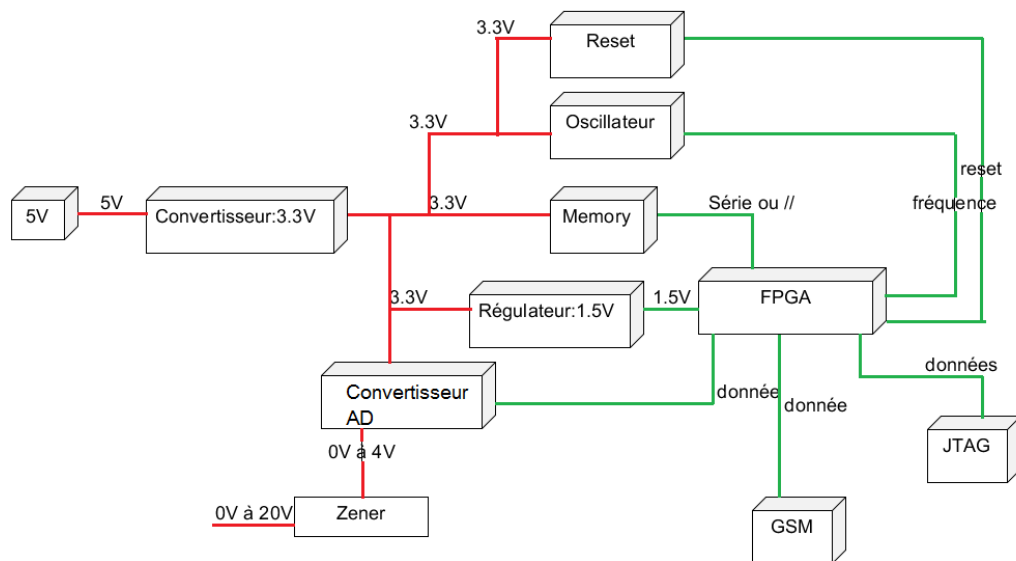


FIGURE 9 – Schéma du circuit.

Nous retrouvons les différents composants de notre circuit. En rouge nous avons la partie alimentation et tensions. En vert, la partie orientée données.

#### 4.2.1 Alimentation

L'alimentation se fait par le biais d'une batterie ayant une capacité supérieure à 10Ah. Ce type de batterie est pourvu d'un connecteur micro-usb. Une batterie avec une grande capacité augmente la durée de fonctionnement sans intervention humaine.

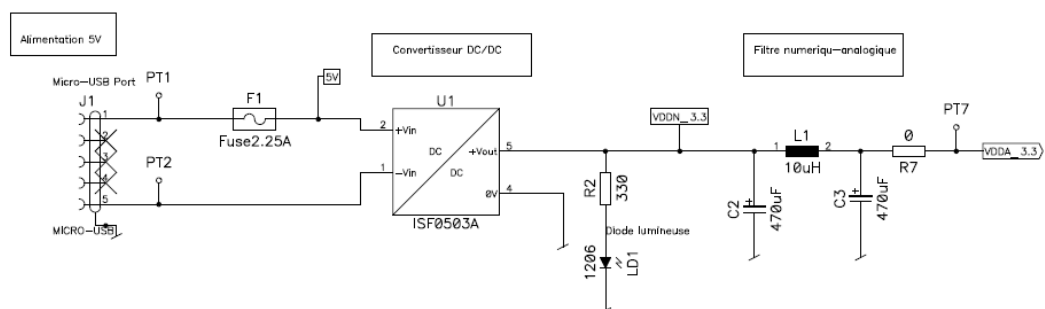


FIGURE 10 – Schéma du bloc alimentation

Un fusible se trouve tout au départ. Il permet de protéger le circuit si le module GSM demande beaucoup de courant. Sur un très court instant, il peut demander entre 2 et 3[A].

Nous avons ensuite un convertisseur DC-DC qui est isolé. Ce dernier possède une isolation galvanique qui permet de découpler les deux circuits (circuit "réseau" ou d'alimentation et le circuit interne). Ceci permet de séparer les deux circuits et donc d'empêcher que



des tensions passent d'un côté à un autre. Cela permet également de "couper" le courant alternatif.

Vient ensuite une diode lumineuse qui permet simplement de voir s'il y a bien une tension en sortie du convertisseur.

La dernière partie est un filtre LCL qui a pour but de lisser la tension.

#### 4.2.2 Convertisseur DC-DC

Nous avons besoin d'un convertisseur DC-DC afin de réduire la tension d'entrée. Nous optons pour le modèle suivant :

##### **XP POWER - ISF0503A - CONVERTISSEUR DC-DC BOITIER CMS 1W 3.3V**

Nous avons choisi le convertisseur XP Power pour les raisons et ses caractéristiques suivantes :

Choix du convertisseur DC-DC		
<i>Désignation</i>	<i>Valeur</i>	<i>Raisons</i>
Puissance maximale	1W	Ce qui donne un courant maximum de 300mA. Nous avons estimé le courant à 60mA (40mA pour l'oscillateur et 10mA pour la diode lumineuse). Mais nous prenons un peu de marge au cas où le circuit tirerait plus de courant. Le convertisseur à 75mA était donc un peu trop proche de la valeur estimée.
Tension d'alimentation	entre 4.5 et 5.5V	Nous avons des batteries qui génèrent 5V.
Température de fonctionnement	-40 à +85 degrés	Pour la vie en montagnes.

TABLE 3 – Raisons du choix du convertisseur DC-DC.

Il nous faut donc, pour avoir 3.3V en sortie, un modèle dont le numéro de série est : ISF0503A. On va prendre le standard qui est du package : B1 (Informations sur les broches dans l'annexe 5) car nous aurons besoin d'une seule sortie (possibilité d'avoir des sorties doubles) et une tension d'isolation de 1kV est suffisant.

#### 4.2.3 Alimentation FPGA Régulateur 3.3V vers 1.5V : TPS70402

Le régulateur permet de réguler la tension. Il a été choisi car il était disponible à l'atelier de la HEI.

**Texas Instrument : TPS70402**

Choix du régulateur		
<i>Désignation</i>	<i>Valeur</i>	<i>Raisons</i>
Courant consommé	2uA	Nous cherchons d'après le cahier des charges, à limiter la consommation et ce convertisseur consomme peu.
Tension de sortie	ajustable	La sortie peut être ajustée grâce à des résistances.
Température de fonctionnement	-40 à +85 degrés	Pour la vie en montagnes.
Disponibilité		Il est disponible de suite à l'atelier d'électronique de l'école.

TABLE 4 – Raisons du choix du régulateur.

C'est un LDO, soit un "Low-Dropout voltage regulator". Le régulateur à une faible chute de tension en sortie et cette tension reste très stable. Malheureusement, il a tendance à consommer plus de courant qu'un convertisseur DC-DC. Nous avons pris un régulateur car nous avons besoin d'une tension stable pour la FPGA.

Afin de pouvoir obtenir la tension désirée, il est capital de choisir les bonnes résistances. Les résistances sont choisies grâce à l'équation suivante :

$$R1 = \left( \frac{V0}{Vref} - 1 \right) * R2$$

Le détail du calcul peut être consulté dans l'annexe 4.

Composants externes au TPS70402		
<i>Composant</i>	<i>Calculé</i>	<i>Normalisé</i>
R2	30.1[kΩ]	18[kΩ] + 12[kΩ] + 100[Ω]
R1	6.7[kΩ]	6,7[kΩ]
C4	22[uF]	

TABLE 5 – Composants externes au régulateur.

#### 4.2.4 FPGA

La FPGA est l'unité de calcul, il faut donc une unité relativement puissante et qui ne consomme pas trop d'énergie. La firme microsemi propose de nombreuses Igloo Nano différentes qui sont peu gourmandes en énergie. Ces Igloos nano sont plus ou moins importantes au niveau de la quantité de mémoire dont elles disposent ainsi que du nombre de gates disponibles.



FIGURE 11 – FPGA Igloo Nano de MICROSEMI.

**Entrées-Sorties :** Afin de pouvoir échanger les FPGAs, nous avons dressé une liste des entrées-sorties qui sont placées de manière identique sur les AGL060, AGL125 et AGL250. Ceci afin de connecter tout d'abord les broches qui sont identiques aux trois FPGAs. Ainsi nous pouvons placer une autre FPGA sans devoir modifier la schématique.

Afin de faire le bon choix quant à la FPGA, il est utile de savoir le taux d'occupation du programme.

Etant donné que nous ne savons pas encore la dimension du programme que nous aurons et qu'en sachant que la FPGA(AGL060) du Kart ( Summer School de première année à l'HEI) est déjà bien remplie, nous allons opter pour une FPGA plus importante, soit la "AGL125", soit la "AGL250" [1] et [4]. De plus, ces FPGA sont interchangeables et de même dimension, il est donc aisé d'en prendre une plus grande si celle que nous avons prise est saturée.

Nous pouvons opter, par exemple, pour :

**ACTEL - AGLN125V2-VQG100 - FPGA,LOW-POWER,IGLOO,100VQFP**

qui peut être commandée chez Farnell[9] au prix de CHF : 24.35.- Ce prix est d'ailleurs bien inférieur au prix du kit de développement comportant la même FPGA ( environ 100 dollars).

Cette FPGA a été choisie pour les raisons suivantes :

Choix de la FPGA		
<i>Désignation</i>	<i>Valeur</i>	<i>Raisons</i>
Consommation	3-4mA	C'est une FPGA qui consomme peu de courant, nous l'avons vu en détail au dessus.
RAM	36KBits	Possibilité de stocker au maximum 2250 nombres de 16bits.
I/O	71	Le nombre d'entrées sorties utilisable par l'utilisateur est important et de plus elles ne sont pas pré-configurée et donc bloquée pour certaines fonctions seulement.
Fréquence	Max : 250MHz	Nous voulions au minimum 16MHz ce qui est donc satisfaisant.
Température de fonctionnement	-20 à +70 degrés	Spécifié par le cahier des charges que les éléments doivent supporter des températures extrêmes.
Compatibilité	CMOS et TTL	Compatible avec les drivers RS-232( qui est spécifié par le cahier des charges) ainsi que le RS-422.
Tension d'alimentation	1.5V pour le coeur 3.3V pour la FPGA	La tension de 3.3V est identique aux autres composants. Par contre celle de 1.5V sera à générer.

TABLE 6 – Raisons du choix de la FPGA.

Le connecteur qui va permettre de debugger le système est le JTAG, et ces broches sont définies sur la FPGA. Ce connecteur est le standard à la HEI c'est pour cette raison que nous l'utilisons.

**Brochage, connexion des broches :** Plusieurs I/Os de la carte sont utilisées. Les broches choisies sont les mêmes pour les trois FPGAs : Soit les AGL060, ALG125 et AGL250. Le but est de prendre uniquement les pattes communes aux trois FPGAs afin de pouvoir les changer si besoin est.

#### 4.2.5 Reset MAX811

Le Reset est identique à celui du projet Kart de la HEI. Il s'agit du :

#### MAX811 TEUS-T

Il a été choisis également pour les raisons suivantes :

Choix du Reset MAX811		
<i>Désignation</i>	<i>Valeur</i>	<i>Fonction</i>
Tension de seuil	3.08V	La tension de seuil est proche de la tension VCC ce qui permet de ne pas enclencher la FPGA lorsque l'on n'a pas toute la tension.
Reset	$VCC < 1V$	Si la tension est inférieure à 1V, il reset. Il ne faut pas une tension de seuil plus élevée car le coeur peut être alimenté en 1.5V voir en 1.2V.
Température de fonctionnement	-40 à + 85 degrés	Doit pouvoir fonctionner à des températures extrêmes. Ceci est mentionné dans le cahier des charges.

TABLE 7 – Choix du reset MAX811.

Ce Reset est composé d'une pull-up en entrée et d'un condensateur de découplage qui permet de lisser la tension. En sortie, nous avons une pull-down qui permet de tirer la ligne à zéro. La sortie est active à l'état bas et les conditions internes au Reset sont satisfaites, la pin passe à 1.

Il faut rajouter une capacité de découplage de 0.1uF entre la broche MR et la masse afin d'être immunisé par rapport au bruit additionnel. De plus, une capacité de 0.1uF montée le plus près possible de VCC offre une immunité transitoire supplémentaire.

#### 4.2.6 EEPROM

L'EEPROM est la mémoire morte de la carte, elle permet de sauver les données que l'on récupère du convertisseur après leur traitement.

**EEPROM 24AA1026-I/SM** de chez Farnell.

Cette mémoire est choisie pour les raisons suivantes :

Choix de l'EEPROM		
<i>Désignation</i>	<i>Valeur</i>	<i>Raisons</i>
Consommation	standby : 5uA lecture : 450uA écriture : 5mA	Bien plus faible qu'une carte SD qui consomme entre 30 et 90mA.
EEPROM		Mémoire effaçable rapidement, correspond au temps d'écriture d'une nouvelle donnée.
Capacité	1MB	Capacité importante pour pouvoir stocker suffisamment de données avant de les envoyer. A raison de données codées sur 16 bits nous pouvons en enregistrer 500'000.

TABLE 8 – Raisons du choix de la mémoire.

Quelques remarques importantes concernant la configuration de certaines broches de la mémoire.

Broches EEPROM		
<i>Broche</i>	<i>Connectée à</i>	<i>Fonction</i>
SDA	VCC avec $R=2[k\Omega]$	Pour des fréquences comprises entre 400Hz et 1Mhz car nous dialoguons avec une fréquence de 400kHz.
WP	la masse	écriture possible
A0 et A1	la masse	adresse de l'EEPROM

TABLE 9 – Configuration de certaines broches de la mémoire.

#### 4.2.7 Convertisseur AD :

Le convertisseur AD choisi précédemment avait un défaut, il ne pouvait convertir qu'une seule entrée, ce qui allait bien pour le géophone actuel. Or la société CREALP est sur le point d'acheter un nouveau géophone muni d'un connecteur SUB-D 15 comportant 3 paires de signaux différentiels. Ces signaux sont donc à mesurer. Le problème avec le AIC-111RHB c'est qu'il ne permette qu'une entrée, il faut en poser trois afin d'avoir le résultat escompté. Nous nous sommes tournés vers une autre solution, un autre convertisseur AD :

**Texas Instruments PCM3794ARHBT**, Source [14]

Il a été sélectionné grâce à ces caractéristiques :

Choix du convertisseur AD		
<i>Désignation</i>	<i>Valeur</i>	<i>Raisons</i>
Consommation	Max : 2-3mA :	Faible pour limiter la consommation totale.
Fréquence échantillonnage	Min : 5kHz Max : 50kHz	Nous devons échantillonner à 30kHz car nous avons trois canaux à 10kHz chacun, ceci est donné par le cahier des charges.
Entrées différentielles	3	Nous avons un géophone avec trois axes, il nous faut donc trois entrées.
Résolution	16bits	Nous avons ainsi une grande plage de nombre possible et donc une précision accrue (39.67uV/pas).
Taux de conversion	Max : 50kSps	Nous avons besoin de 30kSps (3 fois 10'000 symboles par seconde)
Compatibilité	I2C et SPI	La HEI possède des modules VHDL pour chacun de ces protocoles.
Température de fonctionnement	-40 à + 85degrés	Doit pouvoir fonctionner dans des conditions difficiles, ceci est donné par le cahier des charges.
Disponibilité	de suite	Il est disponible de suite alors que d'autres ont 6 semaines de délais.

TABLE 10 – Raisons du choix du convertisseur AD.

C'est un convertisseur 16 bits. nous avons  $2^{16} = 65'536$  échelons. Nous avons des filtres qui précèdent le convertisseur et ces derniers possèdent des diodes. Ces dernières ont des chutes de tension de 0.7V. De plus les amplificateurs opérationnels présents ne peuvent sortir au maximum que 3.3V. Or nous devons enlever la chute de tension des diodes qui est de 0.7V, et nous avons donc des tensions comprises entre 0V et 2.6V. Dans ce cas, nous avons une résolution de :  $\frac{2.6V}{65'536} = 39.67\mu V/\text{pas}$ .

Le convertisseur AD a de nombreuses broches et leurs configurations peuvent être consultées en annexe 20.

#### 4.2.8 Diode Zener

Afin de limiter la tension à l'entrée du convertisseur AD, nous plaçons une diode Zener. Dès que le courant dépasse la tension maximale, l'effet avalanche se produit (Courant très élevé).

Le but est que la diode Zener ne doit jamais être bloquée car dans ce cas on aurait un circuit ouvert. Si le montage consomme trop de courant, la chute de tension dans la résistance va être trop importante et donc la diode Zener va se bloquer (Tension inférieure à la tension de seuil de la diode, donc le courant sera nul).

Nous aurons donc une résistance  $R = 2400[\Omega]$  et une puissance de 2.2mW. (Le détail du calcul peut être vu dans l'annexe 18).

Un test a été fait sur "LTSpice" afin de vérifier le bon fonctionnement du circuit. Voici donc le résultat :

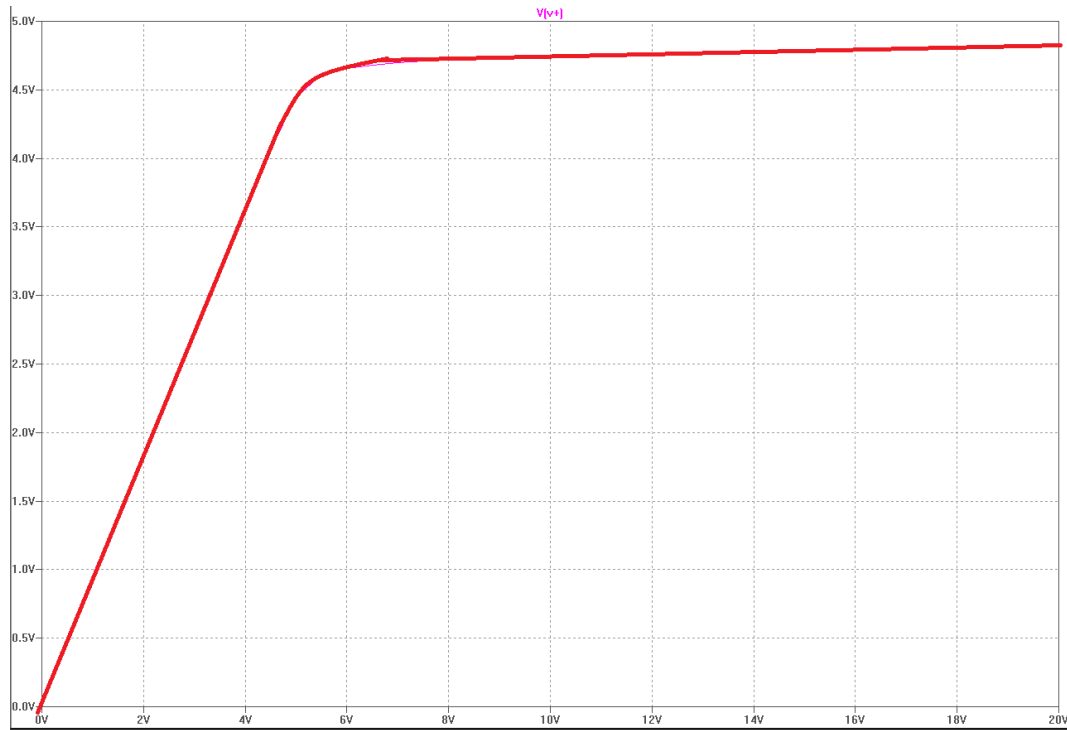


FIGURE 12 – Simulation du circuit contenant la diode Zener.

Nous avons besoin d'une tension de 4V, or, ici la tension y est supérieure. Ceci est simplement dû au fait que la diode choisie a une tension Zener de 4.7Volt et que la diode désirée n'était pas présente dans le package des éléments pour LTSpice. Mais nous pouvons constater que la tension est bien régulée au maximum à 4.7Volt malgré les 20Volts à l'entrée.

#### 4.2.9 TTL et CMOS

Étant donné que le CREALP possède un module GSM avec un connecteur RS-232, nous prenons cette même connectique pour la carte. Il faut adapter les niveaux logiques sans quoi, la transmission pourrait ne pas fonctionner.

Il existe deux technologies pour les drivers de transmission. La technologie CMOS (Complementary Metal Oxide Semiconductor) et la technologie TTL (Transistor-Transistor Logic). Il y a certaines différences entre les deux technologies et si elles sont mélangées, il est possible que le fonctionnement en soit altéré. Les niveaux de tensions sont différents pour les deux technologies.



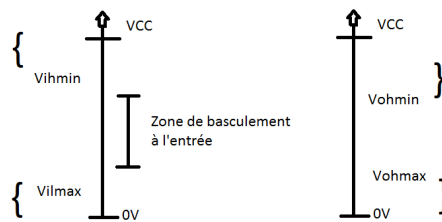


FIGURE 13 – CMOS et TTL

Niveau de tension pour les technologies TTL et CMOS		
Repère	TTL	CMOS
Vohmin	2.4 V	$0.95 * VCC$
Vihmin	2 V	$0.55 * VCC$
Volmax	0.4 V	$0.05 * VCC$
Vilmax	0.8 V	$0.45 * VCC$

TABLE 11 – Résumé des niveaux de tension.

Dans notre cas, le transmetteur-récepteur RS-232 peut être utilisé soit avec la technologie CMOS, soit avec la technologie TTL. Nous optons pour la technologie CMOS car c'est la plus répandue.

#### 4.2.10 Transmetteur et récepteur RS-232 ICL3221E

Nous avons besoin d'un driver RS-232 afin de pouvoir envoyer et recevoir les informations. La transmission se fait entre la FPGA et le module GSM qui sera implémenté plus tard. Ce composant

**INTERSIL - ICL3221EIAZ - IC, TRANSMITTER RECEIVER, RS-232**  
 Référence [10].

a été choisi pour les caractéristiques suivantes :

Choix du driver RS-232.		
<i>Désignation</i>	<i>Valeur</i>	<i>Fonction</i>
Type	émetteur et récepteur	Nous pouvons ainsi envoyer et recevoir des données. Notamment utile lors des phases de tests pour pouvoir connecter à un ordinateur.
Tension d'alimentation	comprise entre 3V et 3.6V	Compris entre 5V qui est notre tension maximale et 1.5V qui est notre tension minimale.
Température de fonctionnement	-40 à + 85 degrés	Doit pouvoir fonctionner à des températures extrêmes. Ceci est mentionné dans le cahier des charges.
Taux de conversion	460Kbps	Nous avons besoin de 4,8Kbps si nous prenons 300 échantillons par seconde à 16bits chacun.
Courant d'alimentation	0.3mA	Le courant d'alimentation est aussi faible que possible afin de limiter la consommation.

TABLE 12 – Choix du driver RS-232.

Des informations supplémentaires sont disponibles en annexe 12.

Le RS-232 est une interface permettant la communication série. En effet, pour la télécommunication par GSM il faut pouvoir envoyer les données au transmetteur et ceci via le port COM.

Le connecteur est de type coudé car il permet de pouvoir brancher une prise depuis le côté, ce qui est plus pratique que depuis le dessus. Le but étant de limiter l'espace pris par la carte. Ce composant a été trouvé à l'atelier de la HEI.

Des informations supplémentaires concernant les broches sont disponibles en Annexe 13.

#### 4.2.11 Driver RS-422

Le RS-422 comporte des entrées différentielles qui servent de récepteur. Lorsqu'il s'agit de transmettre de l'information, il faut utiliser un autre système. On ne peut transmettre directement des valeurs, il faut mettre les lignes dans certains états logiques afin de pouvoir transmettre des informations. La table de vérité qui suit indique de quelle façon il faut activer les lignes pour avoir le résultat escompté.

Le driver choisi pour le RS-422 est le suivant :

**ISL83491IBZ-IC Transceiver RS-485/422[13]**

Ce composant a été choisi pour les caractéristiques suivantes :

Choix du driver RS-422.		
<i>Désignation</i>	<i>Valeur</i>	<i>Fonction</i>
Type	émetteur et récepteur	Nous pouvons ainsi envoyer et recevoir des données. Ce qui est obligatoire si nous voulons connecter plusieurs cartes les unes aux autres.
Tension d'alimentation	comprise entre 3V et 3.6V	Compris entre 5V qui est notre tension maximale et 1.5V qui est notre tension minimale.
Température de fonctionnement	-40 à + 85 degrés	Doit pouvoir fonctionner à des températures extrêmes. Ceci est mentionné dans le cahier des charges.
Taux de conversion	15Mbps	Nous avons besoin de 4,8Kbps si nous prenons 300 échantillons par seconde à 16bits chacun. Nous pouvons donc endormir le driver durant le laps de temps où il n'est pas utilisé et ainsi limiter sa consommation.

TABLE 13 – Choix du driver RS-422.

Le RS-422 utilise la table de vérité suivante afin de pouvoir transmettre ses valeurs.

TRANSMITTING				
<i>INPUTS</i>			<i>OUTPUTS</i>	
RE(low)	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	x	Z	Z
1	0	x	Z	Z

TABLE 14 – table : Table de vérité du RS-422

#### 4.2.12 Connexion SUB-D 15

Afin de relier le géophone à la carte, il serait nécessaire d'avoir un interface SUB-D(D-subminiature) 15 ou connecteur VGA afin de pouvoir récupérer les données. Ceci ne serait valable que pour le géophone de nouvelle génération, soit le Baoding Longet Equipments 4.5 Hz 3.

Ce connecteur a également été fourni par la HEI.

Les informations sur les broches peuvent être consultées dans l'annexe 10.

#### 4.2.13 Mosfet

Afin d'alimenter le module GSM de façon contrôlée, nous avons décidé de placer un transistor. Ce transistor est de type MOSFET, c'est-à-dire qu'il est commandé en tension. Le modèle choisi est le :

## INTERNATIONAL RECTIFIER - AUIRLL014N - MOSFET CANAL N 55V 2A SOT-223

Ce transistor a été choisi pour les raisons suivantes :

Choix du transistor		
<i>Désignation</i>	<i>Valeur</i>	<i>Raisons</i>
Type	Canal N	On y applique une tension positive, entre 0 et 3.3V dans notre cas.
Courant supporté	2A	Le module GSM consomme en permanence 300mA et peut pendant un court instant demander 2A.
VDs Max	55V	Nous avons une tension maximale de 5V qui est fournie par les batteries. Il faut que cette tension soit supérieure pour que rien ne passe dans le cas où le MOSFET est bloqué.
Tension de seuil	2V	Elle est inférieure à 3.3V et donc elle peut être commandée par la FPGA qui peut placer une tension de 0V ou de 3.3V.
Résistance Rds	0,14[mΩ]	Cette résistance faible permet de réduire sensiblement les pertes lorsque le transistor fonctionne (laisse passer le courant).
Prix		Le moins coûteux de ceux qui restaient.

TABLE 15 – Raisons du choix du transistor.

Comme c'est un MOSFET avec un canal N, nous devons monter le drain sur la source de tension, la source sur la charge et enfin la gate sur l'une des I/Os de la FPGA afin de pouvoir la commander.

### 4.2.14 Switchs

Les switches vont nous permettre de choisir l'un ou l'autre des types de géophone. En effet, nous n'avons qu'un seul convertisseur AD, mais nous avons deux types de géophone, nous avons un géophone avec un connecteur de type Coaxial et un autre connecteur de type SUB-D-15. Or il n'est pas possible de pouvoir brancher les deux géophones en même temps. Nous avons donc trouvé un moyen de pouvoir alternativement choisir l'un ou l'autre.

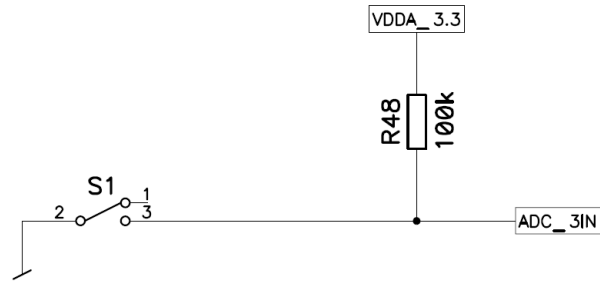


FIGURE 14 – Schéma du switch qui permet de sélectionner le géophone désiré.

Lorsque le switch est en "haut", la ligne est tirée vers le niveau logique haut et donc ceci nous permet de savoir que nous utilisons le géophone avec la connectique coaxiale. Par contre, lorsque le switch est dans l'autre position, la ligne est tirée à "0" et nous savons donc que nous avons trois entrées. Une résistance de 100k a été choisie afin de diminuer le courant qui circule.

Nous avons un autre switch qui permet d'envoyer sur une des broches du convertisseur AD soit l'une ou l'autre des sorties du géophone.

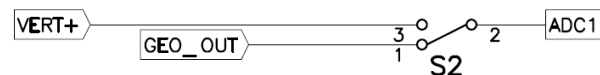


FIGURE 15 – Schéma du switch qui permet de sélectionner la sortie du géophone qui va être redirigée.

Ces deux switches vont nous permettre de sélectionner l'une ou l'autre des entrées de la carte, soit la connectique Coaxiale soit la SUB-D 15. La coaxiale est utilisée pour les anciens géophones, et la SUB-D 15 pour les nouveaux. Les switches doivent être positionnés de la façon suivante pour pouvoir obtenir les différentes entrées :

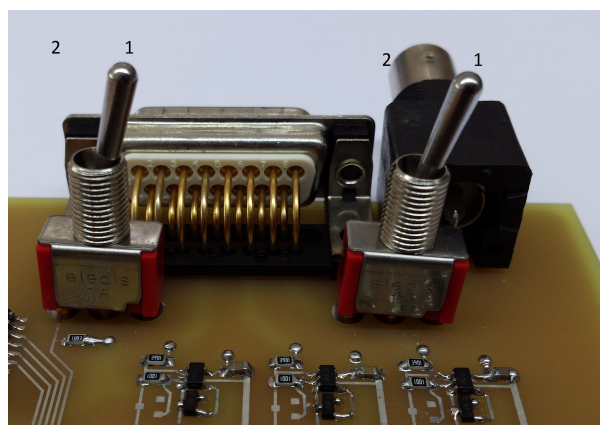


FIGURE 16 – Position des switches pour déterminer les fonctions que l'on désire.

Switchs		
<i>Position switch du bas</i>	<i>Position switch du haut</i>	<i>Fonction</i>
En bas(1)	En bas(1)	SUB-D 15
En haut(2)	En haut(2)	COAX

TABLE 16 – Choix de l'entrée

#### 4.2.15 Résistance de 0 Ohm

Nous avons placé plusieurs résistances de  $0[\Omega]$  dans notre schéma. Ceci permet de calculer le courant qui passe. Il suffit d'enlever cette résistance et de placer un ampèremètre entre les deux bornes.

#### 4.2.16 Divers petits matériels

D'autres petits matériels ont été ajoutés à la carte, comme par exemple des résistances, des condensateurs ou des inductances. Pour les filtres nous avons également dû prendre des amplificateurs opérationnels.

#### 4.2.17 Résumé du matériel à acheter

Voici un tableau résumant les différentes pièces ainsi que leur prix :

Matériel : Résumé		
<i>Composant</i>	<i>Revendeur</i>	<i>Prix</i>
FPGA Igloo	Farnell	24.35.-
EEPROM	Farnell	4.50.-
Mosfet FDFC2P100	Farnell	0.23.-
Driver RS-232	Farnell	2.15.-
Driver RS-422	Farnell	3.70.-
Convertisseur DC/DC	Farnell	8.80.-
Fusible 2AG 2.25A	Farnell	0.29.-
FuseBlock 5*15mm	Farnell	2.20.-
Inductance 820nH	Farnell	0.40.-
Amp OP Micropwr 5SOT23	Farnell	23.10.-
ADC AIC-111RHB	Mouser	12.54.-
ADC PCm3794	Mouser	10.67.-
Reset Max811	Mouser	2.94.-
Capacité 100nF	Farnell	0.78.-
Inducteur SMD 10uH	Farnell	2.20.-
Connecteur Micro-usb	Mouser	1.50.-
Prix total		100.35 $\approx$ <b>101.-</b>

TABLE 17 – Résumé des matériels à acheter

#### 4.2.18 Carte électronique

Voici une photo de la carte électronique assemblée.

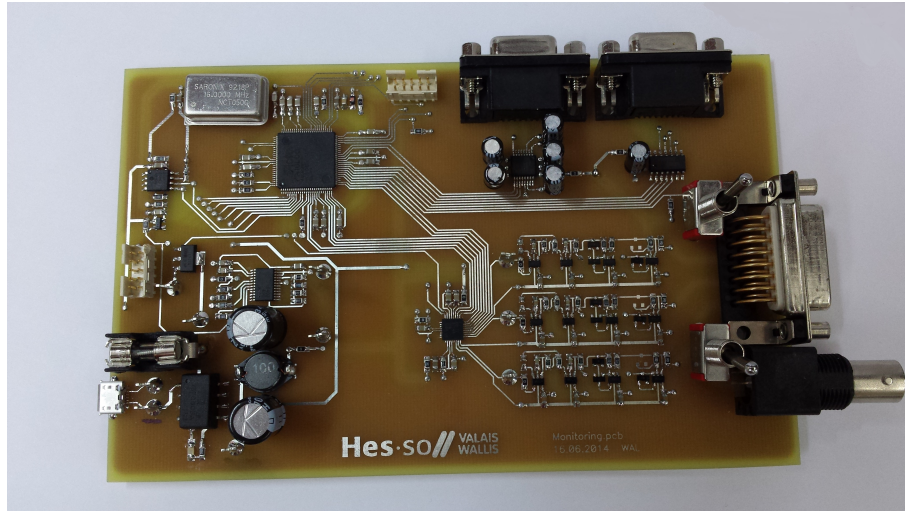


FIGURE 17 – Carte électronique du projet.

#### 4.3 Filtre digital-analogique

A la sortie du convertisseur DC-DC, nous avons une tension digitale. Nous allons appliquer un filtre passe-bas de type *CLC* afin d'obtenir une tension analogique en sortie. Afin de pouvoir lisser au maximum cette tension, nous plaçons de très grandes capacités. Ainsi, plus la fréquence est élevée et plus la résistance en direction du GND sera faible. De la même manière, l'inductance est prise relativement élevée afin de bloquer rapidement les hautes fréquences et donc d'avoir un signal des plus lisse possible.

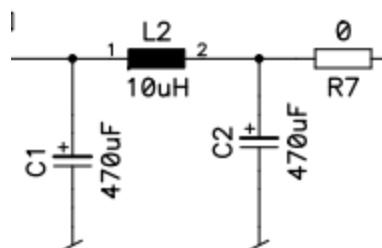


FIGURE 18 – Filtre passe-bas de type CLC

#### 4.4 Premier traitement du signal

Le géophone fournit des tensions très diverses. Or avant de pouvoir échantillonner le signal, il est nécessaire de le filtrer et de le redresser. Le schéma suivant montre les diffé-

rentes étapes qui sont effectuées.

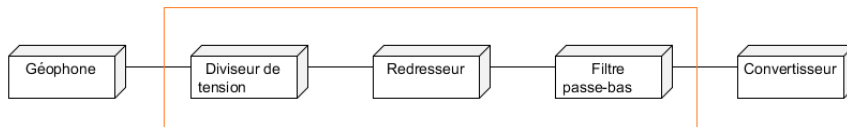


FIGURE 19 – Bloc du traitement du signal avant l'échantillonnage.

Nous allons parler des trois blocs qui se situent à l'intérieur du carré orange. Tout d'abord, le signal va subir une division. On réduit ainsi l'amplitude du signal. Les composants qui viennent par la suite ne peuvent supporter de grandes tensions, les tensions sont limitées à 3.3V. Le signal sera ensuite redressé, car le convertisseur ne peut pas recevoir en entrée des tensions négatives. Enfin, le signal sera filtré passe-bas. Le Théorème de Shannon dit que la fréquence d'échantillonnage doit être au minimum deux fois plus élevée que la fréquence maximale contenue dans le signal. Si nous ne faisons pas attention à ceci, nous pourrions avoir des mesures complètement faussées.

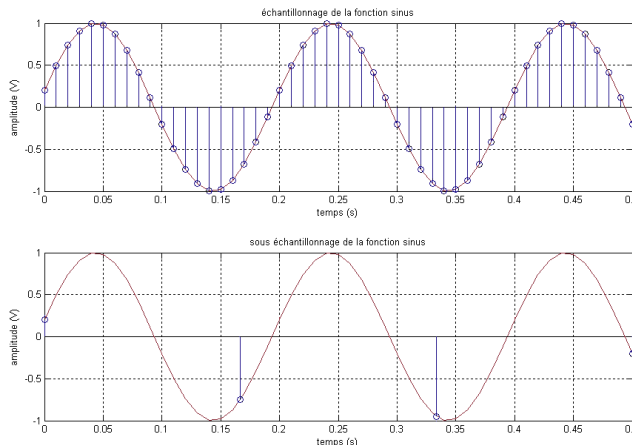


FIGURE 20 – Problématique si nous ne filtrons pas le signal passe-bas avant d'échantillonner. Source [15]

On observe sur cette figure que si la fréquence d'échantillonnage est trop faible par rapport à celle du signal, la forme du signal ne sera pas du tout respectée.

#### 4.5 Diviseur de tension

Tout à l'entrée nous avons un diviseur de tension, qui, comme son nom l'indique, divise la tension d'entrée. Le géophone peut émettre de grandes tensions, jusqu'à 20V mais les amplificateurs opérationnels qui se trouvent après ne peuvent pas recevoir 20V. C'est donc



pour cette raison que nous limitons la tension.

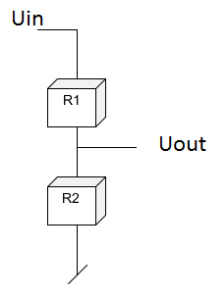


FIGURE 21 – Diviseur de tension.

La tension de sortie est proportionnel à la tension d'entrée et peut se calculer de la manière suivante :

$$U_{out} = \frac{U_{in} * R_2}{R_1 + R_2}$$

## 4.6 Redresseur

Avant d'arriver sur le filtre, nous avons besoin d'un redresseur. Le fait est que la tension générée par le géophone peut être soit positive soit négative. Ce qui nous intéresse c'est uniquement les amplitudes en valeur absolue ainsi que leurs fréquences. Le redresseur est obligatoire car le convertisseur AD ne supporte pas les tensions négatives en entrée. Sa plage de tension d'entrée varie de -0.3V à 4V.

Deux options sont présentées par la suite pour résoudre ce problème.

### 4.6.1 Redresseur à partir d'additionneurs et d'inverseurs

La premier consiste à placer un additionneur et un inverseur avant le filtre. Ceci permet donc de pouvoir faire un offset. Or pour que la plage d'entrée, qui est de -15V à +15V, fonctionne pour des tensions de -0.3V à 4V, il faut également diviser cette tension. Les contraintes étaient que la tension de -15V devait passer au-dessus des -0.3V et que la tension des +15V devait passer au-dessous des 4V, tout ceci avec un offset.

Le schéma équivalent est le suivant :

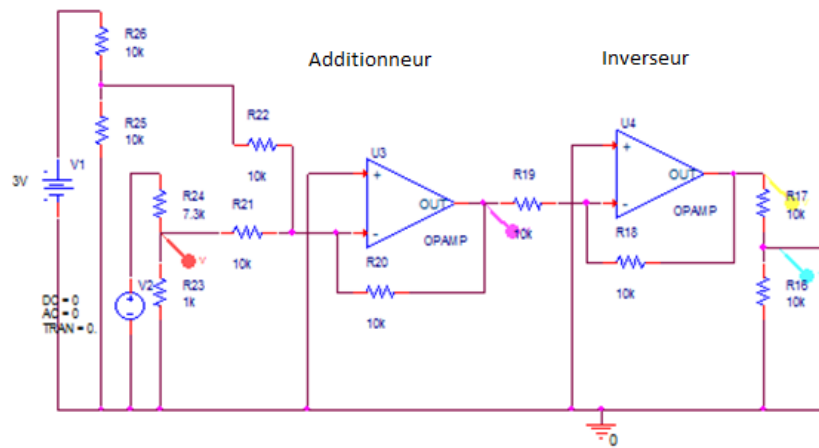


FIGURE 22 – Additionneur et inverseur

La réponse en tension de ce montage est représenté à la figure suivante :

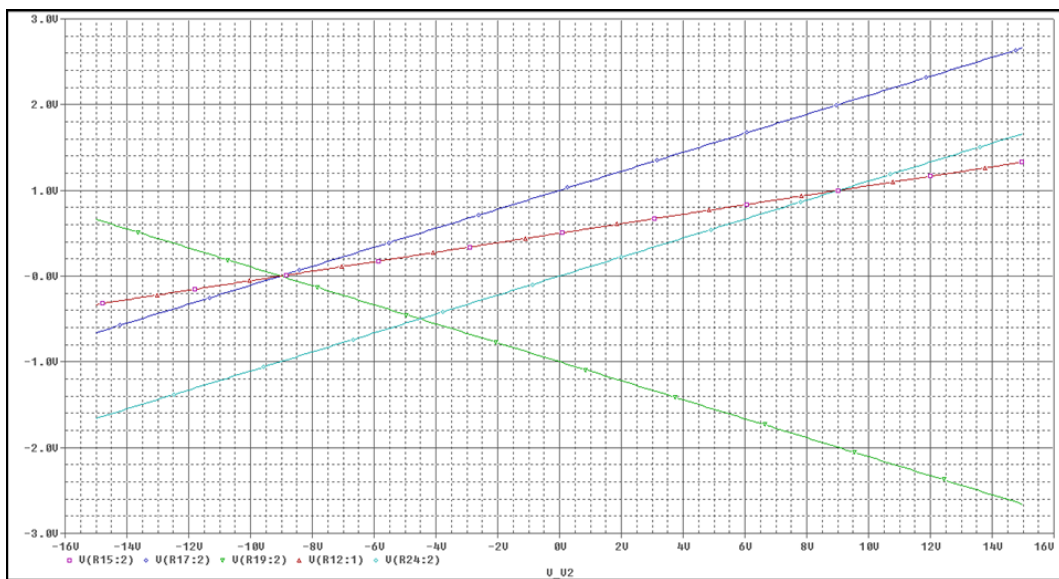


FIGURE 23 – Réponse de l'additionneur et de l'inverseur

Nous pouvons remarquer que la sortie (courbe rouge) se situe entre -0.3V et 1.4V. Ceci n'est pas idéal car nous avons comprimé le signal d'entrée, signal qui a un  $\Delta 30V$  à un signal qui a un  $\Delta 1.7V$ . Nous pouvons encore travailler un peu mais nous n'obtiendrons pas de résultats sensiblement meilleurs.

#### 4.6.2 Redresseur à partir d'amplificateurs opérationnels et de diodes

La deuxième option consiste en un redresseur à double alternances avec des amplificateurs opérationnels. Le schéma est le suivant :

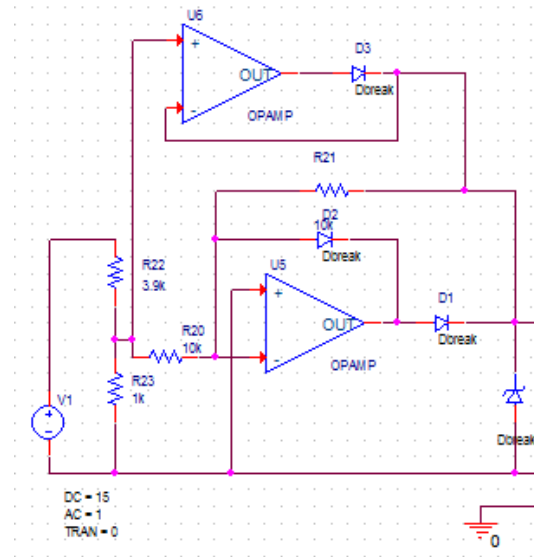


FIGURE 24 – Schéma du redresseur double alternance

Ce schéma est composé de plusieurs parties :

Nous avons tout d'abord un diviseur de tension à l'entrée. Celui-ci est couplé avec une diode Zener afin de supprimer tout ce qui dépasse les 3.3V. En effet, les amplificateurs opérationnels sont branchés à du 3.3V. Ils ne peuvent donc pas tirer la tension plus haut. Nous divisons la tension d'entrée par 5. Nous avons au maximum 15V en entrée et 3.3V en sortie. Donc  $U_{out} = \frac{U_{in} * R_2}{R_1 + R_2}$ .

Après cela, nous avons le redresseur à double alternance qui est essentiellement composé d'amplificateurs opérationnels et de diodes. Enfin, nous avons le filtre passe-bas, qui est décrit dans le chapitre "Filtre passe-bas" à la page 38 .

La réponse de ce filtre est illustrée par la figure suivante :

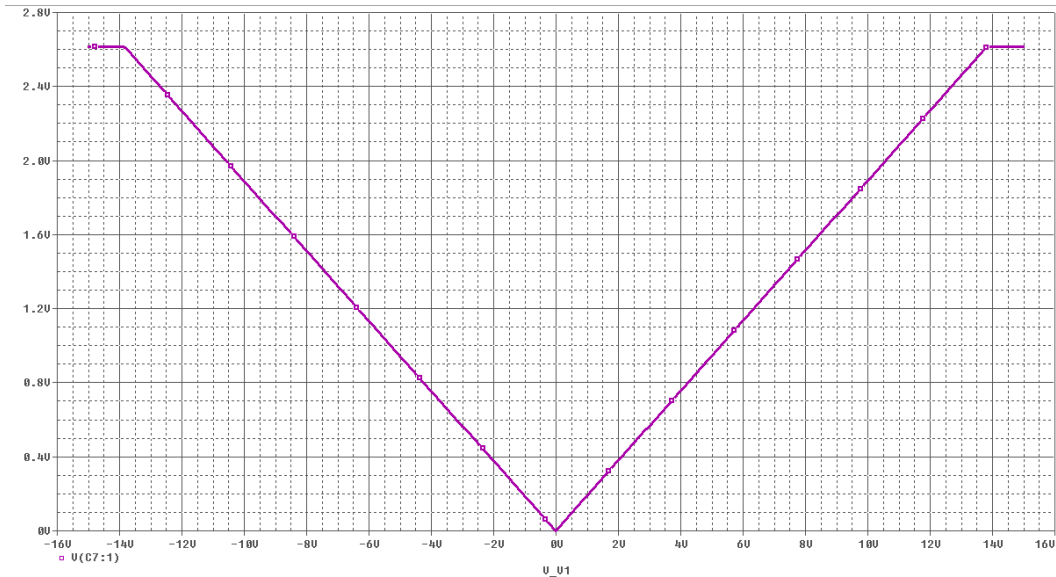


FIGURE 25 – Réponse du redresseur double alternance

Sur cette figure nous avons selon l'axe des X (en V), la tension d'entrée, positive et négative. Sur l'axe des Y la tension de sortie du redresseur (en V). Nous pouvons observer, que nous avons bien un signal redressé qui est identique que la tension soit négative ou positive. Le seul bémol qu'il peut y avoir arrive à cause des diodes ( qui ont une chute de tension de 0.7V) et malgré le fait qu'il y ait un amplificateur suiveur ( $U_{out} = U_{in}$ ), nous pouvons avoir une chute de tension. Nous passerons donc à  $3,3V - 0,7V = 2,6V$ .

#### 4.6.3 Choix du redresseur

Nous avons choisis le redresseur à double alternance, car il redressait la tension et le signal n'est pas trop perturbé. Un point négatif pour le redresseur avec un additionneur et un inverseur, c'est que la tension de sortie est très fortement réduite et que la plage de tension est limitée à 2Volts et nous avons encore des tensions négatives.

### 4.7 Filtre passe-bas analogique

A l'entrée du convertisseur AD il faut placer un filtre passe-bas. Lorsque l'on échantillonne un signal, il est impératif de filtrer ce signal avant d'échantillonner. Si cela n'est pas le cas, nous pouvons prendre des valeurs de très petites oscillations qui ne nous intéressent pas.

Objectif du filtre : Le filtre doit couper le signal avant la moitié de la fréquence d'échantillonnage. Les tremblements de terre et autres variations au niveau du sol ont des fréquences qui ne dépassent pas les 35Hz. Nous avons donc pris comme fréquence de coupure 50Hz. Nous voulons également une atténuation d'environ 100dB à la moitié de la fréquence de coupure.

Le filtre doit être conçu de manière à réduire toutes les harmoniques de fréquence supérieure à la fréquence de Nyquist, qui correspond à la fréquence d'échantillonnage divisée par deux.

Donc l'atténuation[dB] à la fréquence de Nyquist  $\geq 20 \cdot \log(V_{\text{ref}} / \text{résolution})$ .

Dans notre cas la résolution est de  $2^{16}$ . Ce qui implique donc que l'atténuation[dB]  $\geq 6,02 \cdot \text{nombre de bits}$ . Or afin de garder un peu de marge, nous avons la relation suivante :

$$\text{Atténuation[dB]} \geq 6,02 \cdot n + 1,76 = 98,08 \text{dB.}$$

Sachant qu'avec un filtre de butterworth nous obtenons une atténuation de 20dB/décade pour un premier ordre et que nous avons une fréquence de coupure à 50Hz, nous avons deux décades jusqu'à la fréquence de Nyquist. Afin de pouvoir obtenir ces 100dB au minimum, il est indispensable de faire un filtre d'ordre 3. Ceci afin d'avoir une atténuation de 60dB/décade et donc en deux décades d'avoir 120dB.

#### 4.7.1 Filtre passe-bas passif

Un premier essai de filtre a été réalisé avec un filtre passif du troisième ordre.

Filtre passe-bas.	
<i>Désignation</i>	<i>Valeur</i>
Fréquence à 3dB	50Hz
Pulsation normalisée	1 rad/s
DF	$2 \cdot \text{fréque à 3dB} \cdot \pi \cdot 1 \text{ rad/s} = 100 \cdot \pi$

TABLE 18 – Caractéristiques du filtre passe-bas

En regardant les coefficients de Butterworth, nous constatons que nous avons trois fois le coefficient 1 et une fois le coefficient 0.

Les différentes valeurs de capacités et d'inductances sont les suivantes. Le calcul de ces valeurs se trouve en annexe 16.

Composants électronique du filtre passe-bas.	
<i>Désignation</i>	<i>Valeur</i>
Z0	400 [ $\Omega$ ]
C	7.95uF
L	1.272H

TABLE 19 – Composants électronique du filtre passe-bas.

#### 4.7.2 Filtre passe-bas actif

Pour les filtres actifs, nous prenons toujours les coefficients de Butterworth. Nous avons donc un filtre du premier ordre et un filtre du second ordre. Ceci afin de pouvoir avoir un filtre du troisième ordre.

Nous avons un signal qui est redressé, or ce signal a une fréquence double par rapport à la fréquence réel. Afin d'avoir toujours les bons signaux et de ne pas trop couper, nous devons également doubler la fréquence de coupure. Ainsi la fréquence de coupure n'est pas de 50Hz mais de 100Hz.

Les valeurs des différents éléments du filtre sont cités ci-dessous et les calculs peuvent être consultés dans l'annexe 17.

Composants électronique du filtre passe-bas.	
Désignation	Valeur
C	10[nf]
R	160[k $\Omega$ ]
C11	10[nF]
C13	47[nF]
R14	82[k $\Omega$ ]
R12	68[k $\Omega$ ]

TABLE 20 – Composants électronique du filtre passe-bas actif.

Voici le schéma électrique avec les différents composants.

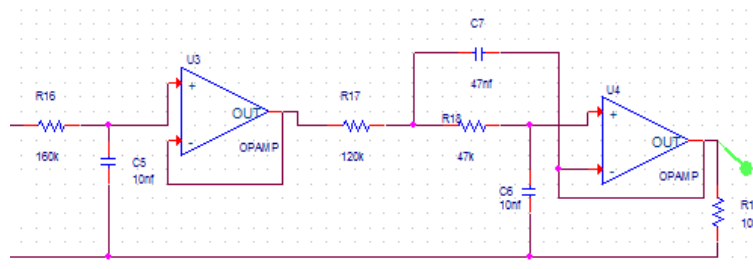


FIGURE 26 – Filtre passe-bas du 3ème ordre selon Sallen-Key

La figure suivante représente les tensions de sortie du filtre en fonction des tensions d'entrée. Les amplificateurs opérationnels choisis ne peuvent supporter en entrée que  $V_{CC}+0,2V$ . Soit 3,5V dans le cas présent.

Les réponses en fréquence du filtre peuvent être consultées dans l'annexe 21.

Afin de pouvoir réaliser ce filtre, l'amplificateur opérationnel suivant a été sélectionné : **TSV621ILT**. Il a les caractéristiques suivantes :

Choix de l'amplificateur opérationnel	
Désignation	Valeur
Tension d'alimentation	1.5V à 5.5V. Ceci rentre dans la plage de tensions que nous possédons.
Bande passante	420kHz ce qui est amplement suffisant car nous avons des fréquences de l'ordre d'une dizaine de Hz.
Température de fonctionnement	-40 à +85 degrés. Spécifié par le cahier des charges.
Slew rate	0.14V/us. Il a un slew rate faible ce qui a pour conséquences d'avoir peu de retard la tension transmise.
Disponibilité	Disponible à l'atelier de la HEI.
R12	68[kΩ]

TABLE 21 – Composants électronique du filtre passe-bas actif.

Cet amplificateur opérationnel n'accepte en entrée que des tensions comprises entre  $-VCC-0.2V$  et  $VCC+0.2V$ . Sachant que le géophone peut fournir des tension de 0 à 20V, il a donc fallu le faire précéder par un diviseur de tension. Le diviseur de tension a été calculé de la façon suivante :

$$U_{out} = \frac{R2 * U1}{R1 + R2}$$

$$5V = \frac{470 * 20}{470 + R1}, R2 = 470[\Omega]. (Choisie).$$

Nous trouvons donc  $R1 = 1,41[k\Omega]$ , soit en prenant des résistance normalisée,  $1,5[k\Omega]$ . Nous multiplions donc la tension d'entrée par :  $\frac{470}{470 + 1500} = 0.2385$ . Ainsi  $U_{out} = 0.2385 * U_{in}$ .

Nous alimentons ces amplificateurs opérationnels avec du 3,3V, ainsi en sortie nous aurons du 3,3V et donc nous n'avons pas besoin de diviser encore la tension. De plus, nous plaçons un filtre passe-bas à la sortie du convertisseur DC/DC.

Il faut respecter le fait que nous devons être en dessous de 100dB à la moitié de la fréquence de Nyquist. Or selon le schéma suivant qui est tiré de matlab, nous obtenons bien un nombre inférieur à -100dB à la moitié de la fréquence d'échantillonnage.

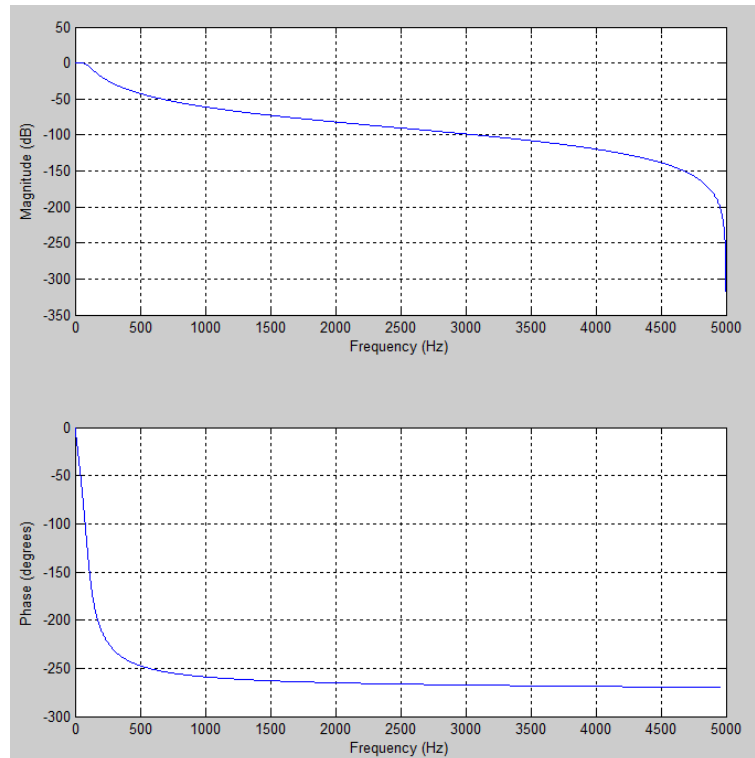


FIGURE 27 – Réponse en fréquence du filtre passe-bas de 3ème ordre avec une fréquence de coupure de 100Hz.

Nous observons qu'à partir de 3kHz, l'amplitude est inférieure à -100dB et donc que le filtre coupe bien ce que nous désirons.

#### 4.7.3 Choix du filtre passe-bas.

Nous avons décidé d'opter pour le filtre passe-bas actif car l'ordre de ce dernier (3ème ordre) est bien inférieur à l'ordre du filtre passif (plus de 500). Le choix est donc vite fait.

### 4.8 Configuration pour la lecture et l'écriture avec l'EEPROM

La lecture et l'écriture en mémoire se font grâce au protocole I2C. L'adresse de l'esclave : 100011 ADR R/W  
 L'ADR est choisi par l'utilisateur, ainsi que R/W qui indique ce que l'on souhaite faire.

L'esclave répond lorsqu'il reçoit son adresse.

#### 4.8.1 Lecture

Nous utilisons la lecture séquentielle. Le principe est d'avoir un pointeur sur la pile de données et lorsque nous voulons lire une donnée, ce dernier commence par la dernière donnée enregistrée et descend jusqu'à l'adresse "0x00".



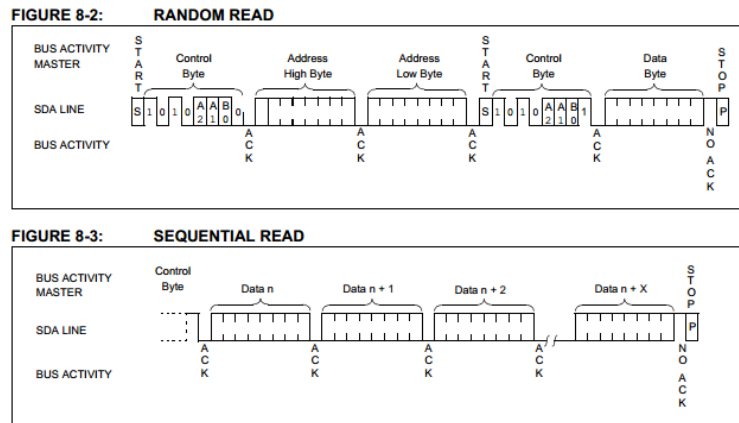


FIGURE 28 – Lecture dans l'EEPROM. (Source : Datasheet de l'EEPROM).

#### 4.8.2 Ecriture

Il est plus avantageux d'écrire par page, car nous avons un environnement favorable, c'est-à-dire beaucoup de données en même temps et donc le plus simple est de les écrire les unes derrières les autres. La figure suivante montre l'écriture par page.

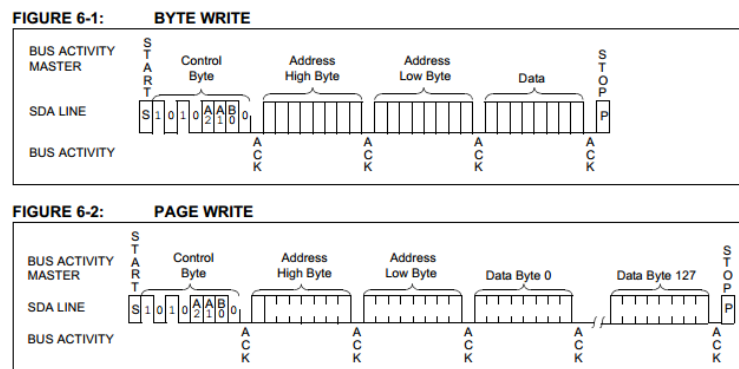


FIGURE 29 – Ecriture dans l'EEPROM. (Source : Datasheet de l'EEPROM).

### 4.9 Algorithmes

Le géophone fournit bon nombre de signaux. Or ces signaux ne sont pas tous intéressants pour ce projet. Nous devons réduire considérablement la quantité de données car la capacité mémoire est limitée.

Le but est donc de repérer les pics de tensions et de mémoriser leur valeur. Ensuite ils seront envoyés à une autre source.

Trois algorithmes ont été imaginés et sont présentés ci-dessous.

#### 4.9.1 Algorithme : Variation d'amplitude

Nous sommes partis de l'idée suivante : nous obtenons un signal redressé avec des amplitudes diverses. La phase positive ou négative du signal sinusoidale ne nous intéresse pas, seul l'amplitude de ce dernier ainsi que sa fréquence seront utilisées. La fréquence change mais nous nous concentrons sur le temps que met le signal pour aller de "0" au point maximum et de revenir. Ceci indique l'amplitude du signal et nous indique également si la secousse était plutôt longue ou courte.

Nous cherchons à obtenir seulement les amplitudes qui ont une certaine taille, nous voulons supprimer les composantes obtenues dans les hautes fréquences qui viennent perturber le signal. Nous savons que les fréquences sismiques se situent entre 1Hz et 30Hz. Donc tout ce qui se situe au-dessus n'a que peu d'intérêt. Nous ne voulons que certaines amplitudes. Nous avons donc cherché à ne garder que les valeurs maximales des signaux. De plus, ces signaux devraient avoir une condition supplémentaire : ne pas être trop court. Il fallait donc un algorithme qui :

- Trouve les maximums.
- ignore le signal s'il est trop rapide.

## 4.9.2 Schéma de l'algorithme

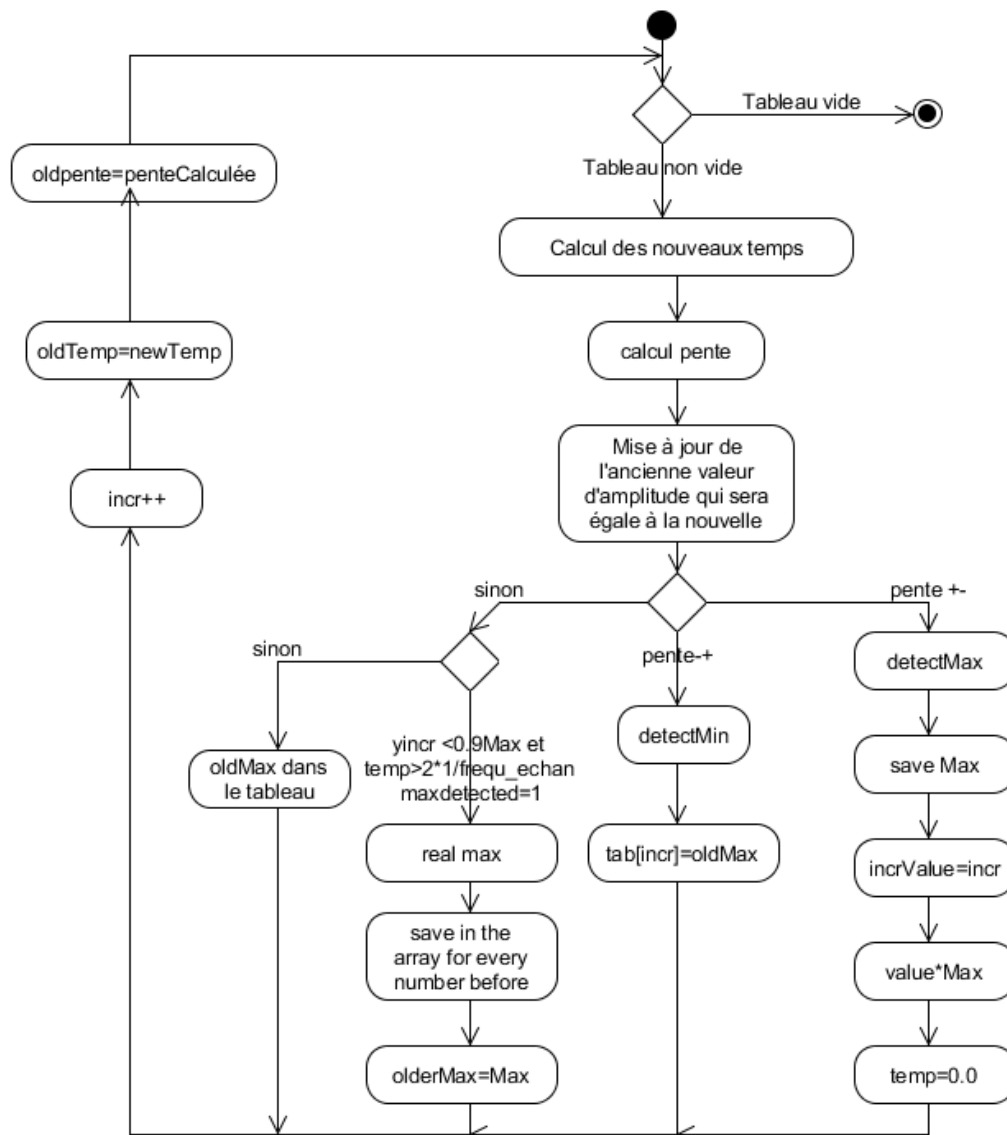


FIGURE 30 – Machine d'état de l'algorithme

La figure 30 montre l'algorithme sous forme d'une machine d'état. Nous allons expliquer cet algorithme.

Nous enregistrons les valeurs fournies par le convertisseur AD dans un petit tableau. Comme chaque échantillon est pris à une fréquence régulière, nous connaissons donc le temps entre deux échantillons.

Puis nous calculons la pente au point où nous sommes. La pente est calculée par la dérivée en ce point et la dérivée n'est autre que :

$$slope = \frac{AmplitudeNow - OldAmplitude}{TimeNow - OldTime}$$

Nous mettons ensuite à jour l'ancienne valeur de l'amplitude avec la nouvelle valeur. Ceci afin d'avoir la bonne valeur au cycle suivant.

Puis, nous cherchons à savoir si nous avons un maximum à un endroit donné. Un maximum se caractérise par une pente nulle. Or nous ne regardons que les changements de signe. Le fait de prendre une pente nulle était plus compliqué car une pente nulle pure n'existe pas, et si nous prenons la différence entre deux pentes successives, il est difficile de choisir un nombre correcte pour que l'algorithme fonctionne à chaque fois.

Afin de détecter un maximum "utilisable" nous allons regarder deux points. En premier lieu, nous cherchons un maximum, soit où les pentes sont successivement positives puis négatives. En second lieu, nous prenons le 80% de cette valeur maximale et nous regardons le temps que l'on fait entre le maximum et cette valeur. En fonction du résultat, nous savons si nous avons une secousse réelle ou si nous n'avons qu'un choc extrêmement rapide.

La figure suivante montre une représentation simplifiée d'une courbe.

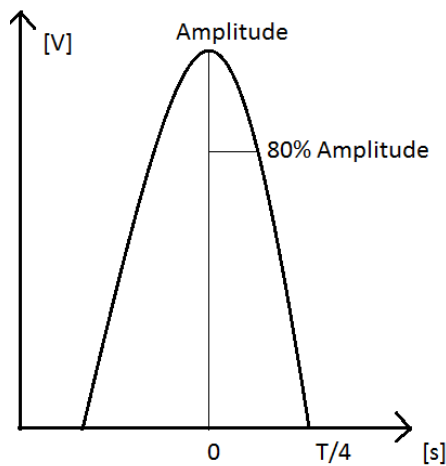


FIGURE 31 – Courbe simplifiée

Afin de trouver le temps entre le maximum et les 80% de ce dernier nous l'approximons avec une droite. Nous obtenons un triangle. Ce triangle se situe sur une période qui est égale à  $T/2$  car le signal est redressé, la fréquence est doublée.

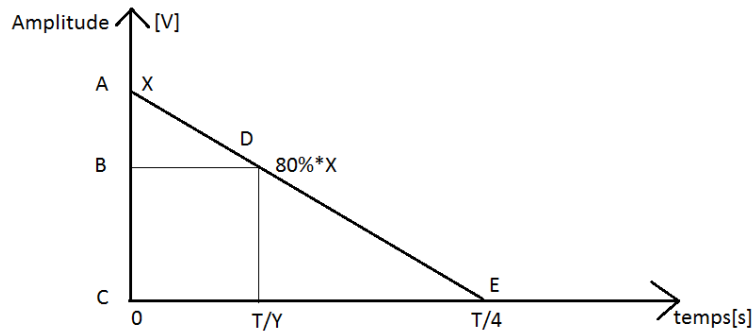


FIGURE 32 – Demi-triangle

Nous pouvons utiliser le théorème de Thalès qui dit :

$$\frac{AB}{AC} = \frac{BD}{CE}$$

Nous pouvons donc poser les équations suivantes :

$$\begin{aligned} \frac{AB}{AC} = \frac{BD}{CE} &\Rightarrow \frac{x - 0.8x}{x} = \frac{T/Y}{T/4} \Leftrightarrow \\ \frac{x - 0.8x}{4x} = \frac{1}{Y} &\Leftrightarrow \frac{Y}{1} = \frac{4x}{x - 0.8x} \Leftrightarrow \\ \frac{Y}{1} = \frac{4}{0.2} &\Leftrightarrow \frac{Y}{1} = \frac{20}{1} \end{aligned}$$

Ceci nous montre que nous avons le 80% de la valeur à environ  $T/20$ . Dans le cas où la fréquence est de 100Hz, la période est de 0.01 seconde. Nous avons donc  $\frac{T}{Y} = \frac{0.01}{20} = 500\mu s$ . Ceci arrive dans le cas où nous avons une droite. Malheureusement dans la réalité, ce n'est pas le cas. Pour avoir un facteur de sécurité nous diviserons ce temps par deux, soit 250 $\mu s$ . Ce facteur deux est également essentiel dans le cas où nous avons une courbe qui a la propriété d'être à basse fréquence et qu'une autre impulsion vienne perturber la mesure comme sur la figure 33. Ainsi nous concluons que le temps minimum à attendre afin de trouver un vrai maximum à 80% est de 250 $\mu s$ . Si nous avons une fréquence d'échantillonnage de 10kHz, nous pouvons simplement attendre deux fois la période d'échantillonnage et regarder si la valeur obtenue à cet instant est inférieure à 80% de l'amplitude maximale et ceci sans que la pente ait changé de signe.

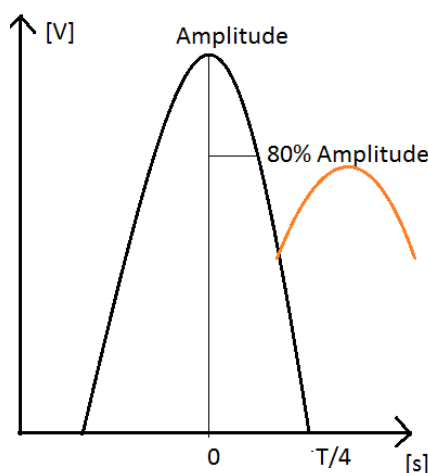


FIGURE 33 – Courbe perturbatrice en orange

### 4.9.3 Tests effectués avec l'algorithme

Nous avons représenté ici uniquement une partie du signal. En effet, le signal possède plusieurs milliers de points, environ 1'680'000. Comme le temps de simulation est proportionnel au nombre de points, nous avons rétréci le domaine afin de diminuer le temps d'attente pour le traitement du signal. Des captures d'écran du signal complet sont présentes plus loin dans le rapport. Il est également plus facile de lire les valeurs et de voir les changements sur une partie de signal que sur sa totalité.

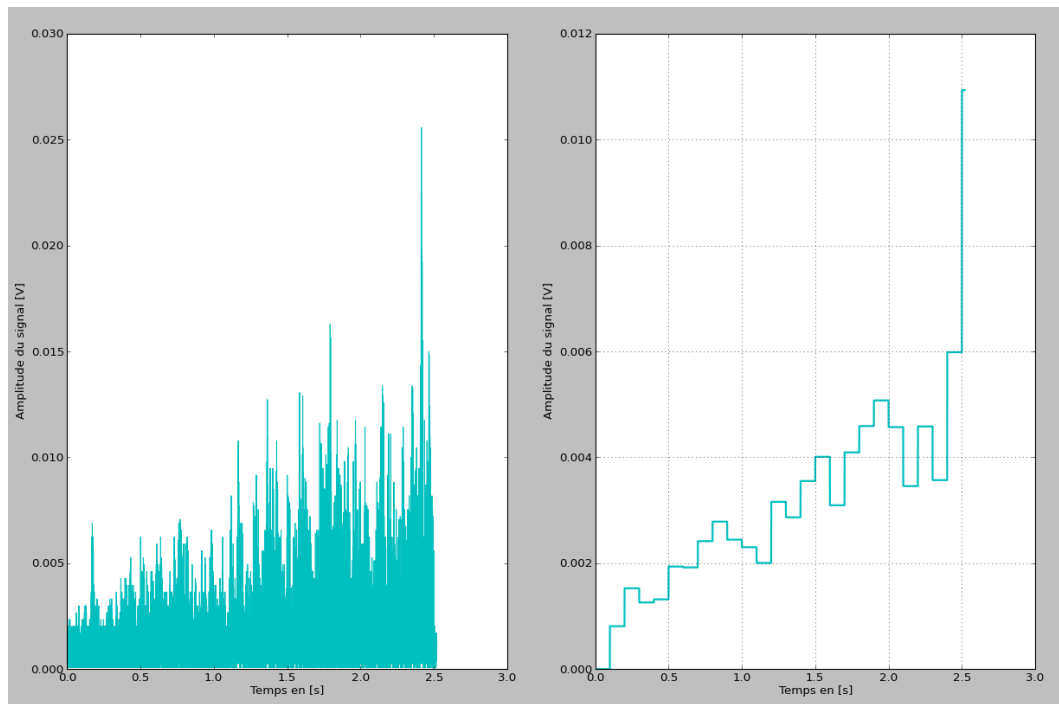


FIGURE 34 – Signal de départ et échantillonnage après filtrage

Nous comparons le signal de départ et le signal après traitement. Sur la gauche de la figure 34, nous avons le signal de départ qui n'est que redressé, et à droite, le signal ayant subi des filtrages, un algorithme et un sous-échantillonnage. Nous remarquons que l'allure générale de la courbe est respectée mais qu'il y a quand même un affaiblissement du signal. Lorsque l'on voudra détecter des amplitudes trop élevées et donc une secousse plus importante, il faudra choisir une valeur d'amplitude inférieure afin de pouvoir être cohérent. Ces valeurs de seuils devront être modifiées par la suite. (Les différentes parties et signaux intermédiaires peuvent être consultés en annexe 15.)

### 4.9.4 Algorithme : Moyenne des maximums

Nous avons mis au point un second algorithme. Il fonctionne de la manière suivante : nous cherchons les minimums. Dès que nous trouvons deux minimums sous certaines conditions, tous les maximums trouvés entre ces deux minimums sont sommés et une moyenne est prise. Puis tous les échantillons entre les deux minimums sont équivalents à la moyenne des maximums.

#### 4.9.5 Schéma du second algorithme

Voilà le schéma de la machine d'état qui correspond à l'algorithme.

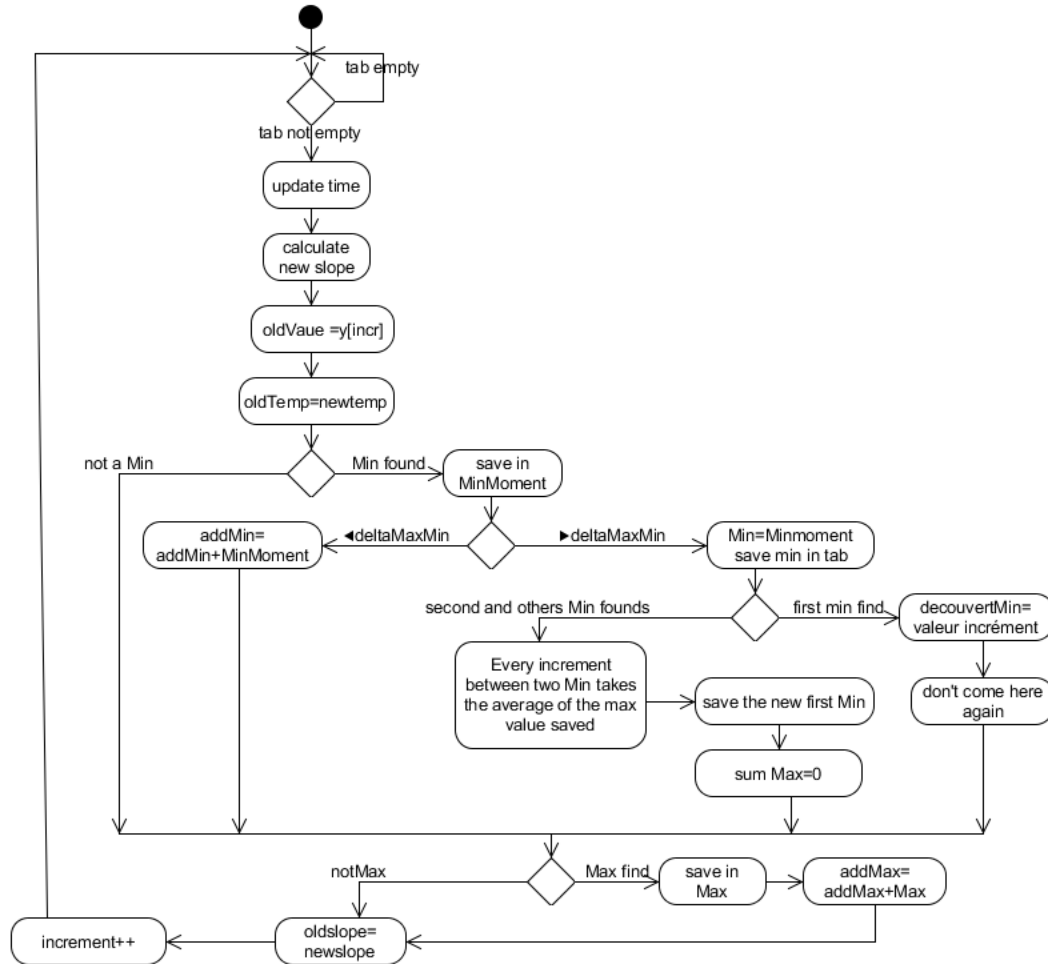


FIGURE 35 – Schéma du second algorithme

Ce deuxième algorithme fonctionne ainsi : on met à jour les diverses valeurs de temps et on calcule la nouvelle pente au point où nous nous trouvons. Ceci nous permettra de détecter un minimum ou un maximum. Dans le cas où nous avons un minimum, nous plaçons une référence et nous regardons si la différence d'amplitude entre le maximum qui le précède et le minimum est plus grand qu'une constante. Si cela est vrai, nous avons un minimum absolu.

Ce minimum va donc être mémorisé. Puis nous continuons à chercher un autre minimum qui a les mêmes propriétés. Et ainsi de suite.

#### 4.9.6 Tests effectués avec l'algorithme

Voici une partie d'un signal que nous avons traité avec cet algorithme.

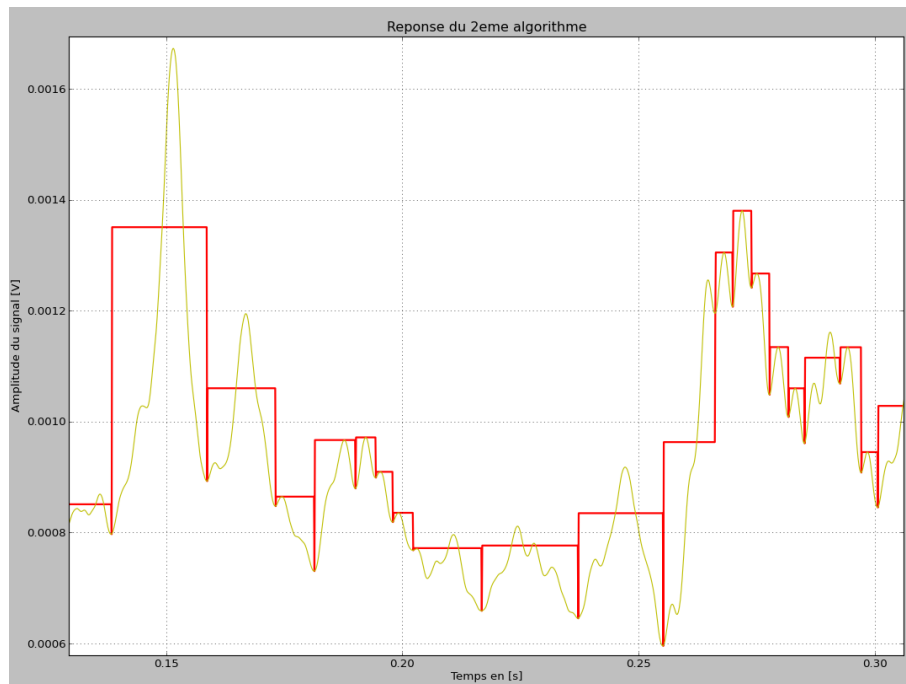


FIGURE 36 – Signal traité avec le second algorithme.

La figure 36 montre une partie du signal traité avec le second algorithme. En jaune nous avons la courbe initiale du signal qui a été filtré à l'entrée par un convertisseur AD (Ordre 3 et fréquence de coupure de 100Hz). Puis ce signal est passé dans l'algorithme. Le signal rouge en est la réponse. On voit que ce signal coupe le signal original car il fait une moyenne arithmétique des maximums locaux.

#### 4.9.7 Algorithme : Valeurs maximums

Ce troisième algorithme ressemble beaucoup au second à la différence que cette fois, au lieu de prendre la moyenne des maximums, nous prenons la valeur extrême. Ceci afin de pouvoir vraiment voir les pics des signaux et de ne pas perdre cette information-là.

#### 4.9.8 Schéma du troisième algorithme

Voici sous forme de machine d'état, l'algorithme numéro 3.



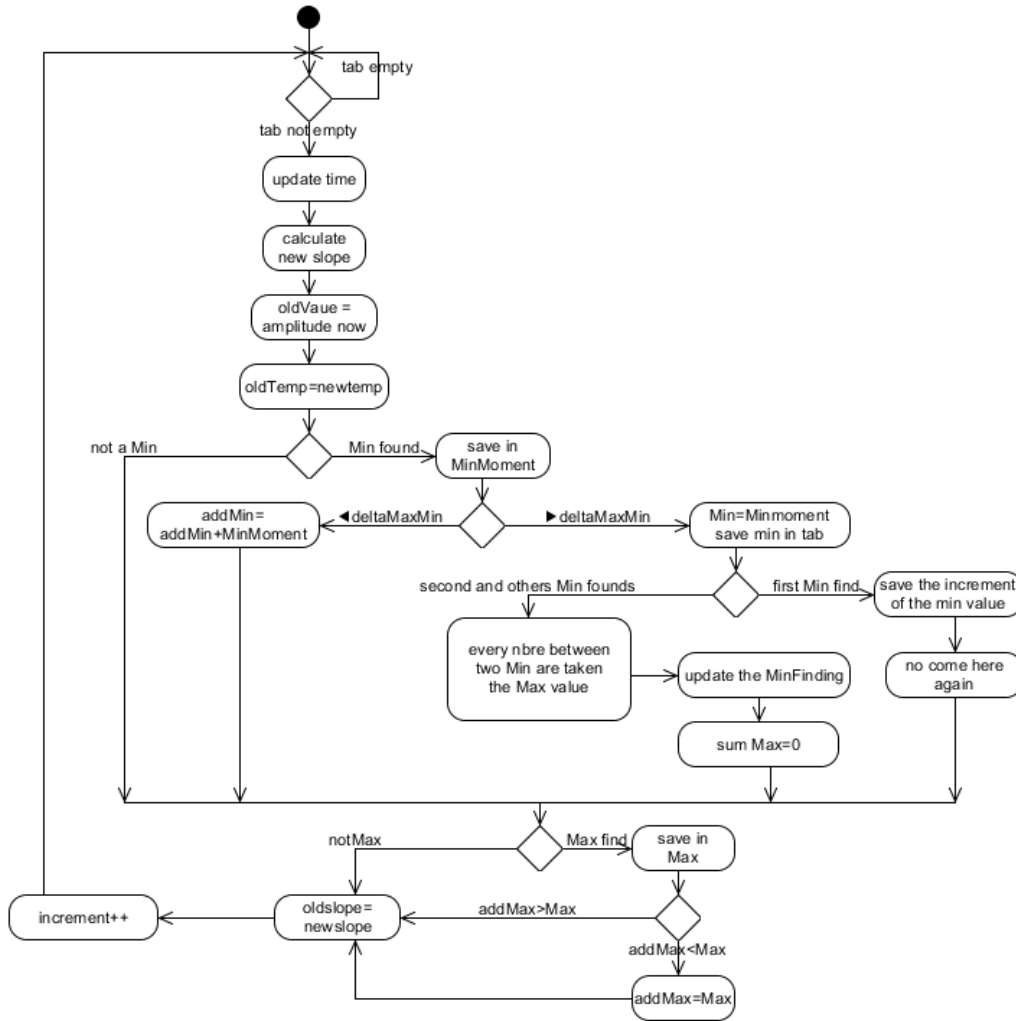


FIGURE 37 – Schéma du troisième algorithme

#### 4.9.9 Tests effectués avec l'algorithme

Voici une partie d'un signal que nous avons traité avec cet algorithme.

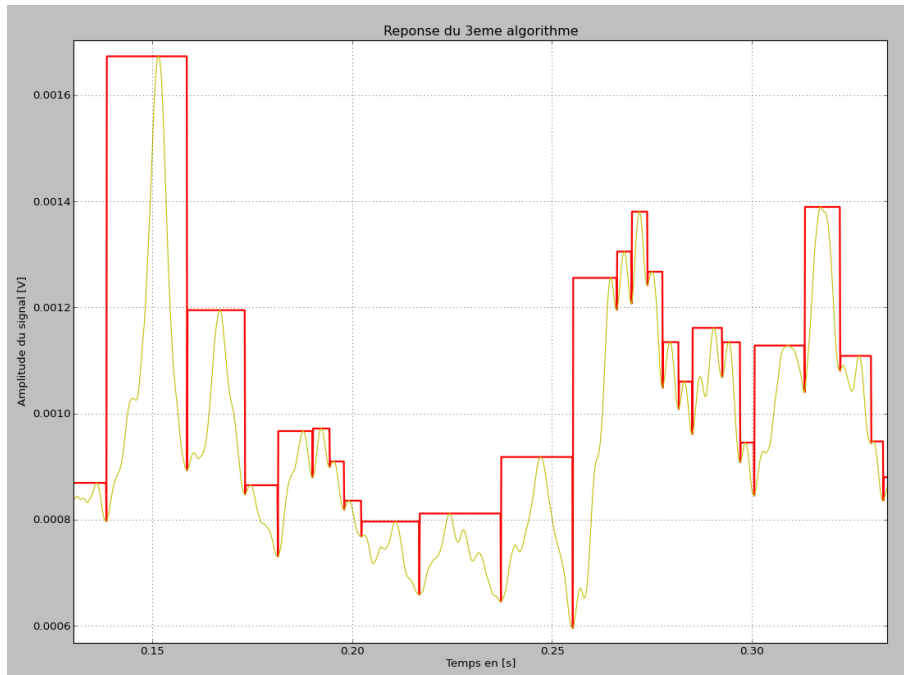


FIGURE 38 – Signal traité avec le troisième algorithme.

La figure 38 nous montre le signal brute en jaune (qui sort du convertisseur AD) et le signal rouge qui est le signal après avoir subi l'algorithme.

Nous avons également, refiltré passe-bas le signal rouge, puis appliqué à nouveau l'algorithme trois et enfin re-échantillonné.

Le moment de l'échantillonnage peut réduire certains pics à néant. Le meilleur moyen de garder les pics est de faire des rectangles. Or, en filtrant à nouveau, on perd beaucoup d'informations. La réponse est moins précise, ce qui est évident étant donné que nous réduisons encore l'intervalle des valeurs.

Ce qui est important dans cette phase, c'est la détection de pics et de savoir si ce dernier est très court ou non. Il est donc important de ne pas trop rétrécir les intervalles car dans ce cas nous ne pourrions plus discerner une secousse provenant d'un animal par exemple de celle qui est vraiment une secousse sismique.

#### 4.9.10 Comparaison des trois algorithmes

Nous obtenons finalement un signal qui a encore subi un filtrage avec une fréquence de coupure de 5Hz et un ordre 2 (soit -20dB/décade). Ainsi nous pouvons échantillonner à 10Hz.

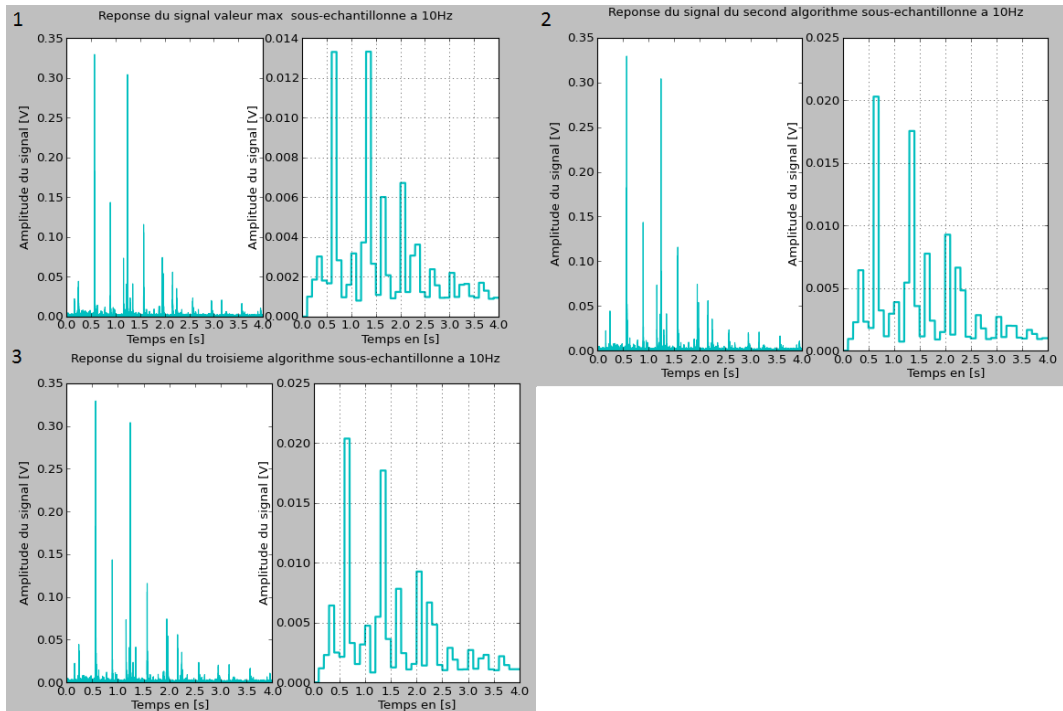


FIGURE 39 – Signal avec une fréquence de coupure à 5Hz.

Si nous abaissons la fréquence de coupure à 1Hz (figure 40) au lieu de 5Hz, nous obtenons quelque chose de beaucoup moins précis, tout est atténué donc les premiers grands pics sont bien visibles mais la suite n'est plus très ressemblante, ce qui n'est pas bon dans notre cas. Il faut donc rester avec une fréquence de coupure plus élevée.

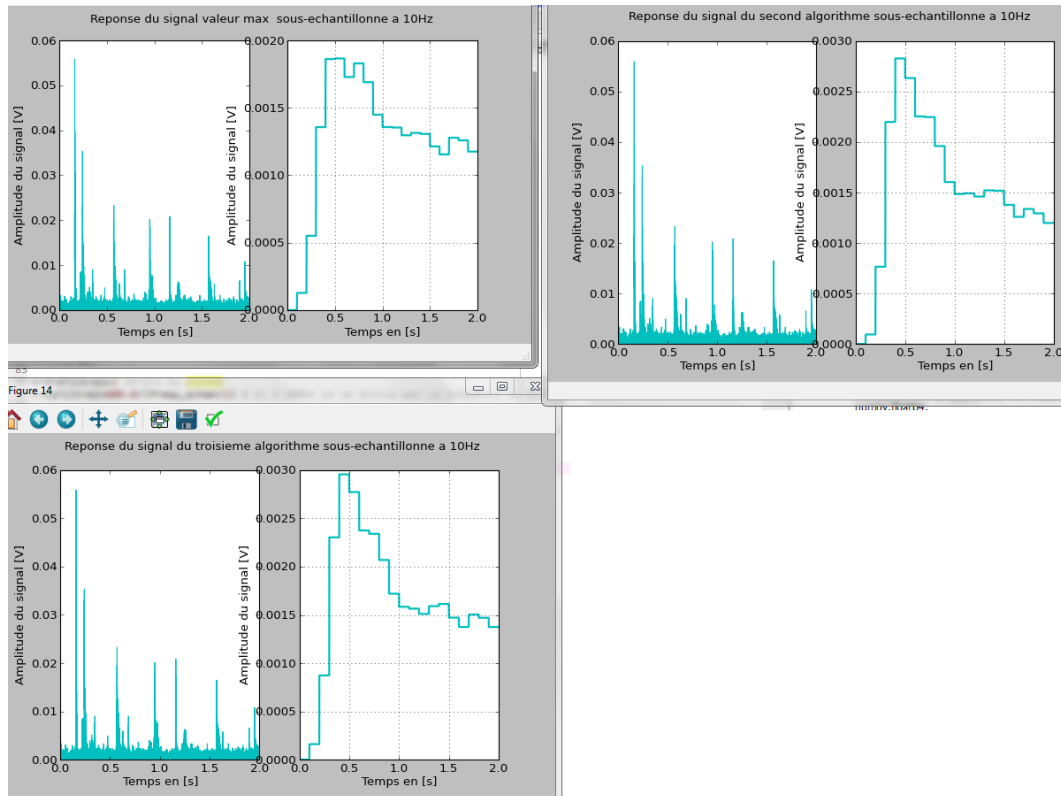


FIGURE 40 – Signal avec une fréquence de coupure à 1Hz.

Pour l'algorithme "Variation d'amplitude", qui correspond au numéro 1 sur l'image 39, on remarque que nous suivons l'allure générale de la courbe. Au départ, il y a quelques secousses avec une plus importante qui se fait ressentir et qui est marquée par un saut plus haut. Or, nous remarquons que le pic qui se situe entre les deux grands est sur le diagramme de gauche plus élevé que les suivants, ce qui n'est pas le cas sur le diagramme de droite qui montre qu'il a une intensité plus faible que ceux d'après. On remarque que ceci se répète pour les trois algorithmes.

Pour l'algorithme "Somme des valeurs maximums", on peut noter que nous sommes un peu plus proche du signal de base, les deux grands pics n'ont pas les mêmes amplitudes, ce qui se voit grâce à ce second test. On peut également relever que vers la fin de la simulation les amplitudes sont quasi identiques sur le signal de départ ainsi que sur le signal après traitement. Ce qui est moins le cas avec le premier algorithme.

Enfin l'algorithme "Valeurs maximums" a, lui, plus tendance à suivre les écarts des valeurs. Comme nous ne faisons pas de moyenne au niveau des maximums, il est clair que les écarts sont plus importants. Nous obtenons une atténuation du signal ce qui est normal étant donné que nous le filtrons, mais de façon générale, les amplitudes sont relativement bien respectées.

A la suite de ces remarques, nous pouvons tirer les conclusions suivantes : tout dépend en grande majorité du temps où est pris l'échantillon et ceci peut donc varier fortement. Mais de manière générale, l'algorithme qui a tendance à mieux représenter la courbe reste l'algorithme "Valeurs maximums". De plus, il montre mieux les pics que les autres algo-

rithmes. Certaines amplitudes sont mal exploitées. En effet, le fait de traiter le signal, puis de le faire à nouveau filtrer passe-bas, a tendance à "arrondir" les angles et donc à ne pas toujours suivre exactement la courbe.

#### 4.9.11 Algorithme : Détection de seuils

Un dernier traitement a été mis en place, c'est la détection de seuils. Le signal est comparé à divers seuils et le seuil. La valeurs du seuil qui lui est directement au dessous est incrémentée. Cela signifie que ce seuil a été dépassé. On commence bien entendu par le seuil le plus élevé. Dès que le nombre d'un des seuils a dépassé une certaine valeur, dans notre schéma 41 cela correspond à la valeur "MaxValue", le seuil le plus élevé qui a été atteint est envoyé. Nous envoyons le nombre de fois que le seuil le plus haut a été dépassé ainsi que le seuil en question. Ceci permet donc de n'avoir que 4 ou 5 valeurs et donc de réduire encore le nombre de données à enregistrer. Nous transmettons ces valeurs au plus tard chaque minute.

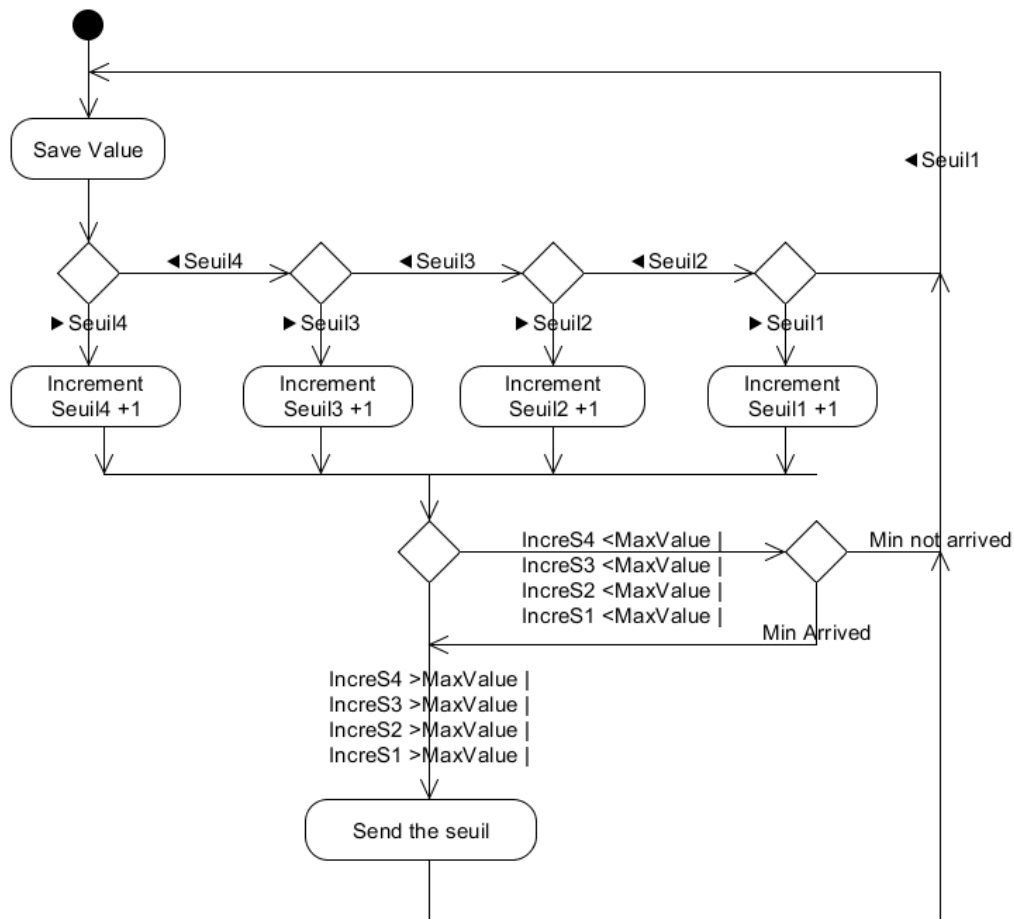


FIGURE 41 – Algorithme de détection des seuils.

C'est cet algorithme qui a été implémenté. Tout d'abord car il est simple à mettre en

placer et que pour débiter les tests il permet de visualiser seulement quelques valeurs.

## 4.10 Implémentation VHDL

Dans cette section nous allons parler des différents points effectués pour implémenter le programme qui se trouve dans la FPGA.

### 4.10.1 Les étapes

Nous recevons un signal filtré passe-bas à 100Hz à l'entrée du convertisseur. Puis celui-ci échantillonne ce signal et l'envoie à la FPGA. L'unité centrale va traiter ce dernier. Nous séparons les valeurs en fonction du géophone accouplé. Puis nous les comparons à différents seuils et incrémentons une valeur pour chacun de ces seuils. Enfin, chaque minute, nous envoyons ces valeurs via le port série.

A l'intérieur de la FPGA, nous trouvons plusieurs bloc avec pour chacun une fonction spécifique.

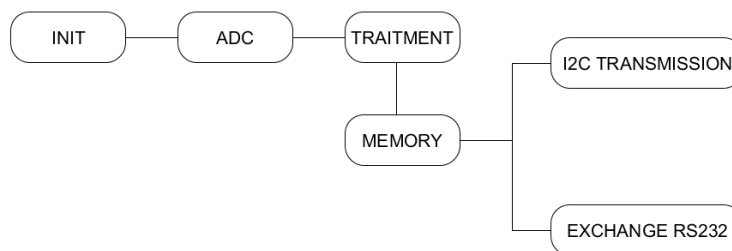


FIGURE 42 – Blocs de l'architecture à l'intérieur de la FPGA.

**ADC :** Ce bloc a pour but d'acquérir les données envoyées par le convertisseur AD. Nous générons également le BCK(Bit clock) et le LRCK(Left Right Channel Clock) à partir du clock afin de pouvoir piloter le convertisseur. Il possède également un sous-bloc qui nous indique si nous avons une entrée simple ou trois entrées différentielles.

**Initialisation :** Ce bloc configure le convertisseur. Il place dans ses registres des valeurs pour qu'on puisse le configurer comme nous le désirons. Lorsque la configuration est terminée, un signal est généré, permettant aux autres blocs de pouvoir commencer leur traitement.

**Traitement :** Dans cette partie, les données envoyées par le bloc ADC sont récupérées, elles sont ensuite traitées pour pouvoir limiter le nombre de données. Puis elle sont envoyées au bloc mémoire.

**Mémoire :** Ici, il est question de gestion de la mémoire. Nous enregistrons et lisons les données dans la mémoire. Les données lues sont ensuite envoyées au bloc Echange RS-232.

**Echange RS-232 :** Les données lues par la mémoire arrivent dans ce bloc et sont envoyées sur le port série.

**Transmission I2C :** Le dernier bloc, est le I2C Transmission. Il permet de transformer les données reçues afin qu'elles puissent être envoyées sur la ligne I2C. Les données qui voyagent sont les registres de configuration du convertisseur et les données lues et écrites dans la mémoire.

#### 4.10.2 Synchrone ou asynchrone

La différence entre les deux est la suivante : en mode synchrone tout est réglé par une horloge, le clock. Alors qu'en mode asynchrone, les entrées changent lorsqu'un nouvel élément arrive. Nous allons tout faire en synchrone ainsi tout sera réglé. Afin de lire les valeurs au bon moment, un "enable" doit être mis en place et être activé lorsque les données sont prêtes sur la ligne.

#### 4.10.3 Horloge

La base de temps est donnée par l'oscillateur qui est à 16MHz.

L'EEPROM a besoin d'une fréquence de 400kHz. Il y a donc un facteur 40 entre les deux. Car  $\frac{16MHz}{400kHz} = 40$ . Or il faut faire attention car ici nous ne comptons qu'au flanc montant, et donc nous devons diviser par 2 ces 40  $\Rightarrow$  20 afin de pouvoir avoir la bonne fréquence. Nous ne changeons d'état qu'au flanc montant de l'horloge et nous ne prenons pas en compte les flancs descendants, nous allons donc deux fois moins vite. Nous devons donc compter jusqu'à 20 et inverser la sortie.

Le convertisseur AD a besoin d'un clock à 100kHz. Donc  $\frac{16MHz}{100kHz*2} = 80$  (facteur 2 pour la même raison qu'au-dessus).

Nous avons trois canaux en série, nous devons donc échantillonner trois fois plus vite pour faire le tour des canaux.

Résumé des fréquences		
Fréquence	Protocole	Valeur
Principale, clock		16MHz
LRCK	I2S	31'250Hz
BCK	I2S	2MHz
Fréquence d'échantillonnage		10'416Hz
Horloge EEPROM	I2C	400kHz.
Horloge convertisseurAD	I2C	100kHz

TABLE 22 – Résumé des fréquences pour chacun des composants.

#### 4.10.4 Configuration pour les tests

Pour la configuration de la FPGA ainsi que des composants, nous avons utilisé un logiciel permettant de simuler qui est le HDL Designer. Nous avons, pour chaque bloc, testé le fonctionnement en simulant et en vérifiant les informations attendues.

#### 4.10.5 Protocole I2C

Le protocole I2C est un protocole de communication qui gère les règles de communication entre deux appareils compatibles avec ce modèle. Les échanges ont lieu entre un seul maître et plusieurs esclaves. La connexion est réalisée par l'intermédiaire de deux lignes :

- SDA : Serial Data Line
- SCL : Serial Clock Line

La HEI possède un bloc I2C, il est disponible en annexe 14.

**Simulation de l'I2C :** Nous avons donc implémenté cette fonction en code VHDL, puis nous avons fait une simulation afin de confirmer son bon fonctionnement. L'image 43 nous montre cette simulation. On peut voir dans la ligne TX, le 300 qui est synonyme de Start, puis divers caractères pour la trames et enfin le 3FF qui est synonyme de Stop.

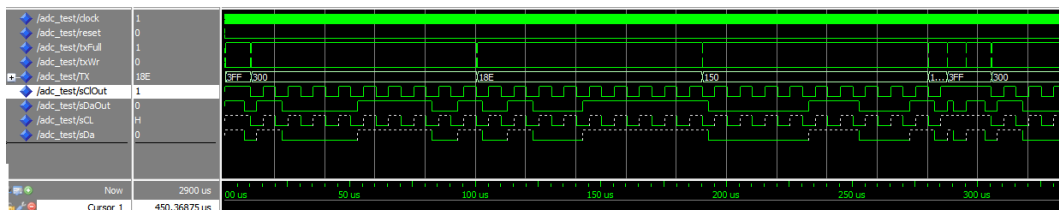


FIGURE 43 – Simulation I2C

Nous voyons sur cette figure des signaux, nous avons en hexadécimal les valeurs envoyées grâce à la ligne comportant le TX. La ligne sClOut est le clock du transfert. C'est elle qui donne le timing pour l'envoi et la réception des données. Le sDaOut comprend les données. Ce sont les valeurs contenues dans le TX qui sont traduites en binaire et envoyées.



Le signal sCl est équivalent au sClOut et le sDa au sDaOut. Les lignes pointillées signifie que la ligne est en haute impédance et donc que l'esclave ou le maître peut mettre un zéro, par exemple un acknowledge ou garder la ligne du clock à l'état haut pour que le composant puisse finir son traitement avant l'envoi de nouvelles données.

Les trames I2C ont été mesurées avec un oscilloscope. Nous avons une première image avec la simulation et la seconde avec la mesure. Les signaux sont identiques de même que le temps de la trame qui est de 2.58ms.

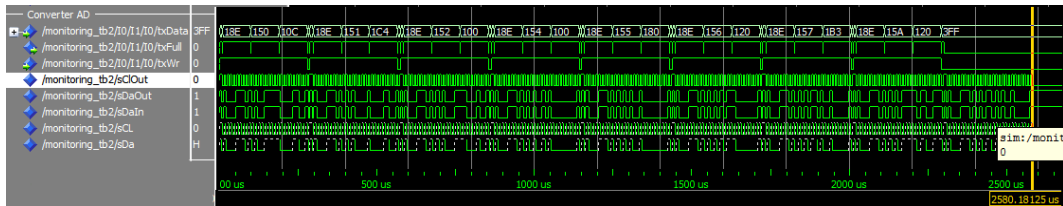


FIGURE 44 – Trame I2C simulée.

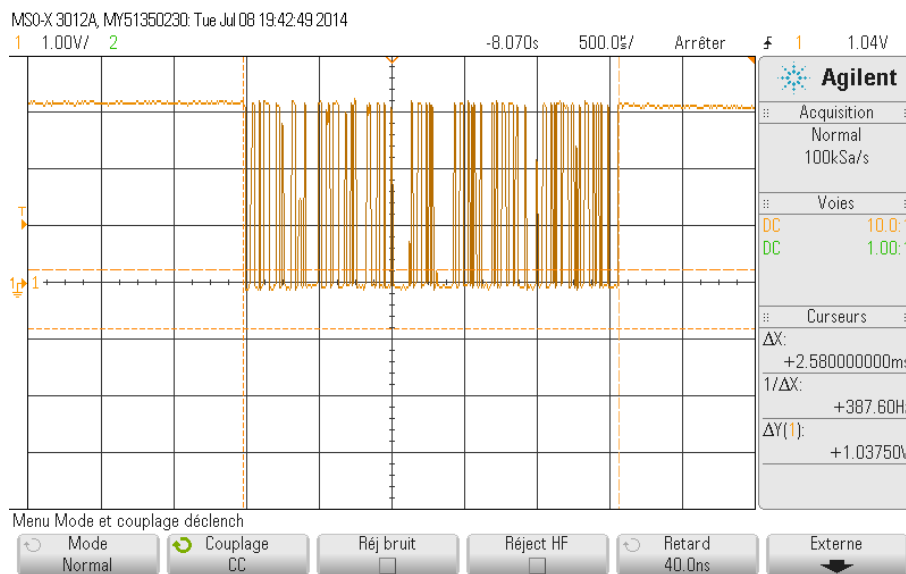


FIGURE 45 – Trame I2C mesurée par un oscilloscope.

#### 4.10.6 Protocole I2S

Le protocole I2S est un protocole de communication très utilisé dans le domaine de l'audio. Il est composé de trois lignes. Une ligne d'horloge bit, un signal d'horloge word et une ligne de données multiplexées.

L'horloge bit opère à une fréquence qui est un multiple de la fréquence d'échantillonnage. On transmet tout d'abord le MSB et on termine par le LSB, ceci permet de ne pas avoir besoin de se soucier de la précision audio de l'autre composant.

Nous avons mesuré à l'oscilloscope les différentes fréquences dont nous avons besoin. Pour le convertisseur AD, nous avons les fréquences BCK à 2MHz et la fréquence LRCK à 31250Hz.

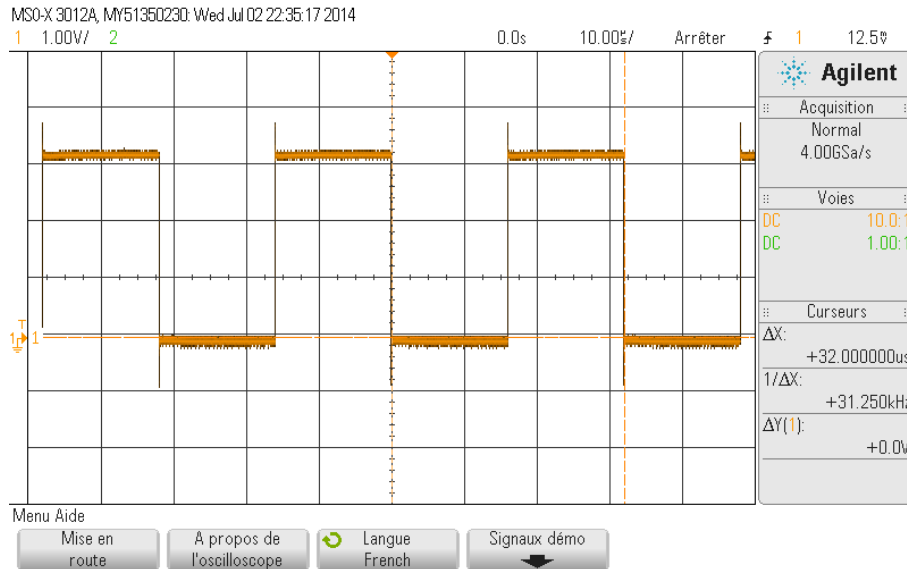


FIGURE 46 – Test LRCK : 31250kHz.

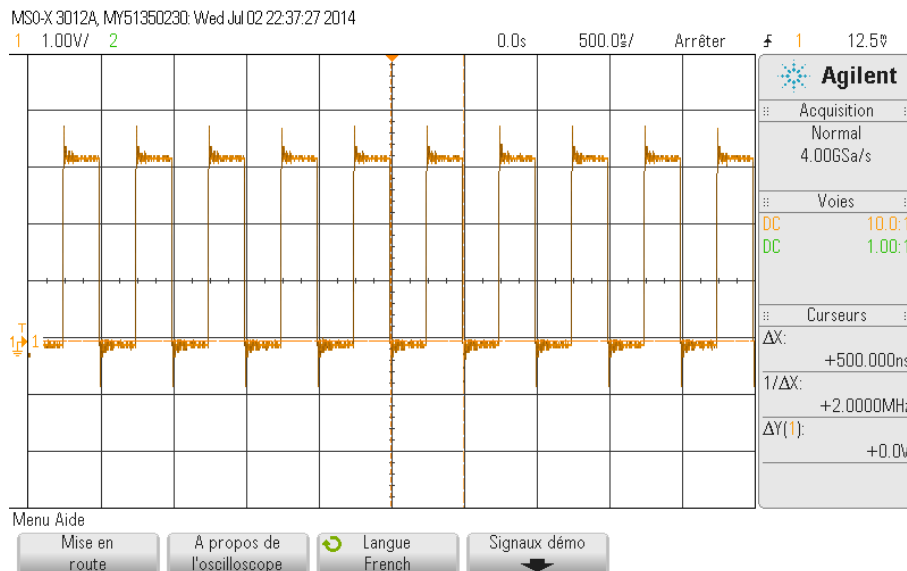


FIGURE 47 – Test BCK : 2MHz.

On constate que les fréquences attendues sont effectivement celles que nous pouvons lire sur l'oscilloscope.

#### 4.10.7 RS-232

Nous avons commencé par tester la connexion RS-232. Nous avons tout d'abord testé les lignes avec un oscilloscope, afin de regarder si leurs tensions sont correctes. La broche INVALID doit être à l'état haut. Dans le cas contraire, le driver RS-232 ne fonctionne pas. Puis nous avons connecté la prise à l'ordinateur et nous avons fait une boucle dans la FPGA afin de renvoyer les données reçues par l'ordinateur. A l'aide du logiciel "Tera Term", nous avons regardé si nous recevions les données que nous envoyons.

Nous devons donner le BaudrateDivide au bloc. cela indique la vitesse de transmission des données. Celui-ci est donné par :

$$\frac{\text{Frquenced'horloge}}{\text{Baudrate}}$$

Remarquons que la transmission avec le RS-232 se fait en commençant avec le bit de poids faible jusqu'au bit de poids fort et non pas l'inverse.

**Test de la communication RS-232 :** Nous avons ensuite simulé cette connexion et fait des tests grâce au logiciel TeraTerm. Ce logiciel permet de récupérer les données transmises par le port Série. La figure suivante montre la simulation et la figure le résultat obtenu avec le logiciel Tera Term.

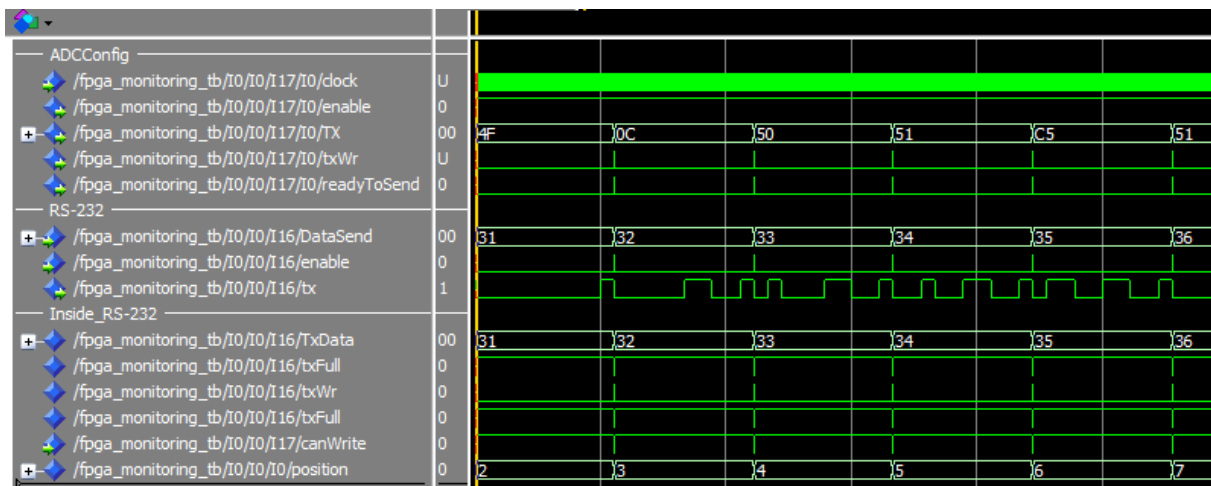


FIGURE 48 – Simulation de la communication RS-232

Sur la figure précédente, nous pouvons visualiser les signaux de données qui sont donnés par la ligne correspondant au TxData. Sur cette ligne nous observons les caractères hexadécimaux suivants : X31, X32, X33 qui correspondent dans le code ASCII aux valeurs 1,2,3 que nous retrouvons dans la figure ci-dessous. Les signaux txWr et txFull nous indiquent que nous voulons écrire et si la FIFO (First In First Out) du transmetteur est pleine.

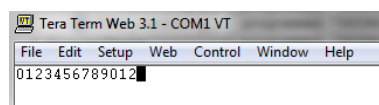


FIGURE 49 – Simulation de la communication RS-232

#### 4.10.8 Convertisseur AD

Afin de pouvoir récupérer les données qui viennent du convertisseur AD PCM3794, nous avons fait 2 blocs. Le bloc Init est composé de deux sous-blocs : Config\_ADC et Timer. Le bloc ADC quant à lui, est composé de deux blocs d'acquisition, ainsi que d'un bloc de choix. Il génère également le LRCK et le BCK qui correspondent à la fréquence d'échantillonnage et à  $64^*$  cette dernière. Puis regroupe les valeurs bit après bit pour ne former qu'un mot de 16bits.

Nous avons créé plusieurs blocs pour pouvoir simuler le convertisseur AD.

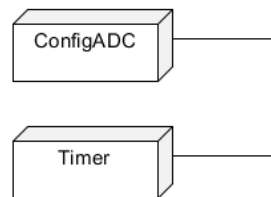


FIGURE 50 – Blocs pour la configuration du convertisseur AD.

**Config\_ADC :** Ce bloc contient les différents registres du convertisseur qu'il faut configurer. Pour se faire nous utilisons le protocole I2C. Nous avons choisis l'I2C car nous avons également l'EEPROM qui est compatible avec ce protocole. Les configurations des différents registres sont en Annexe 3.

**Timer :** Ce timer permet d'activer le reste du circuit dès que l'ensemble des registres du convertisseur ont été configurés.

Nous avons ensuite le second bloc, ADC. Ce dernier s'occupe de récupérer les signaux des switches, de fournir les fréquences pour les BCK et LRCK et d'acquérir les données.

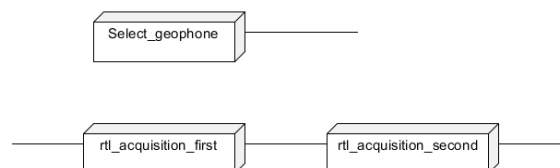


FIGURE 51 – Blocs pour l'acquisition des données.

**Select\_Geophone :** Ce bloc récupère la valeur des switches et permet de savoir si nous travaillons avec un géophone ne possédant qu'un canal ou plusieurs.

**rtl\_acquisition\_first** : Ici nous recevons les données du convertisseur AD. Les bits sont mémorisés dans une variable jusqu'à ce que les 16 bits soient arrivés. Ensuite ces données sont transmises au bloc suivant, qui est le rtl\_acquisition\_second.

**rtl\_acquisition\_second** : Ce bloc génère le BCK et le LRCK du I2S et le transmet au bloc précédent. Il réceptionne les données et les transmet plus loin. Cela permet au premier bloc d'être prêt à récupérer d'autres valeurs.

Voici la simulation obtenue qui montre la prise de valeur des différents canaux, gauches et droites. Or, nous savons que dans notre cas, seuls les canaux droits valent toujours '0'. Ceci est dû au fait que les pattes sont reliées à la masse et que nous ne prenons que les valeurs du canal gauche.

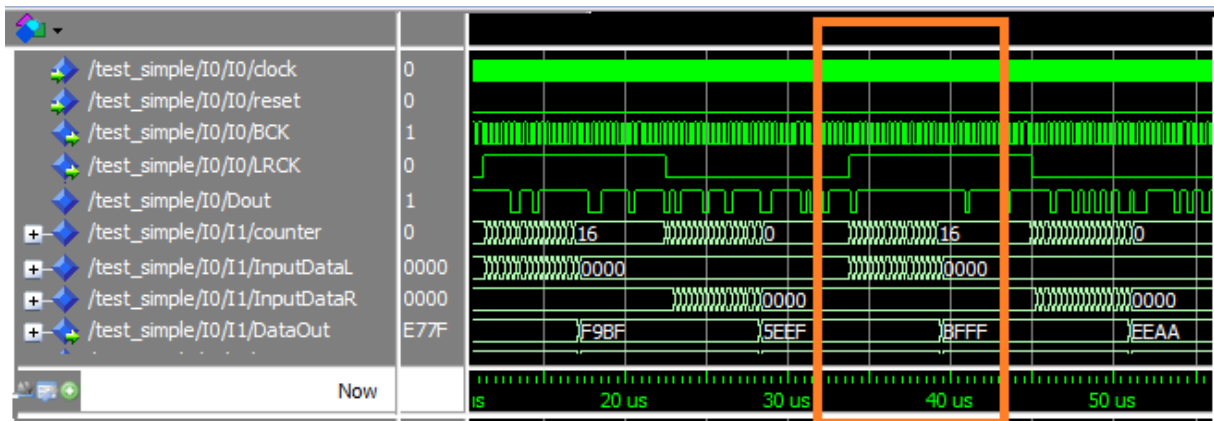


FIGURE 52 – Simulation de la communication avec le convertisseur AD. Valeurs de 16 bits récupérées.

Cette figure 52 nous montre plusieurs signaux. Ces signaux sont les suivants : Nous avons tout d'abord le clock et le Reset, qui sont au sommet. Puis nous avons le BCK que nous recevons du maître et de manière identique le LRCK. Pour rappel, le LRCK correspond à la fréquence d'échantillonnage et le BCK est 64 fois plus rapide. Le Dout est la sortie du testbench et elle nous permet de placer des valeurs afin de faire des tests. Ceci correspond à la ligne counter. Dès que les 16 bits sont atteints, la valeur du canal qui est entrain d'être enregistrée est copiée dans le signal DataOut et le counter est remis à zéro.

Nous pouvons voir dans le rectangle orange, que des valeurs arrivent sur la ligne InputDataL et qu'un compteur s'incrémente. Dès que ce compteur a atteint 16, il envoie la valeur au bloc suivant et remet à zéro son compteur ainsi que la sortie.

Pour enregistrer une valeur, nous faisons un décalage à gauche de la variable et nous y ajoutons la nouvelle valeur. Nous recevons soit un 0, soit un 1. Nous devons donc décaler cette valeur afin de pouvoir trouver la valeur transmise sur 16 bits.

#### 4.10.9 Algorithmes :

Afin de pouvoir limiter grandement le nombre de données transmises, nous avons implémenté l'algorithme de détection des seuils. Il a été séparé en plusieurs sous-blocs. Chaque bloc a une fonction bien précise.

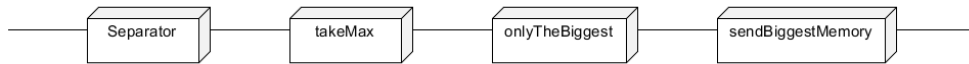


FIGURE 53 – Blocs pour le traitement des données.

**Separator :** Ce bloc sépare les données en trois canaux si le mode trois canaux est activé ou une valeur sur trois si le mode un canal (géophone sur la ligne coaxiale) est sélectionné.

**TakeMax :** A l'intérieur de cette partie, nous comparons les valeurs émises par le convertisseur avec des seuils. Nous incrémentons la valeur du compteur correspondant au seuil le plus grand qui est dépassé.

**OnlyTheBiggest :** Nous recevons les 4 seuils avec les valeurs incrémentées. Nous envoyons seulement le seuil le plus élevé qui a sa valeur incrémentée différente de 0.

**SendBiggestMemory :** Nous envoyons la valeur au RS-232 afin de pouvoir visualiser ces informations sur le port série. Ces informations sont également envoyées à la mémoire pour y être stockées. Pour le moment, les informations sont uniquement envoyées sur le port série car les priorités de lecture et d'écriture sur la mémoire ne sont pas encore totalement validées.

Nous avons ensuite simulé l'algorithme. Nous mesurons les amplitudes des signaux et incrémentons un compteur lors du dépassement du seuil correspondant.

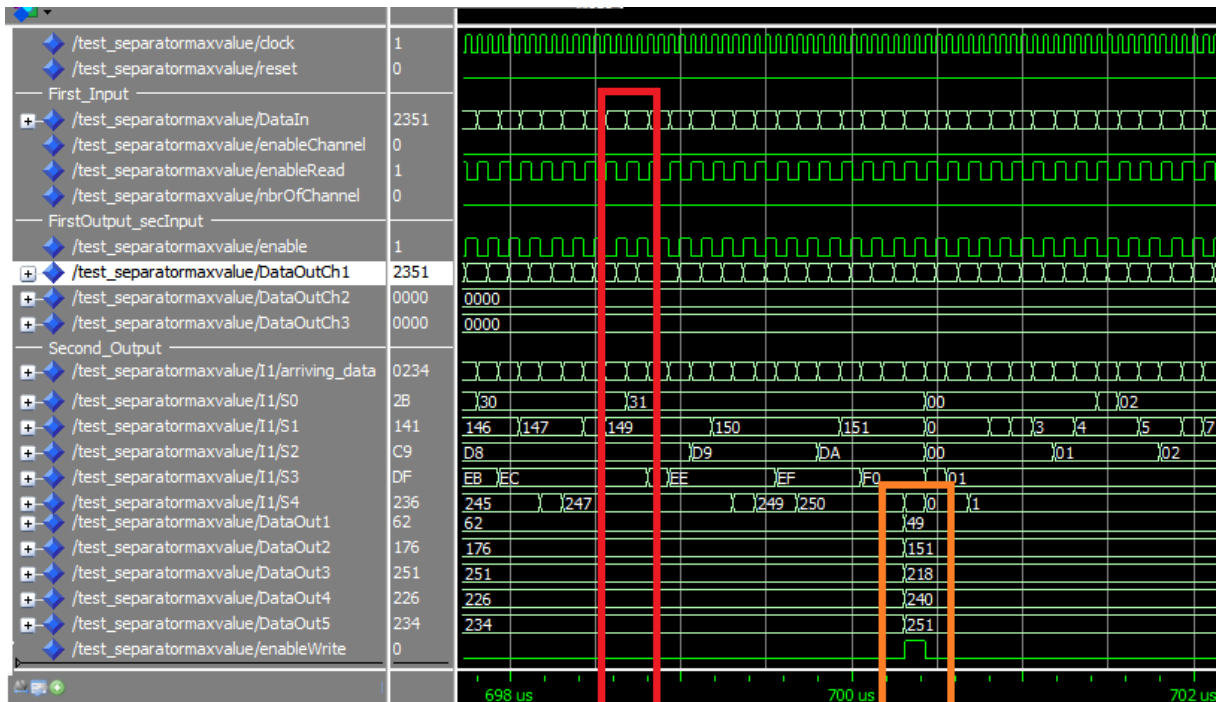


FIGURE 54 – Simulation de l’algorithme des seuils.

On observe les données qui arrivent par le signal à la ligne DataIn. Ces données sont ensuite dirigées vers les sorties. Dans notre cas, il n’y a qu’une seule sortie, c’est le signal DataOutCh1.

La donnée (dans le rectangle rouge) qui est sur le bus DataOutCh1 arrive dans le bloc suivant et est copiée dans le signal InputData. Ce signal est ensuite comparé à un des seuils et s’il est plus grand que la valeur d’un des seuils, il incrémente la variable du seuil. Dans notre cas, la variable qui est incrémentée est celle de S0. On remarque qu’il est passé de 30 à 31.

Nous pouvons voir dans le rectangle orange, que la valeur S4 est arrivée à 251. Dans ce cas, Les données de S0 à S4 sont mises en sortie. S0 et S4 sont remises à 0 et le processus recommence.

#### 4.10.10 Mémoire :

L’EEPROM est la mémoire de notre carte, elle nous permet de stocker les données et par la suite de pouvoir les envoyer par paquet. Cette dernière utilise le protocole I2C pour communiquer.

Voici l’architecture interne du bloc mémoire.

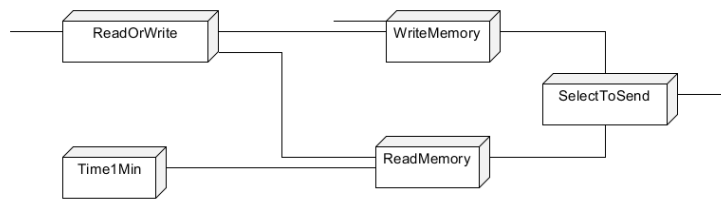


FIGURE 55 – Blocs internes au bloc mémoire.

**ReadOrWrite :** Ce bloc gère de façon très simple l'exclusion mutuelle. Dès qu'on lit, on ne peut écrire. Par contre si deux veulent lire au même moment, nous avons un problème. Une modification doit être apportée.

**WriteMemory :** Nous recevons ici les données traitées et les envoyons sur le bus I2C afin qu'elles soient stockées en mémoire.

**ReadMemory :** Ce bloc a pour fonction de lire les valeurs en mémoire et de les envoyer par la suite sur le port série, c'est à dire à un ordinateur ou un serveur.

**SelectToSend :** Ce bloc ne permet pas de lire une donnée en même temps que de l'écrire sur le bus I2C.

Ce bloc ne fonctionne pas encore totalement, il y a les problèmes de priorités à changer .

#### 4.11 Taux d'occupation dans la FPGA

Après avoir synthétisé le programme, nous pouvons trouver les différents taux d'occupation dans la FPGA.

Résumé du taux d'occupation			
<i>Composant</i>	<i>Utilisé</i>	<i>Total</i>	<i>Pourcentage utilisé</i>
CORE	2749	3072	89.5%
I/O	15	71	21%
Global Chip+Quadrant	6	18	33%
PLL	0	1	0%
RAM, FIFO	2	8	25%
Low Static ICC	0	1	0%
FlashROM	0	1	0%
User JTAG	0	1	%

TABLE 23 – Taux d'occupation de la FPGA



A ce stade nous n'avons pas optimisé les blocs, mais il est possible de réduire les données utilisées en améliorant le code.

#### 4.12 Codage des données

Afin d'envoyer peu de données, nous enverrons les informations suivantes :

- 1) : L'amplitude du signal.
- 2) : Le nombre d'éléments ayant cette même amplitude. Ainsi en connaissant la période d'échantillonnage, nous pouvons calculer ensuite le temps durant lequel les valeurs sont identiques. Pour ce faire :

$$\text{Temps} = \text{Nombre d'éléments} * \text{période d'échantillonnage} = \frac{\text{nombre\_elements}}{\text{frequence\_echantillonnage}}$$

#### 4.13 Tests effectués sur la carte

Nous commençons tout d'abord par contrôler que les pistes aillent bien aux endroits voulus à partir du schéma électrique.

On regarde également qu'il n'y ait pas de court-circuit. Pour ce faire, nous utilisons un multimètre et nous utilisons la fonction permettant d'émettre un son lorsque nous faisons un court-circuit.

Si rien de cela n'est détecté, nous pouvons brancher l'alimentation en commençant avec une tension relativement basse. Nous pouvons ainsi mesurer le courant aux différents endroits et regarder s'il y a une erreur à quelque part. Nous avons placé des résistances  $0[\Omega]$  à certains endroits afin de pouvoir mesurer le courant.

Nous avons donc pratiqué le smoke test, qui consiste à observer si quelque chose fume ou chauffe lors de la mise sous tension. Dans notre cas, rien ne s'est passé et tout à l'air de fonctionner.

Une première "erreur" trouvée est le fait qu'à la sortie du convertisseur DC-DC nous obtenons plus que 3.3V si nous mettons du 5V. Nous montons jusqu'à 3.7V. Ce qui pose problème. Afin d'avoir la bonne tension, nous pouvons faire un diviseur de tension avec le bon rapport. Afin d'obtenir 3.3V à la sortie du convertisseur DC-DC, nous devons mettre en entrée une tension qui vaut : 4.375V. En laissant cette tension et en mesurant aux pointes de tests, les tensions attendues sont bien là.

Un premier test de consommation nous indique que la consommation de courant s'élève à 30mA. En prenant en considération le fait que la batterie a une capacité de 14'000mAh, nous pouvons tenir 19jours. Or, ceci sans qu'aucun calcul n'ait été effectué et que tout soit "en fonction".

Nous regardons ensuite le convertisseur AD. Pour le moment, ce dernier ne fonctionne pas. Il ne nous indique rien sur la ligne Dout qui est la ligne de sortie des datas. Elle va du convertisseur à une patte d'entrée de la FPGA.

Plusieurs problèmes peuvent être la cause de ce dysfonctionnement. Les problèmes possibles sont énumérés ci-dessous :

Cause de dysfonctionnement du convertisseur AD.	
<i>Problème</i>	<i>Méthode de résolution</i>
Il n'est pas monté correctement sur la carte.	Nous avons repris le schéma du PCB et avons comparé la position du convertisseur sur le schéma et sur la carte. La broche 1 e situe au même endroit pour les deux cas. Nous pouvons donc écarter cette hypothèse.
Les soudures ne sont pas bien faites.	Nous avons regardé à nouveau les soudures au microscope et tout semble bon.
Les tensions d'alimentation ne sont pas bonnes.	A part VCom qui est à 1.35 au lieu des 1.65V, toutes les autres tensions sont correctes.
Les registres de configuration ne sont pas correctes.	Les registres de configuration ont été réexaminés mais sans succès.
Le multiplexeur sélectionné n'est pas le bon.	A vérifier
Les trames I2C ne sont pas correctes et donc les données reçues ne sont pas compréhensibles par le convertisseur.	A vérifier
Les fréquences SCKI, BCK et LRCK ne peuvent être choisie de façon aléatoire mais doivent être bien précises.	Dans ce cas il faudrait changer l'oscillateur, ou générer des fréquences grâce à une PLL (Phase-locked loop). Or BCK et LRCK peuvent être des entrées, donc commandable par un autre composant.
Il est défectueux.	Si aucune erreur dans les points précédents n'a été détecté, nous pouvons supposer qu'il est défectueux.

TABLE 24 – Cause probable du dysfonctionnement du convertisseur AD.

#### 4.13.1 Test du signal réel

Nous pouvons nous faire une idée des signaux à la sortie du filtre et du redresseur. Suivant la fréquence de ces derniers, ils doivent être omniprésents ou quasi inexistant si leur fréquence est trop élevée (plus de 50Hz).

Pour se faire nous avons branché sur la prise COAX un générateur de fréquence. Nous avons obtenu les résultats suivants :

**Signal sinusoïdal :** Nous avons généré un signal sinusoïdal à 10Hz environ. La mesure a été prise à la sortie du redresseur. Si nous envoyons un signal sinusoïdal, nous attendons un signal redressé avec une fréquence double. Le résultat obtenu est celui que nous attendions.

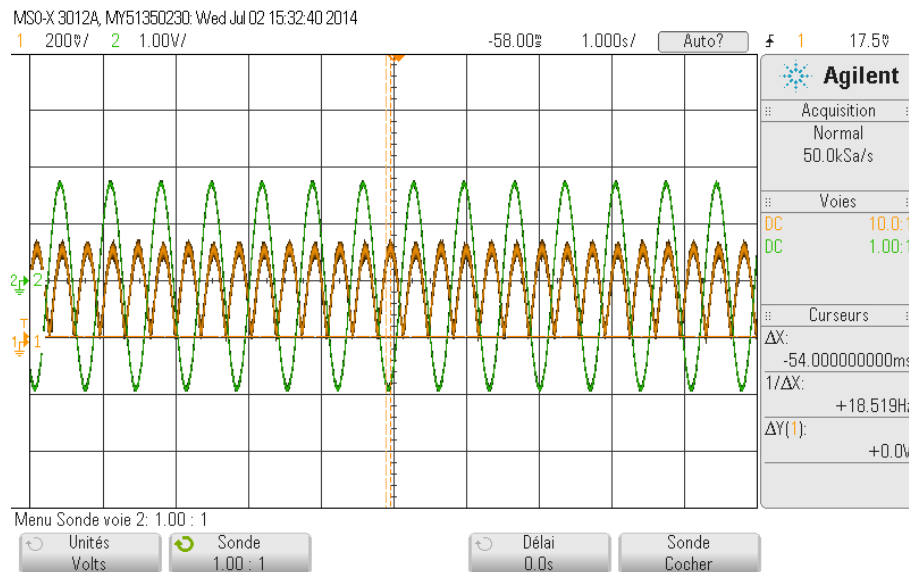


FIGURE 56 – Signal sinusoïdal redressé.

Pour le filtrage passe-bas du signal, nous devons avoir quasiment le même signal, sauf qu'il est déphasé de quelques degrés. Celui en vert est celui qui sort du générateur de fonction. Le orange, quant à lui, est la réponse à la sortie des filtres. On remarque qu'il est redressé et que son amplitude est atténuée. Il est atténué d'un facteur 10. En entrée du redresseur la tension est divisée par 5. Puis son amplitude est encore divisée par deux. Ceci vient probablement du fait que le filtre réel n'a pas totalement les mêmes caractéristiques que le filtre simulé.

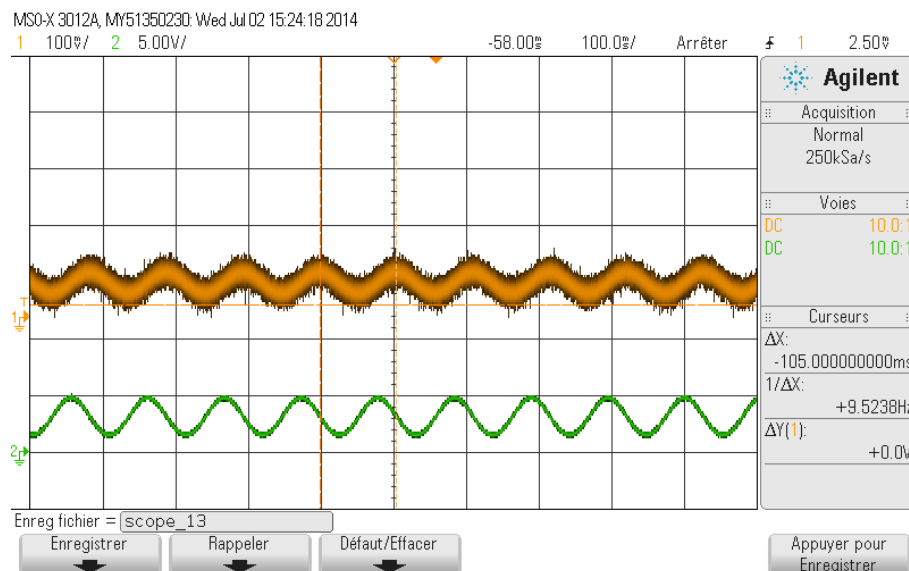


FIGURE 57 – Signal sinusoïdal après avoir été filtré passe-bas.

Nous avons obtenu ceci et constatons que nous avons un affaiblissement et un décalage dans le temps.

Par contre un signal avec une fréquence beaucoup plus haute, par exemple 100Hz, doit être presque plat car le générateur a une fréquence fixe et le filtre doit donc couper toutes ces fréquences.

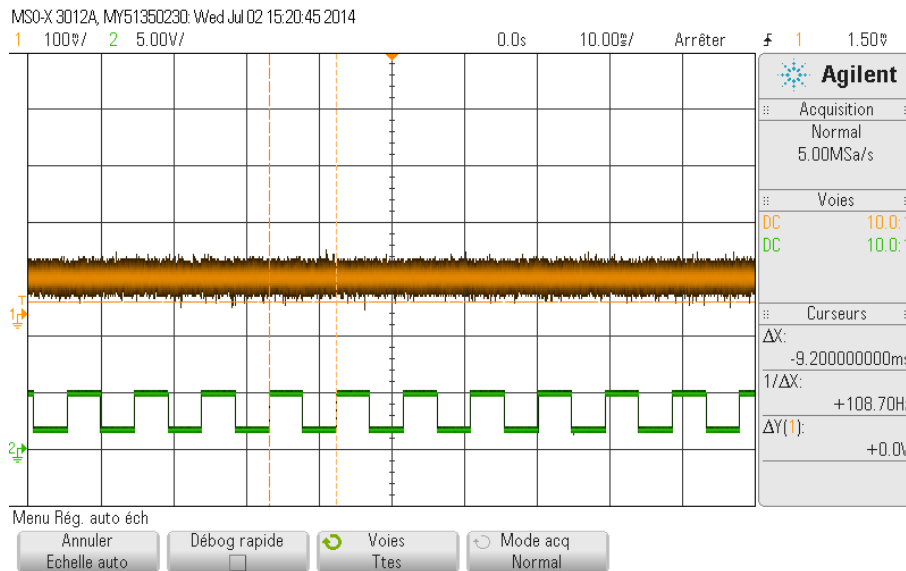


FIGURE 58 – Signal sinusoïdal à après avoir été filtré passe-bas.

Nous avons donc bien le résultat attendu, le signal est quasi nul car la fréquence est bien plus élevée.

#### 4.14 Calcul de la consommation de courant totale

Afin de calculer les différents courants consommés, nous avons placé des résistances  $0[\Omega]$  afin de pouvoir calculer les différents courants. On dé-soude donc les résistances et nous plaçons à la place un ampèremètre. Il va donc nous permettre de mesurer chaque consommation et donc de pouvoir par la suite changer l'un ou l'autre composant afin de réduire cette consommation.

Voici un tableau récapitulatif des différentes consommations des composants.

Consommation de courant	
<i>Composant</i>	<i>Courant [mA]</i>
FPGA	3.81mA
EEPROM	5mA
Convertisseur AD	3uA
Driver RS-232	3.3mA
Driver RS-422	non-testé
Régulateur LDO	0.1uA
Led	10mA
Oscillateur	40mA
Consommation totale	76mA

TABLE 25 – Choix de l'entrée

## 5 Discussion

### 5.1 Partie Matériel

La carte a été assemblée et les tensions mesurées étaient celles que nous attendions. Aucun composant n'a grillé et nous avons pu faire des tests à partir du code VHDL qui a été programmé.

Nous avons déjà remarqué que le convertisseur DC-DC ne donnait pas vraiment la bonne tension. C'est un convertisseur et non un régulateur, ce qui implique qu'au cas où la tension varie, nous n'aurons peut-être plus 3.3V. Et ceci peut être embêtant, donc c'est un élément à changer pour un prochain design. Le choix du convertisseur permet de baisser la tension, il serait donc possible d'ajouter un régulateur après afin de pouvoir réguler cette tension.

Nous avons prévu que le driver RS-232 s'éteigne lorsque nous ne l'utilisons pas. Or, en connectant les broches FORCEOFF et FORCEON à VCC et respectivement à la masse (comme mentionné dans le datasheet) et malgré les indications d'entrées valides, le driver n'arrive pas à communiquer. Nous avons donc dû connecter ces deux pins à VCC et ceci a permis aux composants de fonctionner correctement.

Pour les filtres analogiques, nous avons coupé la fréquence relativement bas, ce qui a pour effet d'affaiblir les oscillations et les pics d'amplitude. Il serait judicieux de couper avec une fréquence moins élevée afin de ne pas trop lisser le signal et donc d'avoir un signal plus précis avant l'échantillonnage. Une autre possibilité serait d'augmenter l'ordre du filtre afin d'avoir une pente à la fréquence de coupure qui soit plus raide et donc que les fréquences avant la fréquence de coupure soient moins atténuées.

Le convertisseur AD n'est pas des plus aisé à configurer mais grâce à ses trois entrées il nous a été très utile. Ces entrées peuvent être connectées au même géophone, celui qui possède un SUB-D 15 mais il est également possible pour une autre utilisation, de changer ce connecteur par trois connecteurs coaxiaux et donc de pouvoir connecter trois autres géophones ne possédant qu'un axe de mesure chacun. Pour faire du monitoring, il est plus judicieux d'avoir trois géophones pour pouvoir comparer les signaux qu'un seul géophone avec trois axes de mesure.

### 5.2 Partie Algorithmique

Différents algorithmes ont été testés sur des signaux réels afin de trouver un bon moyen de détecter les pics d'amplitude. Les algorithmes qui calculent des pentes, soit les trois premiers algorithmes, nous assurent de détecter des signaux qui ont des fréquences basses et laissent de côté les pics d'amplitudes pour des fréquences très élevées. Ceci est un très bon point pour ces algorithmes.

Quant à la détection de seuils, l'algorithme est facile à mettre en place, il détecte le moindre petit choc ce qui peut poser des problèmes dans le cas où un seul géophone serait en service. Si effectivement nous en plaçons plusieurs et triangulons, nous savons si la secousse qui a généré ce signal était aux alentours proches du géophone ou si cette secousse s'est ressentie à plusieurs endroits. Ceci permet de pouvoir mesurer la pertinence d'une secousse. L'algorithme est rapide, c'est une simple comparaison et donc ce qui limite éga-

lement la place prise dans la mémoire de la FPGA.

Nous pouvons encore émettre la remarque suivante, si nous n'envoyons qu'une seule valeur chaque minute, nous enverrons le seuil le plus haut atteint. Or si nous n'avons atteint le seuil numéro 4 qu'une seule fois et que nous avons atteint le 3, plus de 200 fois, il est à se demander quelle information est plus pertinente.

Les algorithmes qui calculent une pente sont aisés à mettre en place en Python. Or dans une FPGA, il est difficile de comprendre ce que c'est qu'un chiffre réel. L'idée serait de calculer un peu différemment : nous savons qu'un maximum se trouve entre une pente positive et une pente négative. Or la valeur de la pente ne nous intéresse pas beaucoup au final, nous cherchons juste un maximum local. A partir de ce raisonnement, nous ne regardons que la variation des amplitudes.

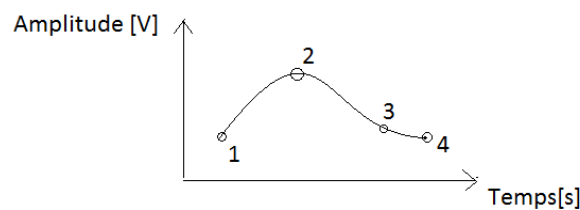


FIGURE 59 – Autre possibilité pour la détection de maximums.

Nous prenons la figure 59 ce qui va nous permettre d'illustrer plus aisément l'idée. Nous sommes au point 1. Nous sauvons son amplitude. une période d'échantillonnage plus tard nous sommes au point 2. Nous calculons la variations d'amplitudes, celle-ci vaut "Amplitude au point 2 - amplitude au point 1". Nous avons une réponse positive et nous sauvons cette valeur dans une variable de type  $\delta\text{Amplitude}$ . Nous passons ensuite au point 3. Nous répétons la même opération, c'est à dire, "Amplitude au point 3 - amplitude au point 2". La réponse est cette fois négative. Nous sauvons à nouveau cette valeur dans une nouvelle variable, par exemple  $\delta\text{AmplitudeNext}$ . Nous regardons ensuite le signe de chacune de ces variables et cela nous indique si nous avons un maximum et un minimum. Il est à noter que les deux variables  $\delta\text{Amplitude}$  doivent être de type signé.

Nous pouvons émettre une première critique concernant les algorithmes 1, 2 et 3. En effet, si un pic n'est suivi d'aucune oscillation, il n'y a plus de maximum et donc, la valeur ne change pas. Ce qui pose problème car le signal envoyé ne correspond pas au véritable signal. Ceci serait donc un point à modifier par la suite. Une idée pourrait être de regarder que si l'amplitude reçue est proche de zéros pendant un certain temps, et que nous sommes bloqués à une amplitude élevée, nous devrions mettre à jour la valeur de l'enveloppe.

### 5.3 Partie logicielle

De nombreux tests ont été effectués afin de contrôler le bon fonctionnement des blocs. Certains blocs fonctionnent bien, comme par exemple le bloc d'initialisation du convertisseur, le bloc ADC qui permet de récupérer les valeurs, et de les coder sur 16 bits. Le partie traitement du signal, fonctionne bien, mais nous avons un petit soucis entre ce bloc et le bloc mémoire. En effet, le bloc mémoire n'est pas le plus simple à réaliser.

Nous avons un petit problème d'exclusion mutuelle avec la mémoire. Nous avons implémenté une solution très basique. Mais nous pourrions améliorer ceci avec un véritable algorithme d'exclusion mutuelle pour être sûr que rien ne puisse perturber le système. Nous avons par exemple la priorité pour la lecture ou l'écriture, ou une égalité parfaite.

Pour la partie RS-232, les données reçues par le bloc sont ensuite envoyées sur la ligne communiquant avec le driver RS-232. La partie RS-422 n'a pas encore été implémentée par manque de temps.

Le convertisseur AD peut poser quelques soucis. En effet, en utilisant le protocole I2C, nous sommes limités par la fréquence du SCL qui est de 100kHz. Par contre en utilisant le protocole SPI, nous pouvons aller plus rapidement. Le temps minimum pour le Master Clock est de 500ns. Donc si nous gardons un peu de marge avec un temps de 550ns, ceci nous donne une fréquence de 1,81GHz, ce qui diminue considérablement le temps d'écriture dans le registre. Nous avons 8bits de registre et 8bits de données, soit un total de 16bits. Avec cette fréquence, nous avons donc transmis ces données en :

$$temps = \frac{nbrBit}{frquence} = \frac{16}{1,81*10^9} = 8.8ns$$

Nous devons donc échantillonner trois valeurs et changer trois fois de registre. Pour pouvoir toujours échantillonner à 10kHz environ, nous devons augmenter la fréquence d'échantillonnage. Dans le cas actuel, nous échantillonons à 31250Hz. Ce qui nous donne une période de  $1/31250Hz = 32\mu s$ .

Dans le cas où nous voudrions garder ces 32 $\mu s$ , nous devrions changer la fréquence d'échantillonnage. Donc si nous enlevons à 32 $\mu s$ ,  $3*8.8ns$  pour l'envoi des données par SPI, nous avons une période de 31.97 $\mu s$ . Ceci correspond donc à une fréquence de 31275Hz. nous pourrions donc prendre 32kHz.

## 5.4 Améliorations futures

Nous pouvons dans le futur apporter les améliorations suivantes afin de rendre le produit plus attractif :

Au niveau de la carte, il est tout à fait possible de diminuer sa taille, en mettant des composants plus petits et en les plaçant également sur le dessous de la carte. Les connecteurs SUB-D 9 et SUB-D 15 sont difficilement réductible, et ceci bloque en partie la réductibilité de la carte. Nous avons remarqué que l'oscillateur demandait beaucoup de courant, dans les 40mA. Il est possible de réduire cette fréquence et donc la consommation de courant. La fréquence maximale que nous utilisons est celle de 2MHz pour le protocole I2S. Nous pouvons donc remplacer l'oscillateur par un plus petit. Or en faisant ceci nous réduisons également la vitesse de calcul et de transmission. Une donnée de 16bits étant transmise en 1  $\mu s$  à 16MHz, elle mettra désormais 8 $\mu s$  avec 2MHz. Il faudrait donc optimiser le code afin de réduire le traitement car sinon la FPGA pourrait ne pas arriver à faire ses calculs avant que la valeur suivante arrive.

Les composants suivants seraient à changer : Le convertisseur DC-DC ne régule pas la tension et peut provoquer des soucis si la tension n'est pas parfaitement linéaire. En effet,



il ne régule pas la tension. Lors de nos mesures, avec une tension de 5Volts à son entrée, il nous transmettait en sortie une tension de 3.7V. Le datasheet demande environ 10% de la charge. Le convertisseur peut fournir un courant de 300mA et donc à 10 % de la charge il fournira 30mA. Or notre carte demande 75mA, nous sommes donc au dessus de ce qui est recommandé par le fournisseur, mais la tension de sortie du convertisseur n'est pas correcte.

Les algorithmes fonctionnent bien, nous pouvons observer que le signal est bien respecté pour chacun des trois algorithmes qui traitent des pentes, mais que certains sont plus précis que d'autres. Mais ils peuvent également être optimisés afin de diminuer le temps de calcul et donc la consommation. Une possibilité pour augmenter l'efficacité et la véracité des valeurs transmises à un serveur, serait par exemple de traiter le signal selon deux algorithmes et d'en faire ensuite la synthèse. Par exemple, nous pourrions prendre l'algorithme de détection des seuils, ceci indique donc le nombre de fois qu'un certain seuil a été dépassé, mais nous pourrions le coupler avec l'algorithme numéro 1 qui cherche à déterminer le temps de montée et de descente d'une amplitude. Ceci permettrait donc de laisser de côté les hautes fréquences et de ne retenir que les fréquences plus basses. Enfin nous pourrions incrémenter la valeur du seuil correspondant seulement dans le cas où les pics a une fréquence suffisamment basse pour être retenue.

La FPGA consomme peu de courant, environ 4mA. Son taux d'occupation est de l'ordre de 90% mais le code n'est pas optimisé. Dans cette optique il est possible de le réduire et donc de diminuer ce taux. Par contre, pour des traitements plus importants ou si plusieurs géophones sont ajoutés ( plus de 3), il serait judicieux de se tourner vers une FPGA plus grande, par exemple une AGL250.

Le convertisseur AD a été un bon choix quant au fait qu'il possède de nombreux points qui peuvent être configuré, comme par exemple la fréquence d'échantillonnage, les protocoles de communication mais aussi sur le fait qu'il possède 3 entrées différentielles, ce qui peut être intéressant dans le cas où nous aimerions connecter trois géophones sur une même carte. Sa taille est également restreinte ( 5mm \* 5mm \* 1mm) et sa consommation de l'ordre de 1mA. En contrepartie, ces trois entrées ne peuvent être lues en même temps. Elles sont reliées à un multiplexeur qu'il faut sans cesse changer pour choisir la bonne entrée. Et donc le temps de calcul est rallongé. Mais nous pouvons utiliser le protocole SPI qui permet d'augmenter sensiblement (d'un facteur de 20 environ), la vitesse de transmission par rapport à celle de l'I2C.

Du côté du software, il serait possible d'endormir les composants lorsque ceux-ci ne sont pas utilisés. Ceci pourrait limiter la consommation, comme les drivers RS-232 et 422 par exemple.

## 6 Conclusion

La fondation CREALP, nous a demandé de créer un prototype permettant l'acquisition de données provenant d'un géophone, qui soit autonome et qui puisse être placé en milieu montagnards.

Durant ce projet, nous avons réalisé une carte électronique Low Power (70mA) munie d'une FPGA, d'un convertisseur AD, d'une mémoire ainsi que de filtres pour traiter le signal. Plusieurs tests ont été réalisés sur la carte et à ce stade nous avons le convertisseur AD qui n'émet encore aucune donnée. En contrepartie, possédant une FPGA, nous avons pu simuler sur ordinateur le fonctionnement de la FPGA et la grande majorité des fonctions ont été validées. Il reste encore la partie mémoire où nous devons effectuer des modifications afin de ne pas lire et écrire en même temps.

Plusieurs algorithmes ont été mis au point, certains détectent les changements de pentes, tandis que d'autres détectent uniquement l'amplitude des signaux. Nous stockons soit les différentes amplitudes maximales trouvées en fonction du temps, soit le nombre de fois que le seuil le plus important a été dépassé.

Pour terminer, il reste à vérifier le bon fonctionnement du convertisseur AD, l'implémentation de la fonction pour la transmission sur les lignes RS-422 et la correction de certains problèmes de lecture et d'écriture simultanée dans la mémoire.

## 7 Date et signature

Sion, le 10 juillet 2014

Romain Franzen

## 8 Référence

- [1] MICROSEMI, IGLOO nano Low Power Flash FPGAs with Flash\*Freeze Technology [En ligne], 2013.  
[http://www.actel.com/documents/IGLOO\\_nano\\_DS.pdf](http://www.actel.com/documents/IGLOO_nano_DS.pdf) (Consulté le 23.03.2014)
- [2] SEISMICNET, New and used geophones for sale [en ligne].2014.  
[seismicnet.com/geophone/index.html](http://seismicnet.com/geophone/index.html) (Consulté le 27.02.2014)
- [3] LGT 4.5Hz SENSOR TESTING BY MICHAEL CRANFORD [en ligne].2013.  
[http://seismicnet.com/geophone/lgt45\\_testing.html](http://seismicnet.com/geophone/lgt45_testing.html) (Consulté le 27.02.2014)
- [4] MICROSEMI, Igloo Nano Starter Kit User's Guide [En ligne], 2013.  
[http://www.actel.com/documents/IGLOO\\_nano\\_DS.pdf](http://www.actel.com/documents/IGLOO_nano_DS.pdf) (Consulté le 23.03.2014)
- [5] SCIENCE IN SCHOOL, Building a seismograph from scrap, [en ligne].  
[www.scienceinschool.org/repository/images/issue23earthquakes14\\_1.jpg](http://www.scienceinschool.org/repository/images/issue23earthquakes14_1.jpg) (Consulté le 29.04.2014).
- [6] WIKIPEDIA, Géophone,[en ligne].  
[fr.wikipedia.org/wiki/G%C3%A9ophone](http://fr.wikipedia.org/wiki/G%C3%A9ophone) (Consulté le 2.05.2014).
- [7] EEM, Electronic products [en ligne], 2014.  
[www2.eem.com/Actel\\_Corp\\_.aspx](http://www2.eem.com/Actel_Corp_.aspx) (Consulté le 18.03.2014).
- [8] MICROSEMI, [en ligne]. 2013.  
[www.microsemi.com/products/fpga-soc/power/comparison](http://www.microsemi.com/products/fpga-soc/power/comparison) (Consulté le 12.04.2014).
- [9] FARNELL, Composants électroniques, pièces électroniques [en ligne], 2014.  
[ch.farnell.com/actel/agln125v2-vqg100/fpga-low-power-igloo-100vqfp/dp/2061102?ref=lookahead](http://ch.farnell.com/actel/agln125v2-vqg100/fpga-low-power-igloo-100vqfp/dp/2061102?ref=lookahead) (Consulté le 12.04.2014).
- [10] FARNELL, Composants électroniques et pièces électroniques [en ligne], 2014.  
<http://ch.farnell.com/jsp/search/productdetail.jsp?id=1561964&Ntt=1561964> (Consulté le 18.05.2014).
- [11] MOUSER ELECTRONICS, Distributor elektronischer Bauteile [en ligne], 2014.  
<http://ch.mouser.com/ProductDetail/Hirose-Electric/ZX62D-AB-5P8/?qs=%2fha2pyFadujQUnsCaaZSAdOkzEbiM7IWsu0JtqjHnJldtWCmWcE7lQ%3d%3d> (Consulté le 20.05.2014).
- [12] MICROSEMI, [en ligne]. 2013.  
[www.microsemi.com/products/fpga-soc/power/comparison](http://www.microsemi.com/products/fpga-soc/power/comparison) (Consulté le 12.04.2014).
- [13] FARNELL, Composants électroniques et pièces électroniques [en ligne], 2014.  
<http://ch.farnell.com/intersil/isl83491ibz/ic-transceiver-rs-485-rs-422-3/dp/1562055?Ntt=1562055> (Consulté le 4.06.2014)
- [14] MOUSER ELECTRONICS, Distributor elektronischer Bauteile [en ligne], 2014.  
<http://ch.mouser.com/ProductDetail/Texas-Instruments/PCM3794ARHBT/?qs=sGAE->

piMZZMtq3QB8qGen7cwPmLfcwVowbVARtpkYMHA%3d (Consulté le 1.05.2014)

[15] WIKIMEDIA COMMONS, [en ligne], 2014.

[http://commons.wikimedia.org/wiki/File:Echantillonnage\\_sinus.JPG](http://commons.wikimedia.org/wiki/File:Echantillonnage_sinus.JPG) (Consulté le 10.07.2014)

## 8.1 Datasheets

Voici la liste des datasheets qui peuvent être consulté directement en ligne ou sur le CD fourni avec le rapport.

ACTEL - AGLN125V2-VQG100 - FPGA, LO-PWR, IGLOO, 100VQFP :  
[www.farnell.com/datasheets/1738886.pdf](http://www.farnell.com/datasheets/1738886.pdf)

Texas Instrument PCM3794ARHBT :  
<http://www.ti.com/lit/ds/slas529a/slas529a.pdf>

Small Signal Zener Diodes :  
[www.vishay.com/docs/85614/tzx-seri.pdf](http://www.vishay.com/docs/85614/tzx-seri.pdf)

MICROCHIP - 24AA1026-I/SM - EEPROM SERIE 1024K 8SOIC :  
<http://www.farnell.com/datasheets/1669522.pdf>

INTERSIL - ICL3221EIAZ - IC, TRANSMITTER RECEIVER, RS-232 :  
<http://www.farnell.com/datasheets/32527.pdf>

XP POWER - ISF0503A - CONVERTISSEUR DC/DC BOITIER CMS 1W 3.3V :  
[www.xppower.com/pdfs/SF\\_ISF.pdf](http://www.xppower.com/pdfs/SF_ISF.pdf)

INTERSIL - ISL83491IBZ - IC TRANSCEIVER, RS-485/ RS-422, 3.3V  
<http://www.farnell.com/datasheets/31954.pdf>

LITTELFUSE - 02292.25MXP - FUSIBLE TEMPORISE 2AG 2.25A  
<http://www.farnell.com/datasheets/48329.pdf>

Oscillateur ABRACON - ASFLMB-BLANK-XY - MEMS OSC PROG  
1-150MHZ CMS 5X3.2 : [www.farnell.com/datasheets/1670837.pdf](http://www.farnell.com/datasheets/1670837.pdf)

LITTELFUSE - 02540101Z - FUSE BLOCK, 5 X 15MM  
[http://www.littelfuse.com/data/en/data\\_sheets/littelfuse\\_254\\_omniblock.pdf](http://www.littelfuse.com/data/en/data_sheets/littelfuse_254_omniblock.pdf)

VISHAY DALE - IMC0805ERR82J01 - HF INDUCTOR, 820NH, 150MA 5% 550MHZ  
<http://www.vishay.com/docs/34115/imc08051.pdf>

STMICROELECTRONICS - TSV621ILT - OP AMP, RRIO, 420KHZ, MICROPWR,  
5SOT23 <http://www.farnell.com/datasheets/1690544.pdf>

3M - 2510-6002UB - EMBASE MALE DROITE 10 CONTACTS  
[multimedia.3m.com/mws/mediawebserver?6666660Zjcf6lVs6EVs66SyIrCOrrrrQ-](http://multimedia.3m.com/mws/mediawebserver?6666660Zjcf6lVs6EVs66SyIrCOrrrrQ-)

HARTING - 09 64 213 7802 - CONNECTEUR SUB D FEMELLE FILTRE US. 15 V  
[www.farnell.com/datasheets/524731.pdf](http://www.farnell.com/datasheets/524731.pdf)

TE CONNECTIVITY / AMP - 1734348-1 - SUB-D FEMELLE  
COUDEE MONTAGE CI 9 VOIES  
[ch.farnell.com/te-connectivity-amp/1734348-1/sub-d-femelle-coudee-montage-ci](http://ch.farnell.com/te-connectivity-amp/1734348-1/sub-d-femelle-coudee-montage-ci)

/dp/1228088?Ntt=TE+CONNECTIVITY+//+AMP+-+1734348-1  
+-+SUB-D+FEMELLE+COUDEE+MONTAGE+CI+9+VOIES

Connecteurs USB MICRO AB RECEPT RA TH BTM MNT :  
<http://www.mouser.com/ds/2/185/e24200011-37133.pdf>

Reset Manuel MAX811TEUS+T :  
[www.mouser.com/ds/2/256/MAX811-MAX812-190465.pdf](http://www.mouser.com/ds/2/256/MAX811-MAX812-190465.pdf)

lastPage

# Appendices

1. Divers type de géophones Figure :60
2. Sortie par rapport à la fréquence du LGT Figure : 61 et facteur d'amortissement
3. Configuration des registres du convertisseur AD
4. Calcul des résistances pour le régulateur 1.5V
5. Convertisseur DC-DC 5V - 3.3V
6. Informations sur les broches du régulateur TPS70402
7. Broches du JTAG
8. Broches et informations sur l'interface SUB-D 15
9. Informations utiles pour la consommation de la FPGA
10. Liste des Igloos
11. Pin compatibles pour les 3 FPGAs Igloo
12. Énumération des broches utilisées et connectées à la FPGA
13. Protocole de communication RS-232
14. Broches RS-232
15. Bloc I2C de la HEI
16. Traitement du signal en Python
17. Calculs pour le filtre passif
18. Calculs pour le filtre actif
19. Calcul de la résistance pour la diode Zener
20. Tensions de sorties maximales et minimales du géophone
21. Broche du convertisseur AD
22. Réponse en fréquence du filtre passe-bas actif
23. Schéma électrique
24. Schéma du PCB

## Annexe 1 :

<b>Instruction</b>					
LGT seismic geophone element is designed to offer long service life in seismic exploration, its bandwidth is 10 Hz to 190 Hz in 2D and 3D seismic exploration. and it can be installed in variety of LGT geophone case					
<b>Specifications</b>					
Type	LGT-4.5	LGT-10	LGT-14	LGT-28	LGT-35
Natural Frequency (Hz)	4.5±10%	10 ± 5%	14 ± 5%	28 ± 5%	35 ± 5%
Open Circuit Damping	0.6 ±10%	0.271 ± 5%	0.2 ± 5%	0.6 ± 5%	0.6 ± 5%
Close Circuit Damping		0.6 ± 5%	0.6 ± 5%		
Intrinsic Voltage Sensity (V/cm/s)	0.288 ± 10%	0.288 ± 5%	0.288 ± 5%	0.31 ± 5%	0.325 ± 5%
Sensitivity with Shunt Resistance (V/cm/s)		0.227 ± 5% (1400Ω)	0.182 ± 5%		
Coil Resistance	375 ± 5%	375 ± 5%	375 ± 5%	385 ± 5%	560 ± 5%
Close Direct Current Resistance		296 ± 5%	236.5 ± 5%		
Harmonic Distortion	≤0.3%	≤0.2%	≤0.2%	≤0.2%	≤0.2%
Typical Spurious Frequency	≥180	≥200	≥250	≥400	≥500
Moving Mass (g)	11.3	11.3	11.6	10.2	9.3
Coil Excursion P-P (mm)	4	2	2	1.6	1.8
Maintains Fn Specifications to Tilt Angle	≤10	≤20	≤20	90	90
Core Unit Diameter (mm)	25.4	25.4	25.4	26	26
Core Unit Height (mm)	36	32	32	32	32
Core Unit Mass (g)	74	74	74	95	95
Operating Temperature Range	-40~+90	-40~+90	-40~+90	-40~+90	-40~+90
Warranty (year)	1	3	3	3	3

FIGURE 60 – Divers type de géophones, nous prenons le LGT-4.5.



## Annexe 2 :

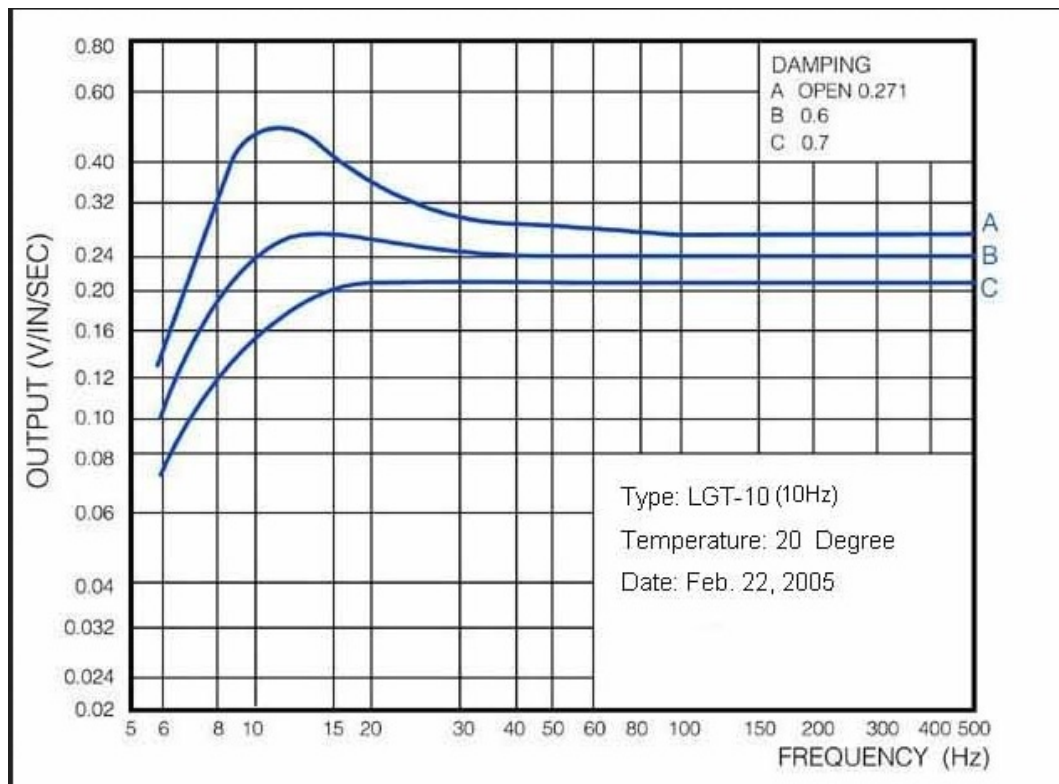


FIGURE 61 – Sortie par rapport à la fréquence du LGT-4.5

Le damping factor ou facteur d'amortissement permet de connaître l'état transitoire d'un système.

- Apériodique si :  $\zeta > 1$
- Critique si :  $\zeta = 1$
- Pseudo-Critique si :  $\zeta < 1$
- Harmonique si :  $\zeta = 0$

Annexe 3 :

**Configuration des registres du convertisseur AD** Plusieurs registres du convertisseur AD doivent être configurés avant de pouvoir utiliser ce dernier correctement.

Les registres à configurer sont les suivants :

- 79-82
- 84-87
- 90

Les valeurs "x" sont des variables réservées, elles ne sont donc pas à configurer.

**Registre 79 : Volume for ADC Input L**

Configuration registre : 01001111xx001100

- Level Gain 0db : 001100

**Registre 80 : Volume for ADC Input R**

Configuration registre : 01010000xx001100

- Level Gain 0db : 001100

**Registre 81 : ADC High pass filter, soft mute, audio interface**

Configuration registre : 010100011100x101

- HPF : 1 1 : Filtre disabled
- RMUL : 0 : Disable, pas besoin d'avoir un accroissement de volume
- RMUR : 0 : identique à RMUL
- DSMC : 0 : No delay
- RFM : 01 : right justified, ceci nous permettra de prendre les 16bits qui arrivent a partir du 1er.

**Registre 82 : ADC,MCB, PG1,2,5,6, D2S power up/down**

Configuration registre : 01010001xx000000

- PAIR : 0 : Default
- PAIL : 0 : Default
- PADS : 0 : Default
- PMCB : 0 : Not used
- PADR : 0 : Default
- PADL : 0 : Default

**Registre 84 : MasterMode**

Configuration registre : 01010100xxxxx011

- MSTR : 0 : Il est esclave
- BITO : 1 : 16 bits
- SRST : 1 : Reset enable

**Registre 85 : System reset, sampling rate control**

Configuration registre : 010101011x000000

- SRST : 1 : Reset Enable
- NPR[5 :0] : 00'0000 : System clock rate selection, system audio clock for common frequency.

**Registre 86 : BCK Configuration, sampling rate control, zero-cross**

Configuration registre : 0101011000100xx0

- MSR[2 :0] : 011 : Select the system clock rate
- MBST : x : Car nous ne sommes pas en mode master
- ATOD : 0 : Disabled (loopback)
- ZCRS : 0 : Mais pas sur du tout

**Registre 87 : Analog input select (MUX 1,2,3,4)**

Configuration registre : 0101011110xx00xx

A la place des "-" on met les valeurs qui correspondent aux mux voulu.

- ADS2 : 1 : Differential Input Amplifier
- AIL[1 :0] : xx : Variable pour choisir les entrées du Mux
- AIR[1 :0] : xx : Variable pour choisir les entrées du Mux

**Registre 90 : VCC Power Up/Down, ramp up/down time, boost**

Configuration registre : 01011010xx10xx00

- CMT[1 :0] : 10 : Car R<sub>cmt</sub>=33[kΩ]
- G20L : 0 : Pas -20dB boost
- G20R : 0 : Pas -20dB boost

Annexe 4 :

**Calcul des résistances :**

Le but est d'avoir une tension V<sub>0</sub> de 1.5V. Selon le datasheet, la résistance R<sub>2</sub> doit valoir 30,1[kΩ]. Et V<sub>ref</sub> = 1,224V. Ceci donne donc une résistance R<sub>1</sub>.

$$R1 = \left( \frac{1.5}{1.224} - 1 \right) * 30'100 = 6,7[k\Omega]$$

Afin d'arriver à 30,1[kΩ], il est nécessaire de placer plusieurs résistances en série car la résistance R<sub>2</sub> n'est pas normalisée. Les résistances placées en série sont donc : 18[kΩ] + 12[kΩ] + 100[Ω].

Une capacité à la sortie doit également être placée et celle-ci aura la valeur de 22uF.

**Informations supplémentaire sur certaines broches.**

La broche PG = Power Good. Cette broche rapporte les conditions de tensions de V<sub>out</sub>1. Elle peut être utilisée pour implémenter un Reset ou Power on reset pour la circuiterie du régulateur numéro 1.

RESET : Il est actif à l'état bas et demande une pull-up pour les opérations normales. Le but est de limiter grandement la consommation de courant, nous allons donc prendre une résistance relativement élevée. Grâce à l'équation suivante :

$$U = R * I \Rightarrow I = \frac{U}{R} = \frac{3,3V}{100'000[\Omega]} = 33[uA]$$

Nous plaçons donc une résistance de 100[kΩ] pour faire une pull-up.

MR, actif à l'état bas, permet de contrôler V<sub>lout</sub>. La broche de sortie PG1 peut être connectée à MR (Reset manuel).

Pour que le reset passe à l'état haut, il faut que V<sub>lin</sub> soit au dessus de l'état de sous-tension et que la broche manuelle reset(MR qui est actif à l'état bas) soit en haute impédance. Pour contrôler V<sub>out1</sub>, la broche PG1 qui est en sortie doit être connectée à MR. Reset peut être utilisée pour driver le Power-On-Reset. Si Reset n'est pas utilisé, la sortie peut être laissée flottante. Or nous ne voulons pas forcément de Reset, le cœur de la FPGA étant tout le temps actif.

EN1 est actif à l'état bas.

Quand V<sub>out1</sub> atteint 95 % de la tension régulée, PG1 se trouve en haute impédance. La sortie Open-Drain du terminal PG requière une résistance pull-up. De plus, PG1 indique le statut de chaque régulateur.

Une capacité de découplage de 100nF est nécessaire à la broche d'alimentation afin de limiter les oscillations de la tension. La broche "enable"(active à l'état bas) est directement branchée à la tension d'alimentation. Ainsi elle est toujours active.

La broche d'entrée Reset est branchée au Power-On-Reset du circuit effectuant le Reset. La broche PG1 (power good) et MR(manuel Reset), actifs à l'état bas sont connectés ensemble. Ils sont pourvus de résistance 0ohm afin de pouvoir débrancher le cas échéant. En effet, le datasheet n'est pas très explicite, c'est pourquoi cette option a été choisie.

Au niveau des sorties, nous avons des résistances qui permettent de choisir la tension que nous désirons. Le calcul des résistances peut être trouvé dans le chapitre *Liste du Matériel, sous-chapitre Régulateur 3.3V - 1.5V :TPS70402* qui se situe à la page 12.

Annexe 5 :

Convertisseur DC-DC 5V - 3.3V

Les broches sont organisées de la façon suivante :

- Pin 1 : Input Voltage -
- Pin 2 : Input Voltage +
- Pin 3 : N/A
- Pin 4 : 0 V
- Pin 5 : Output Voltage +
- Pin 6 : N.C.
- Pin 7 : N.C.
- Pin 8 : N.C.

Le prix de ce matériel est de : 9.90.-. Ce composant peut être commandé chez Farnell<sup>1</sup>

---

1. FARNELL, Composants électroniques et pièces électroniques [en ligne], 2014.  
[ch.farnell.com/xp-power/isf0503a/convertisseur-dc-dc-boitier-cms/dp/1776111](http://ch.farnell.com/xp-power/isf0503a/convertisseur-dc-dc-boitier-cms/dp/1776111)

Annexe 6 :

Broches du JTAG

- TMS : Test Mode Select.
- TCK : Test Clock
- TDI : Test Data Input
- TDO : Test Data Output
- TRST : Test Reset

Annexe 7 :

Broches et informations sur l'interface SUB-D 15

Nous avons trouvé chez Farnell une embase SUB-D 15 : **HARTING - 09 64 213 7802 - CONNECTEUR SUB D FEMELLE FILTRE US. 15 V** . Cette pièce<sup>2</sup> peut être commandée au prix de SFr : 8.05.



FIGURE 62 – Interface Sub-D 15

Les broches de sortie du géophone sont connectées de la façon suivante :

- D-Sub DB-15 Connector Pinout for the PSN-ADC24 Digitizer board :
- Pin 1 - Solid Orange Wire = Vertical Sensor Coil + Signal
- Pin 2 - Orange/White Wire = Vertical Sensor Coil - Signal
- Pin 3 - Solid Green Wire = Horizontal Sensor Coil #1 + Signal
- Pin 4 - Green/White Wire = Horizontal Sensor Coil #1 - Signal
- Pin 5 - Solid Blue Wire = Horizontal Sensor Coil #2 + Signal
- Pin 6 - Blue/White Wire = Horizontal Sensor Coil #2 - Signal
- Pin 11 = Ground for Foiled Shielded and Drain wire. Not connected inside the sensor.

2. FARNELL, Composants électroniques et pièces électroniques [en ligne], 2014.  
[ch.farnell.com/harting/09-64-213-7802/connecteur-sub-d-femelle-filtre/dp/1779284](http://ch.farnell.com/harting/09-64-213-7802/connecteur-sub-d-femelle-filtre/dp/1779284)

## Annexe 8 :

## Informations utiles pour la consommation de la FPGA

- Carte SD : Clock, Data 0-1-2-3, command line := 6Pins
- Sub-D 15 : 3 \* 2 directions := 6pins
- AIC-111 : Data, IMODE, SCLK,FRAME,RESET ?? := 4-5Pins
- Oscillateur : Output, Standby := 2Pins
- JTAG : TMS, TCK, TDO, TDI, TRST := 5Pins
- RS-232 : R-Data,T-Data, DTE-Ready, DCE-Ready, Clear to Send, Request to Send, Receive line signal Detect, Ring Indicator := 8Pins
- Reset : POR :=1broche

Il faut donc au total := **33 broches**.

- Nombre de I/Os : 33
- Nombre de Spines :
- Nombre de Versatiles Row utilisées :
- Fréquence globale du système :
- Nombre de Versatiles séquentiel utilisées :
- Toggle rate of the Versatiles output :
- Nombre de Verstiles combinatoires utilisées :
- Nombre de I/Os en Input buffer :
- Nombre de I/Os en Output buffer :
- Buffer Toggle rate :
- I/O buffer enable rate :
- Fréquence pour la lecture dans la RAM :
- Fréquence pour l'écriture dans la RAM :
- Fréquence pour le Write Enable de la RAM :
- Fréquence pour le Read Enable de la RAM :
- Fréquence de sortie du clock de la PLL :

## Annexe 9 :

## Liste des Igloos

IGLOO Devices	AGL015 <sup>1</sup>	AGL030	AGL060	AGL125	AGL250	AGL400	AGL600	AGL1000
ARM-Enabled IGLOO Devices <sup>2</sup>					<b>M1AGL250</b>		<b>M1AGL600</b>	<b>M1AGL1000</b>
System Gates	15,000	30,000	60,000	125,000	250,000	400,000	600,000	1,000,000
Typical Equivalent Macrocells	128	256	512	1,024	2,048	–	–	–
VersaTiles (D-flip-flops)	384	768	1,536	3,072	6,144	9,216	13,824	24,576
Flash*Freeze Mode (typical, $\mu$ W)	5	5	10	16	24	32	36	53
RAM kbits (1,024 bits)	–	–	18	36	36	54	108	144
4,608-Bit Blocks	–	–	4	8	8	12	24	32
FlashROM Kbits (1,024 bits)	1	1	1	1	1	1	1	1
AES-Protected ISP <sup>2</sup>	–	–	Yes	Yes	Yes	Yes	Yes	Yes
Integrated PLL in CCCs <sup>3</sup>	–	–	1	1	1	1	1	1
VersaNet Globals <sup>4</sup>	6	6	18	18	18	18	18	18
I/O Banks	2	2	2	2	4	4	4	4
Maximum User I/Os	49	81	96	133	143	194	235	300
Package Pins								
UC/CS		UC81 CS81	CS121 <sup>3</sup>	CS196	CS81, CS196 <sup>5</sup>	CS196	CS281	CS281
QFN	QN68	QN48, QN68, QN132	QN132	QN132 VQ100 FG144	QN132 <sup>5,6</sup>			
VQFP FBGA		VQ100	VQ100 FG144 <sup>6</sup>		VQ100 FG144	FG144, FG256, FG484	FG144, FG256, FG484	FG144, FG256, FG484

FIGURE 63 – Liste des FPGA IGLOO



Global Pin which are the same for the 3 FPGAs, Quadrant Global

PIN Number	AGL060	AGL125	AGL250
2	GAA2	GAA2	GAA2
4	GAB2	GAB2	GAB2
6	GAC2	GAC2	GAC2
20	GEB1	GEB1	-----
21	GEB0	GEB0	-----
22	GEA1	GEA1	GEA1
23	GEA0	GEA0	GEA0
26	GEA2	GEA2	GEA2
27	GEB2	GEB2	GEB2
28	GEC2	GEC2	GEC2
43	-----	-----	GDC2
44	GDC2	GDC2	GDB2
45	GDB2	GDB2	GDA2
46	GDA2	GDA2	-----
57	GDA1	GDA1	GDA1
58	GDC0	GDC0	GDC0
59	GDC1	GDC1	GDC1
70	GBC2	GBC2	GBC2
71	GBB2	GBB2	GBB2
73	GBA2	GBA2	GBA2
76	GBA1	GBA1	GBA1
77	GBA0	GBA0	GBA0
78	GBB1	GBB1	GBB1
79	GBB0	GBB0	GBB0
80	GBC1	GBC1	GBC1
81	GBC0	GBC0	GBC0
93	GAC1	-----	GAC1
94	GAC0	-----	GAC0
95	GAB1	GAC1	GAB1
96	GAB0	GAC0	GAB0
97	GAA1	GAB1	GAA1
98	GAA0	GAB0	GAA0
99	-----	GAA1	-----
100	-----	GAA0	-----

Autres entrées sorties identiques		
Pin Number		
1	3	5
7	8	9
12	14	17
18	25	29
30	31	32
33	34	35
36	37	38
40	41	42
47	48	49
51	52	53
54	55	56
67	68	69
72	75	82
83	84	85
86	87	88
89	90	91
92		

Se sont de simple entrées sorties.

75 pins identiques

Global Pin which are the same for the 3 FPGAs, Chip Global

PIN Number	AGL060	AGL125	AGL250
10	GFB1	GFB1	GFB1
11	GFB0	GFB0	GFB0
13	GFA0	GFA0	GFA0
15	GFA1	GFA1	GFA1
16	GFA2	GFA2	GFA2
60	GCC2	GCC2	-----
61	GCB2	GCB2	GCB2
62	GCA0	GCA0	GCA1
63	GCA1	GCA1	GCA0
64	GCC0	GCC0	GCC0
65	GCC1	GCC1	GCC1

Les pins sur le chip global seront prise pour le clock par exemple

## Annexe 11 :

La AGL125 comporte plusieurs types d'entrées-sorties. Les FPGAs sont divisées en quadrants. Ces quadrants possèdent un certain nombre d'entrées-sorties. Il y a donc les "Quadrant global location", ce sont donc les I/O qui sont commune à un quadrant et les "Chip global location" qui sont communes à l'ensemble des quadrants. Ces broches qui sont placées sur le "Chip global" sont utilisées pour le clock par exemple qui peut être pris par toutes les broches.

Entrées-Sorties de la FPGA		
<i>Nom I/O FPGA</i>	<i>Direction</i>	<i>Composant</i>
clk	<-	oscillateur
mode_control_clock	->	PCM3794
mode_control_data	<->	PCM3794
Input_clock	->	PCM3794
RX_RS232	<-	RS-232
TX_RS232	->	RS-232
SCL	->	EEPROM, JTAG
OUT_RS4222	<-	RS-422
OUT_RS4221	<-	RS-422
Driver_Input	->	RS-422
Driver_In_Enable	->	RS-422
Mode_Control_select	->	PCM3794
Clock_Bit	<->	PCM3794
Sampling_Clock	->	PCM3794
SPI_I2C	->	PCM3794
JTAG_IO	<->	JTAG
DOUT_FPGA	->	PCM3794
POR	<-	Reset
ADC_3IN	<-	Switch
SDA	<->	EEPROM, JTAG
GSM_Module_ON	->	Transistor
DIN_FPGA	->	PCM3794
VALID_RS232	<-	RS-232

TABLE 26 – table :Table des entrées-sorties

Annexe 12 :

Pour la communication RS232.

Protocole de transmission :

- 1 bit : Start
- 7/8 bits : Données
- 1 bit : Parité
- 1 bit : Stop

Vitesse max de transmission en fonction de la longueur du câble :

Vitesse max de transmission	
<i>Débits [bit/s]</i>	<i>Longueur[m]</i>
19200	15
9600	150
4800	300
2400	900

TABLE 27 – table :Vitesse pour le RS-232

- FORCEON et FORCEOFF(actif à l'état bas), détermine le IC's mode. Pour que les opérations soient toujours activées, les deux sont à l'état haut. Pour activer le mode "powerdown", on peut contrôler de façon logique ou software en agissant seulement sur FORCEOFF(active à l'état bas). Si les deux sont à la masse, nous sommes en powerdown normal.
- Pour limiter la consommation durant le powerdown, le récepteur peut aussi être désactivé en mettant la ligne EN(active à l'état bas) à l'état logique haut.
- Invalid, qui est actif à l'état bas, indique en permanence si il y a un signal RS-232 valide ou non. Il n'y a pas de signal valide dans le cas ou INVALID est à l'état bas.
- Ce système comporte également un powerdown automatique.
- En : Est actif à l'état bas et permet d'activer ou non le composant. Il sera mis à la masse afin de l'activer de façon permanente.

Annexe 13 :

Les broches ont les caractéristiques suivantes :

Broches RS-232	
<i>Numéro broche</i>	<i>Fonction</i>
1	Data Carrier Detect : Détection d'un signal sur la ligne
2	Received Data : Réception des données
3	Transmitted Data : Données transmises
4	Data Terminal Ready : Données prêtes
5	Signal Ground : GND
6	Data set Ready : Envoi des données
7	Request to send : Demande de transmission
8	Clear to Send : Prêt pour la transmission
9	Ring Indicator : Indicateur de sonnerie

TABLE 28 – table :Broches RS-232

## Annexe 14 :

La HEI a un block I2C déjà implémenté. Ce dernier se représente sous la forme suivante :

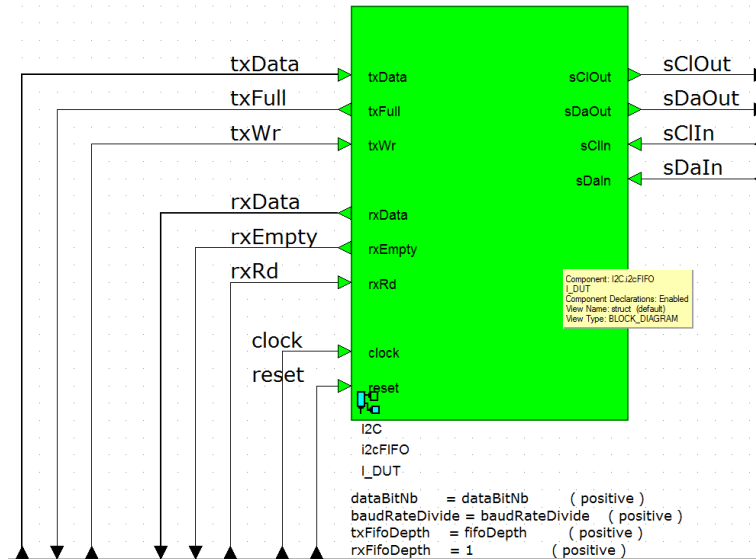


FIGURE 64 – Block I2C

Nous remarquons divers I/Os dans ce bloc. Nous avons :

- txData : Transmet les données à envoyer.
- txFull : Indique si la Fifo est pleine.
- txWr : Indique si nous sommes en lecture ou en écriture.
- rxData : Renvoie les données reçues par l'I2C.
- rxEmpty : Indique si la Fifo de réception est vide.
- rxRd : Indique que le contenu a été lu.
- sCIOut/sCIIn : Serial clock du côté de l'esclave.
- sDaOut/sDIn : Data du côté de l'esclave.

Le bloc fourni par la HEI pour la transmission de données par I2C doit être caractérisé par quatre éléments. Deux sont simplement des FIFO pour la transmission et la réception. Les deux autres sont le  $\text{baudRateDivide} = \text{Fréquence d'horloge} / \text{fréquence de l'I2C}$  et enfin le nombre de bits. Or ce nombre de bits est à prendre avec précaution. En effet, il ne s'agit pas du nombre de bits à transmettre mais du nombre de bits de données plus deux bits qui permettent un certain contrôle. La figure suivante explique en détail son fonctionnement.

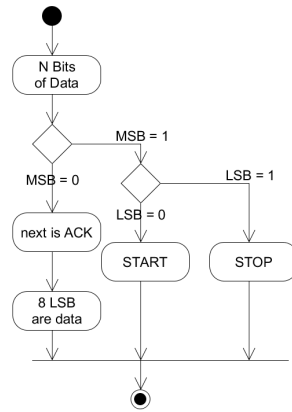


FIGURE 65 – Block I2C

Annexe 15 :

Nous avons ici différentes figures traitée avec le logiciel python et qui conduisent au résultat final.

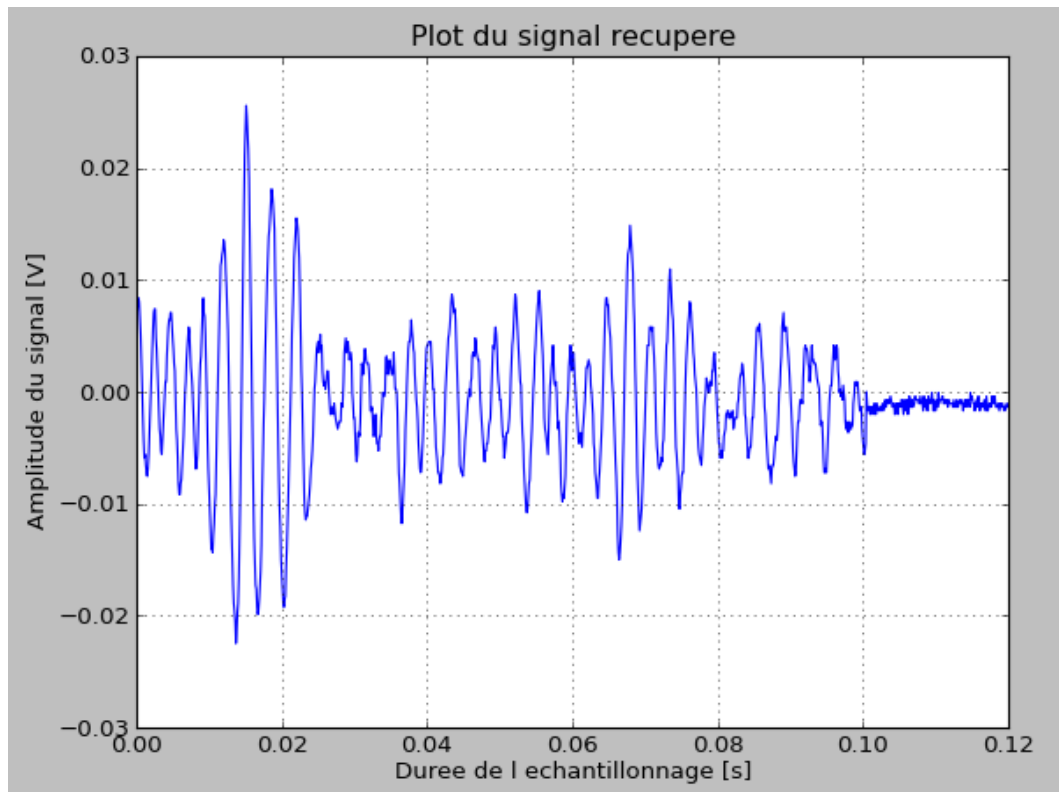


FIGURE 66 – Signal récupéré

La figure 66 nous montre une partie d'un signal qui sort du géophone sans qu'aucun traitement n'ait été effectué.

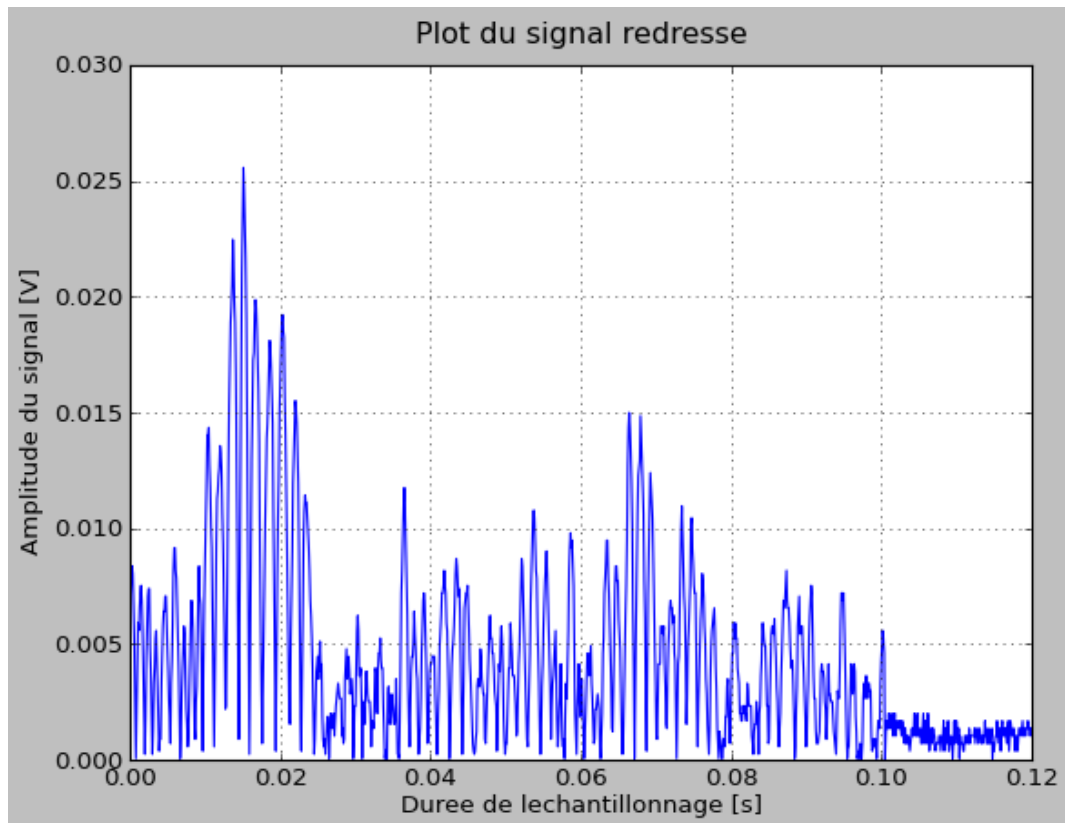


FIGURE 67 – Signal redressé

La figure 67 montre le signal simplement redressé. Il est à noter que la fréquence est doublée car le signal est redressé. Donc les parties négatives sont maintenant positives et le temps entre deux maximums positifs est divisé par deux.



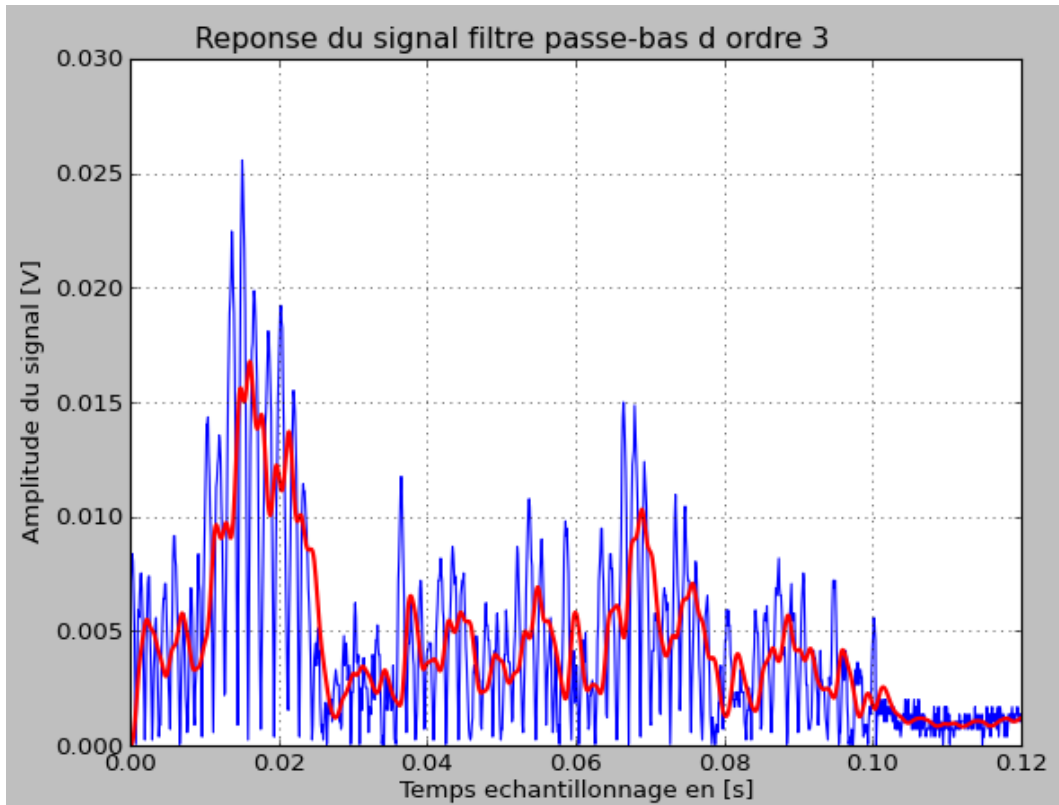


FIGURE 68 – Signal filtré passe-bas 3ème ordre à 100Hz

Sur la figure 68 nous visualisons le signal filtré à 100Hz. Le filtre utilisé est de troisième ordre, ce qui correspond au filtre analogique placé à l'entrée du convertisseur AD.

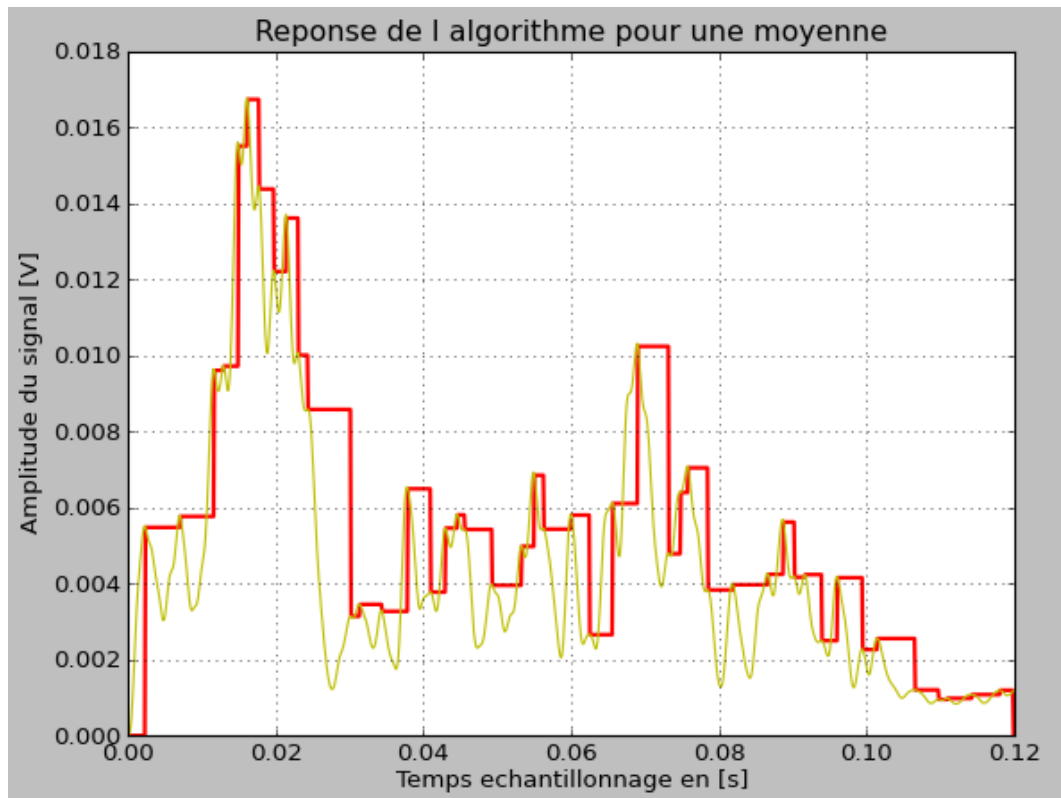


FIGURE 69 – Réponse de l'algorithme pour les maximums

Après que le signal soit traité avec l'algorithme précédemment expliqué, nous constatons sur la figure 69 que ce dernier suit bien la courbe du signal filtré. Le but est de limiter au maximum les valeurs à sélectionner et d'en sauver le minimum, tout en gardant une certaine cohérence et justesse.

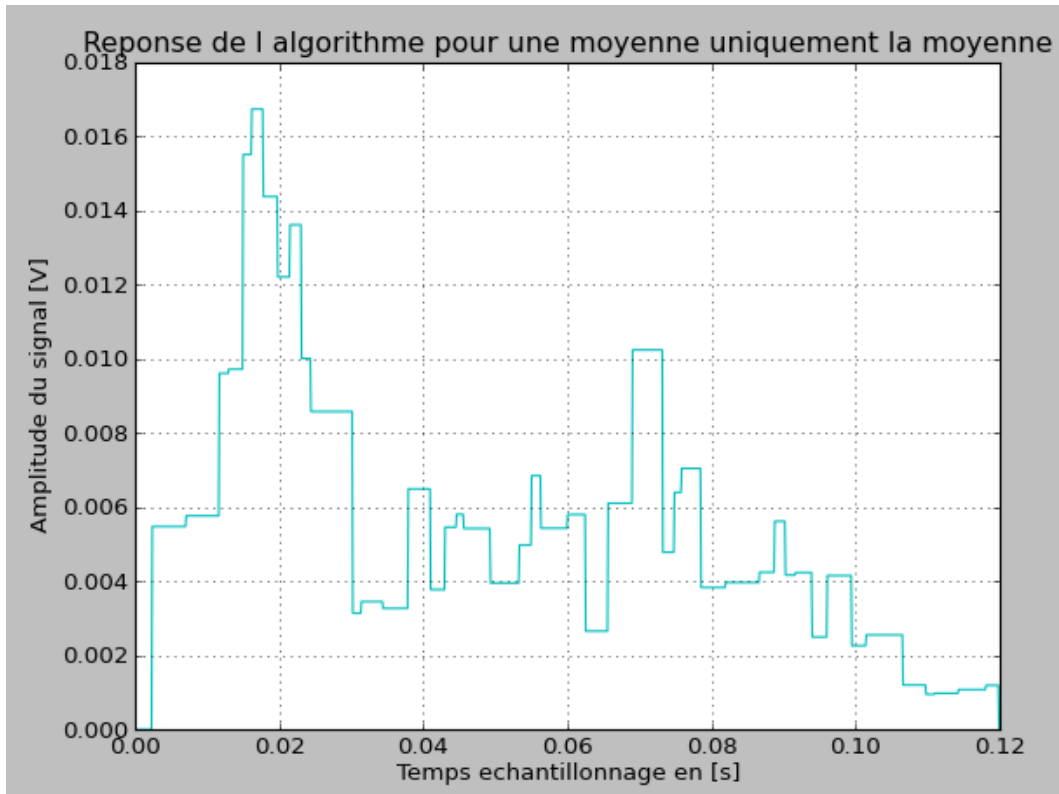


FIGURE 70 – Réponse de la moyenne uniquement

La figure 70 nous montre l'enveloppe du signal après traitement des valeurs maximales.

Nous avons également rajouté un traitement, nous filtrons à nouveau le signal comportant uniquement les valeurs maximales avec un filtre passe-bas. La fréquence de coupure de ce dernier est de 5Hz. Nous allons ensuite sous-échantillonner ce signal, c'est-à-dire prendre une valeur toutes les  $T$  valeurs. Nous désirons prendre une valeur sur 1000. Par le théorème de Shannon, nous devons échantillonner au minimum à deux fois la fréquence maximale du signal, ce qui correspond dans notre cas à 10Hz. Nous avons échantillonné à 10kHz et cette fois à 10Hz, nous avons donc un facteur 1000. C'est donc pour cette raison que nous n'allons prendre qu'une valeur sur 1000.

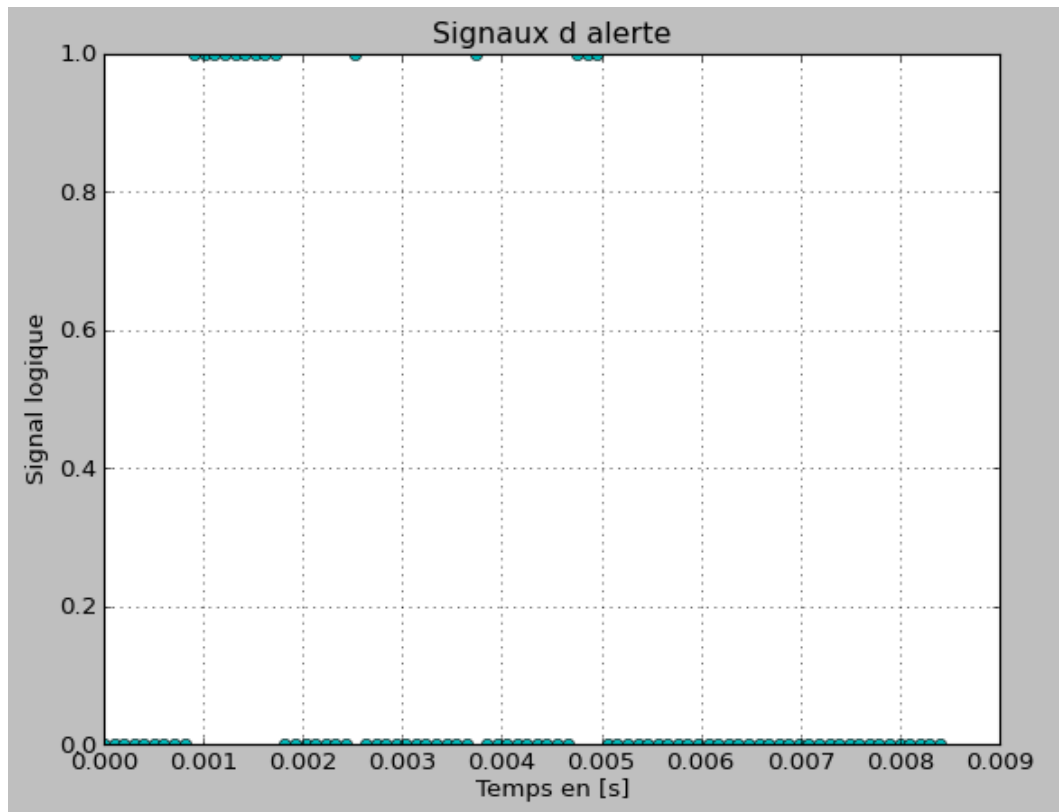


FIGURE 71 – Niveau logique de dépassement d'amplitude

Nous avons également fait un petit test sur les amplitudes, car au final ce qui est intéressant dans ce projet c'est d'avoir une alerte en cas de dépassement d'amplitude. Nous avons donc pris une valeur maximum d'amplitude et dès que cette valeur était dépassée, une alerte est émise. Ceci est représenté sur la figure 71.

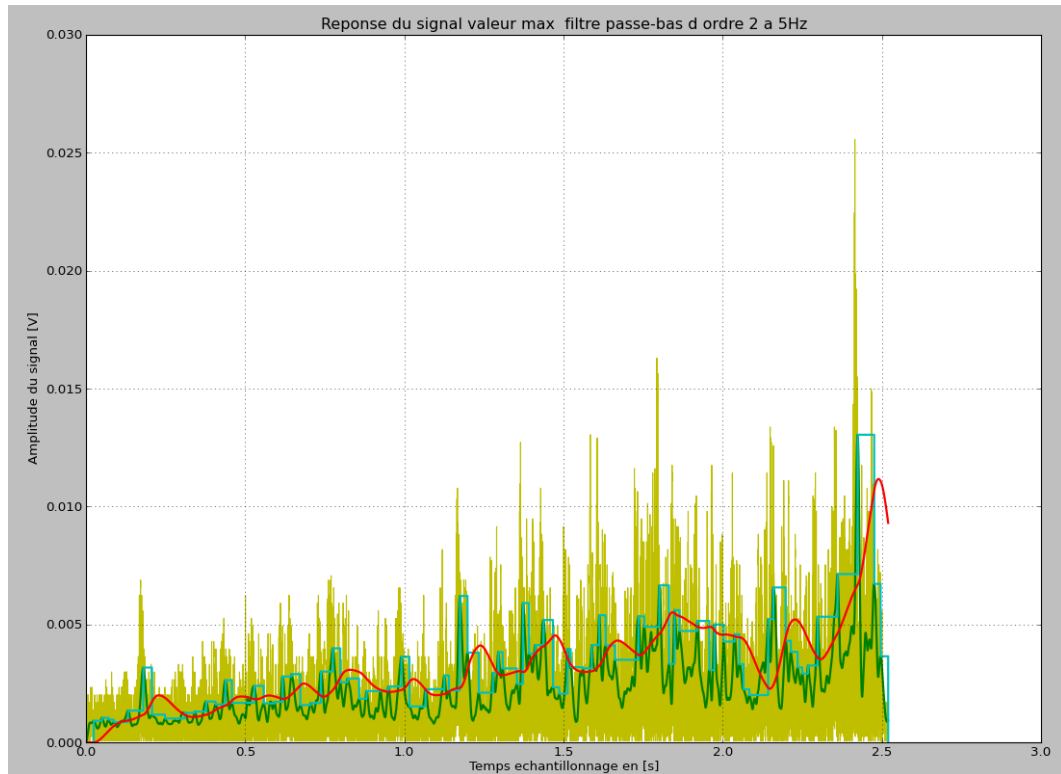


FIGURE 72 – Ensemble de courbes

Sur la figure 72, nous avons plusieurs courbes.

En jaune, nous avons le signal original.

En vert le signal qui est filtré passe-bas avec un filtre de 3ème ordre et une fréquence de coupure de 100Hz, ce qui correspond au filtre analogique qui se trouve à l'entrée du convertisseur.

En cyan, les valeurs maximales qui ont été prises grâce à l'algorithme mis en place.

Enfin en rouge, les valeurs maximales de l'algorithme qui ont été à leur tour filtrées passe-bas grâce à un filtre du second ordre avec une fréquence de 5Hz.

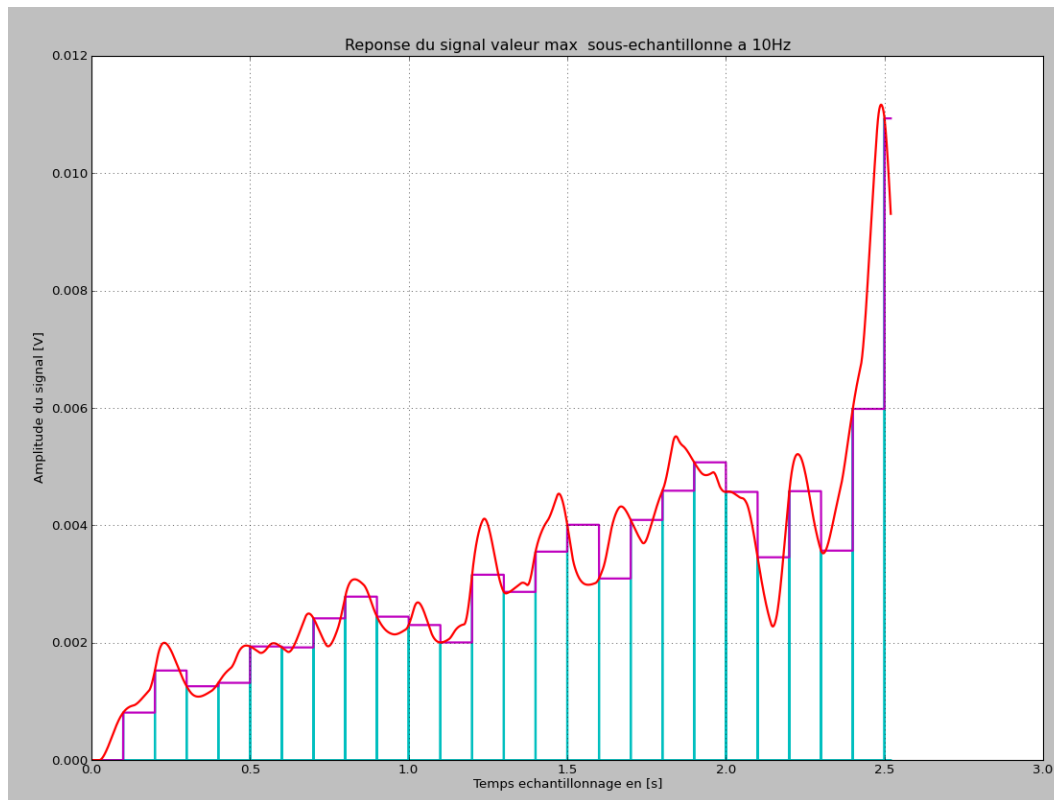


FIGURE 73 – Filtrage des valeurs maximales et sous-échantillonnage

La figure 73 montre en rouge le signal, comportant les valeurs maximales, filtré passe-bas. En cyan, les valeurs sous-échantillonnées. Et en magenta, ces mêmes valeurs mais tenues jusqu'à l'échantillonnage suivant.

## Annexe 16 :

Ainsi :  $C1 = 1$ ,  $L1 = 1$  et  $C2 = 1$ , donc  $C1=C2$ . Afin de trouver les valeurs de chacun des composants, nous devons les diviser par  $Df$ . Donc  $C1=C2=L1 = 1F/(100*\pi) = 3,18 * 10^{-3}$ . Puis pour chaque élément nous allons soit devoir le multiplier par l'impédance  $Z0$  dans le cas d'une inductance soit le diviser par  $Z0$  dans le cas d'une capacité.

Sachant que  $Z0$  serait de  $400[\Omega]$ , nous pouvons déduire les valeurs de  $C$  et de  $L$ . Pour les capacités, nous avons

$$C = \frac{3,18 * 10^{-3}}{400[\Omega]} = 7,95 uF$$

et pour l'inductance :  $L = 3,18 * 10^{-3} * 400[\Omega] = 1,272 H$ .

Or une inductance de cette taille est extrêmement élevée et coûte dans les CHF.20.- . Le seul moyen de pouvoir faire baisser ce valeurs est de jouer avec les fréquences et l'impédance  $Z0$ , or nous devrions prendre une impédance  $Z0$  très faible afin de pouvoir limiter l'inductance mais dans ce cas nous augmenterions les capacités. L'autre possibilité, est de varier la fréquence, or nous pouvons prendre une fréquence de coupure deux fois plus haute mais au final, les valeurs ne changeraient pas de manière significative. C'est pourquoi nous nous tournons vers un autre style de filtre, les filtres actifs.

Annexe 17 :

Pour le filtre du 1er ordre : Nous avons les équations suivantes :

$$RC = \frac{a1}{2 * \pi * f_{coupure}}$$

En posant  $C = 10\text{nF}$ ,  $a1=1$  selon les tables de Butterworth, nous obtenons :

$$R = \frac{a1}{2 * \pi * f_{coupure} * 10\text{nF}} = 159'154[\Omega] = 159[\text{k}\Omega].$$

Nous allons donc prendre une capacité de  $160[\text{k}\Omega]$ .

Pour le filtre du second ordre, les équations sont les suivantes :

$$\frac{C13}{C11} \geq \frac{4 * b1}{a1^2}.$$

Nous choisissons  $C11 = 10\text{nF}$ . Dans ce cas :

$$C13 \geq \frac{4 * b1 * C11}{a1^2} = 40\text{nF},$$

Or cette capacité n'existe pas, nous prenons donc  $47\text{nF}$ .

Nous devons ensuite calculer les résistances du schéma :

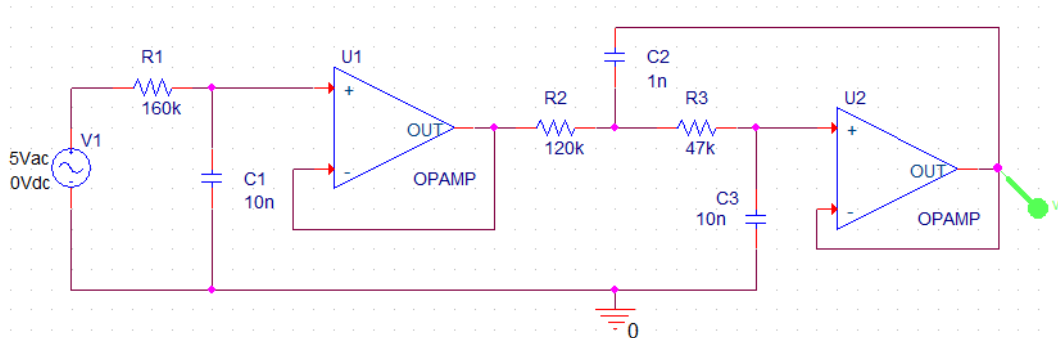


FIGURE 74 – Filtre passe-bas du 3ème ordre selon Sallen-Key

$$R14 = \frac{a1 * C13 + \sqrt{(a1 * C13)^2 - 4 * b1 * C11 * C13}}{4 * \pi * f_{coupure} * C11 * C13} = 79,579[\text{k}\Omega].$$

Cette résistance n'est pas non plus une résistance normalisée, nous prenons donc  $82[\text{k}\Omega]$ . Enfin, il s'agit de calculer la dernière résistance :

$$R12 = \frac{b1}{C13 * C11 * R14 * (2 * \pi * f_{coupure})^2} = 65,727[\text{k}\Omega].$$



Cette résistance n'est pas non plus normalisée, nous prenons donc une résistance de  $68[k\Omega]$ .

La figure suivante représente les tensions de sortie du filtre en fonction des tensions d'entrée. Les amplificateurs opérationnels choisis ne peuvent supporter en entrée que  $VCC+0,2V$ . Soit  $3,5V$  dans le cas présent.

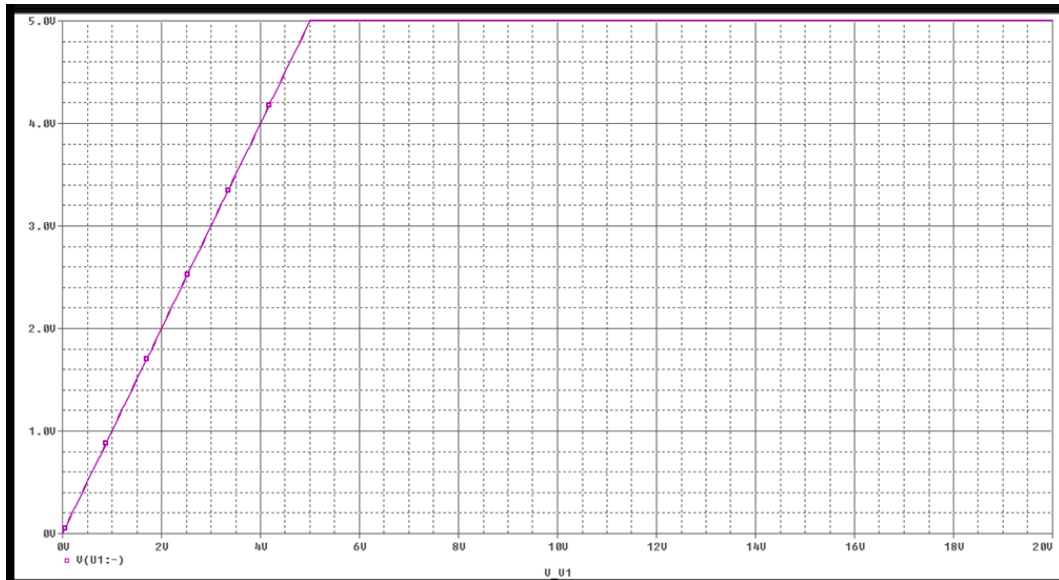


FIGURE 75 – Gamme de tensions pour le filtre passe-bas

La gamme de fréquences va de 0 à 32Hz environ, voici la réponse en fréquence du filtre dans les basses fréquences.

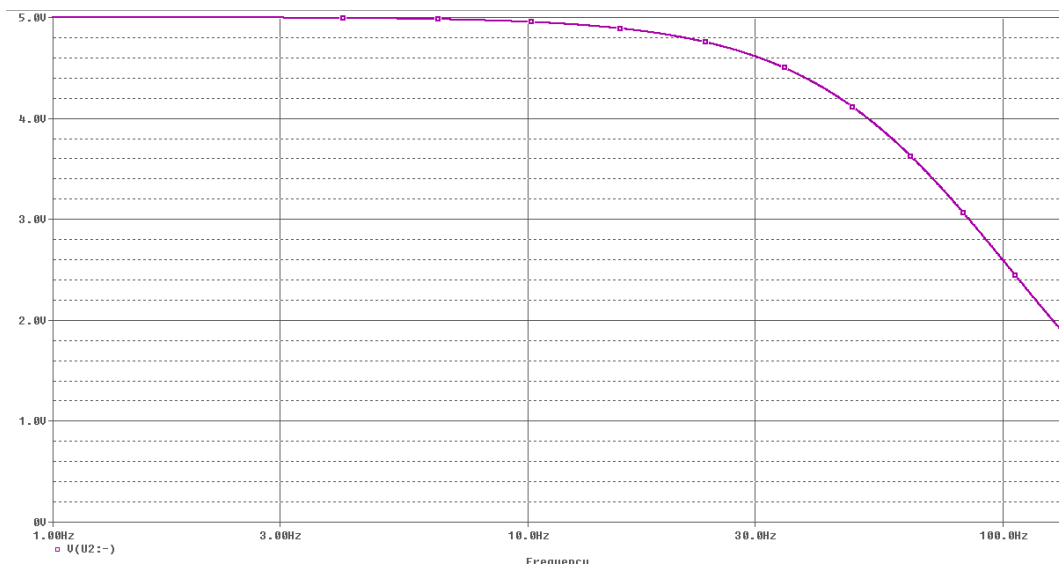


FIGURE 76 – Réponse en fréquence du filtre, dans les basses fréquences.

La figure suivante nous montre la réponse du filtre lorsque la fréquence augmente jusqu'à

1kHz.

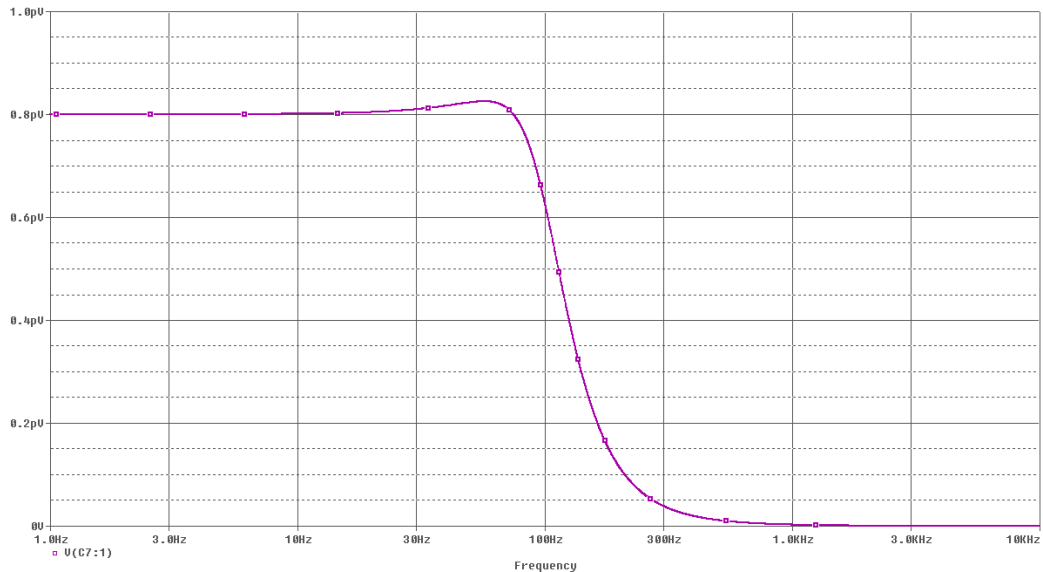


FIGURE 77 – Réponse en fréquence du filtre sur une grande plage de fréquence.

Afin de pouvoir réaliser ce filtre, l'amplificateur opérationnel suivant a été sélectionné : **TSV621ILT**. Il a les caractéristiques suivantes :

- Tension alimentation : 1.5 à 5.5V.
- Bande passante : 420kHz
- Température de fonctionnement : -40 à +85 degrés.
- Vitesse de balayage, Slew Rate : 0.14V/us.
- Disponible à la HEI.

Cet amplificateur opérationnel n'accepte en entrée que des tensions comprises entre  $-V_{CC}-0.2V$  et  $V_{CC}+0.2V$ . Sachant que le géophone peut fournir des tensions de 0 à 20V, il a donc fallu le faire précéder par un diviseur de tension. Le diviseur de tension a été calculé de la façon suivante :

$$U_{out} = \frac{R2 \cdot U1}{R1 + R2}$$

$$5V = \frac{470 \cdot 20}{470 + R1}, R2 = 470[\Omega]. (Choisie).$$

Nous trouvons donc  $R1 = 1,41[k\Omega]$ , soit en prenant des résistances normalisées,  $1,5[k\Omega]$ . Nous multiplions donc la tension d'entrée par :  $\frac{470}{470+1500} = 0.2385$ . Ainsi  $U_{out} = 0.2385 \cdot U_{in}$ .

Nous alimentons ces amplificateurs opérationnels avec du 3,3V. Ainsi en sortie nous aurons du 3,3V et donc nous n'avons pas besoin de diviser encore la tension. De plus, nous plaçons un filtre passe-bas à la sortie du convertisseur DC-DC.

Des essais ont été faits avec matlab afin de concevoir le filtre en question. Il reste identique au niveau des ordres par rapport à celui calculé précédemment mais les valeurs des

capacités et des résistances se verraient changées.

Les valeurs pour le filtre du premier ordre seraient :

- $C = 10\text{nF}$
- $R = 160[k\Omega]$

Les valeurs pour le filtre du second ordre seraient :

- $C11 = 10\text{nF}$
- $C13 = 47\text{nF}$
- $R14 = 82[k\Omega]$
- $R15 = 68[k\Omega]$

Il faut respecter le fait que nous devons être en-dessous de 100dB à la moitié de la fréquence de Nyquist. Selon le schéma suivant, qui est tiré de matlab, nous obtenons bien un nombre inférieur à -100dB à la moitié de la fréquence d'échantillonnage.

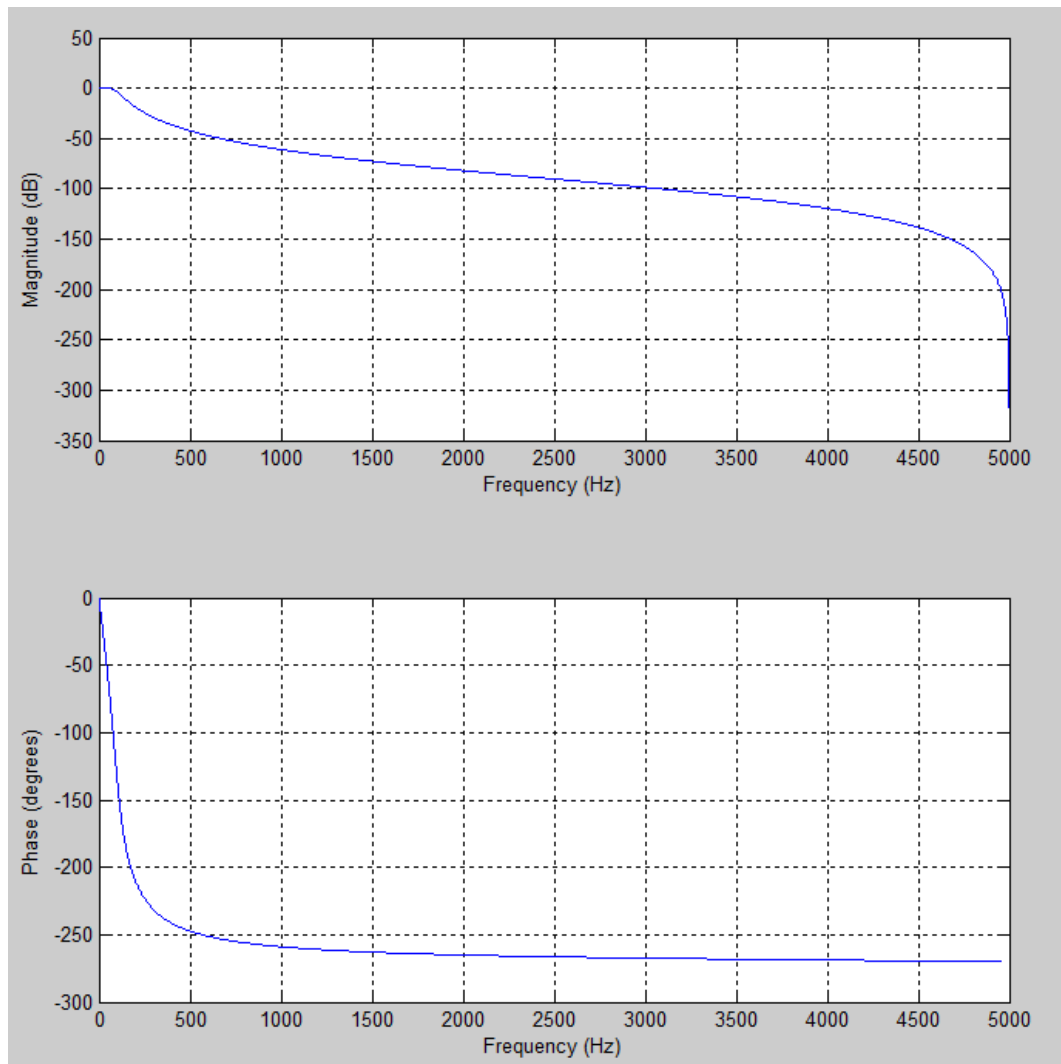


FIGURE 78 – Réponse en fréquence du filtre passe-bas de 3ème ordre avec une fréquence de coupure de 100Hz.

Nous observons qu'à partir de 3kHz, l'amplitude est inférieure à -100dB et donc que le filtre coupe bien ce que nous désirons.

Annexe 18 :

On a donc  $R = (V_{\text{alimentation}} - V_{\text{zener}}) / (1.5 * I_{\text{max}})$

1.5 est un facteur de sécurité afin que la diode soit toujours passante et qu'on ne soit pas dans un cas limite. Nous prendrons  $I_{\text{max}} = 5\text{mA}$ , ainsi  $R = (20\text{V} - 4\text{V}) / (1.5 * 5 * 10^{-3}) = 2133[\Omega]$ . Nous avons ici la résistance à placer afin d'avoir un courant maximum de 5mA.

On trouve donc  $R = 2400[\Omega]$ .

Le courant maximum est donc  $I_{\text{max}} = (V_{\text{alimentation}} - V_{\text{zener}}) / R / 1.5 = 4.44\text{mA}$  et on pose ceci =  $I_r$ .

La puissance dans la résistance est donc  $P_r = R * I^2$

Calcul de la puissance dans la diode Zener : On a  $I_{\text{zener}} = I - I_r$ , donc  $P_{\text{zener}} = V_{\text{zener}} * I_{\text{zener}} = V_{\text{zener}} * (I - I_r) = 4 * (5\text{mA} - 4.4\text{mA}) = 2.2 * 10^{-3}\text{W}$ , ce qui correspond à 2.2mW.

Annexe 19 :

Tensions de sortie maximales et minimales pour un géophone.

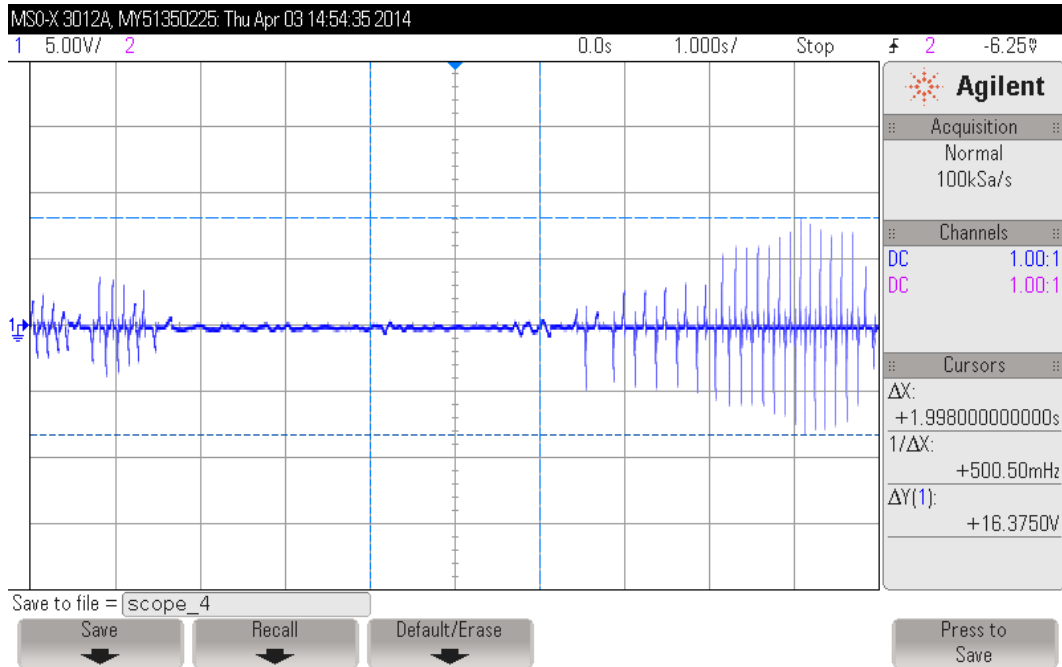


FIGURE 79 – Tension maximale de sortie du géophone Dynamic Technologies DT20DX 4.5Hz

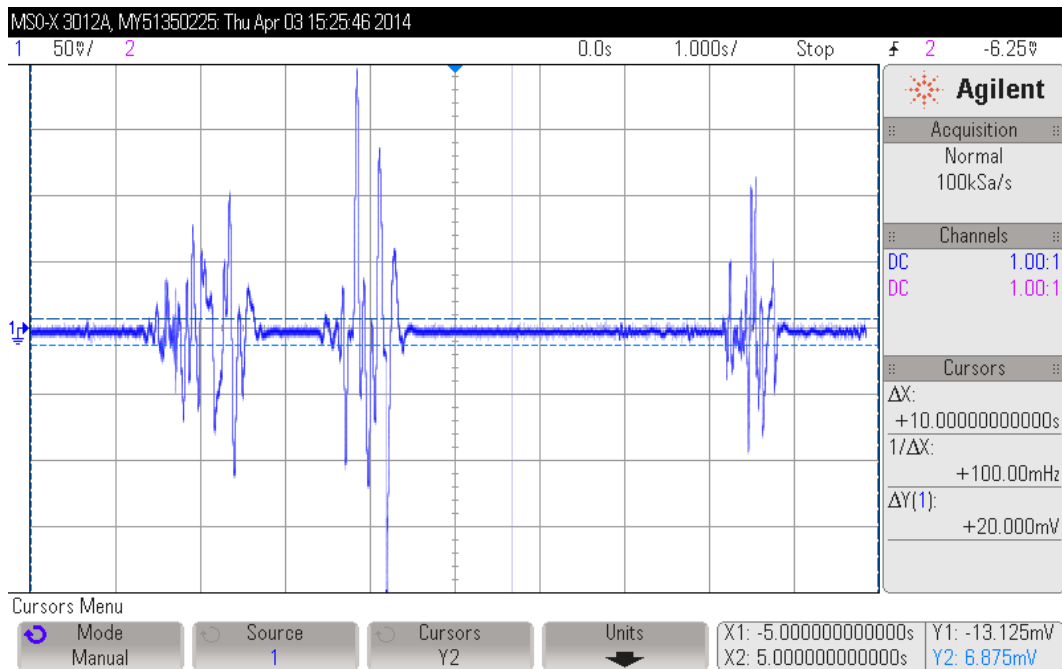


FIGURE 80 – Tension minimale de sortie du géophone Dynamic Technologies DT20DX 4.5Hz

Annexe 20 :

Broches du convertisseur AD :

Le convertisseur AD a de nombreuses broches. Nous avons une partie qui est alimentée grâce à une alimentation digitale et une autre partie avec une analogique.

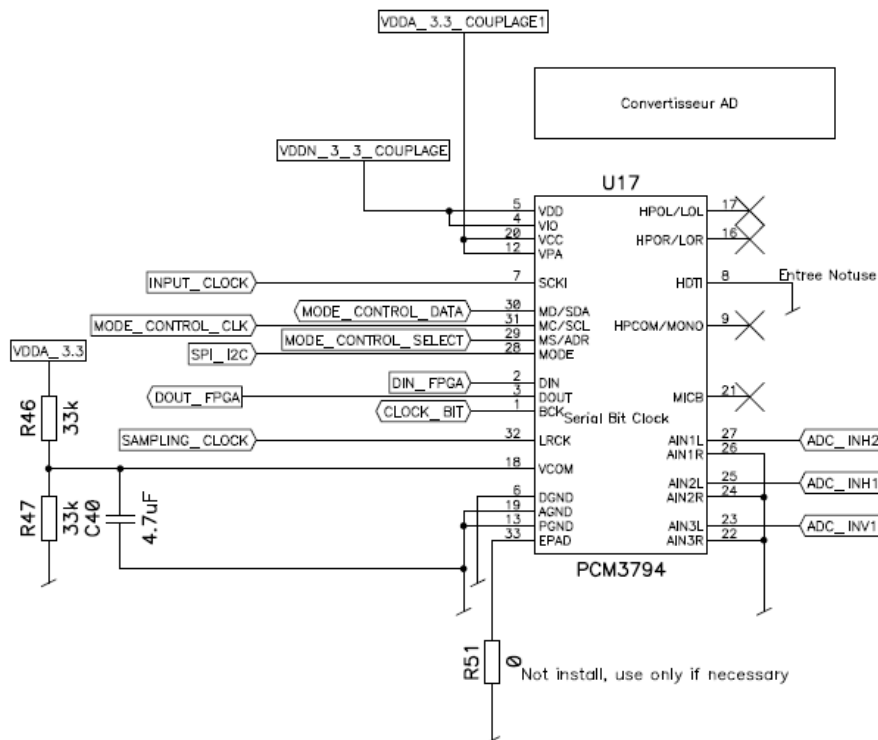


FIGURE 81 – Schéma du convertisseur AD : PCM3794

Il faut tout d'abord connecter la pin Vcom(Analog common Voltage) à la moitié de VCC(Analog power Supply) et ajouter une capacité de 4.7uF entre la broche Vcom et AGND (Analog Ground).

Il y a un signal power on Reset à VDD(Digital Power Supply) = 1.2V. Le système Reset est activé si le SRST(System Reset)=1 du registre 85. Après la séquence de Reset, SRST est mis à 0 automatiquement.

Dout(Serial Data Out) transmet les données qui viennent de l'ADC. BCK(Serial Bit Clock) s'occupe de réguler le transfert des données sur Dout lors de la transition de l'état haut à l'état bas. De plus BCK et LRCK(Left and Right Channel Clock) doivent être synchronisés avec la fréquence du système audio. Idéalement il faudrait qu'ils en dérivent. Dans le cas où le Convertisseur AD est master, il génère lui-même le BCK et LRCK.

MC(Master clock)/SCL représente le clock pour le SPI ou pour l'I2C. MD(Master Data)/SDA est la broche des données pour le SPI ou pour l'I2C. MS/ADR C'est le Master-Select ou l'adresse lors de l'utilisation du protocole I2C. La broche MODE nous permet

de sélectionner soit le protocole SPI en mettant la ligne à l'état bas, ou I2C en mettant la ligne à l'état haut. SCKI(System Clock) est la fréquence d'horloge du système.

Broches spécifiques		
<i>Fonction</i>	<i>SPI</i>	<i>I2C</i>
Clock	MC	SCL
Datas	MD	SDA
Address	MS	ADR
Mode	Actif à l'état bas	Actif à l'état haut

TABLE 29 – Quelques broches plus spécifiques

Tous les bits de contrôle et de datas commencent avec le MSB.



Annexe 21 :

Réponse en fréquence du filtre passe-bas actif.

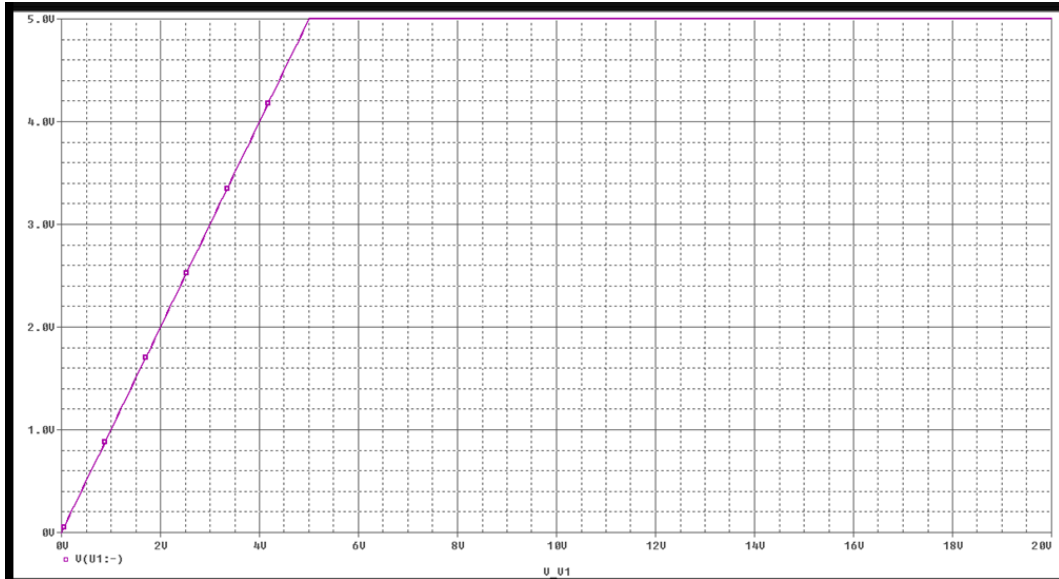


FIGURE 82 – Gamme de tension pour le filtre passe-bas

La gamme de fréquence va de 0 à 32Hz environ, voici la réponse en fréquence du filtre dans les basses fréquences.

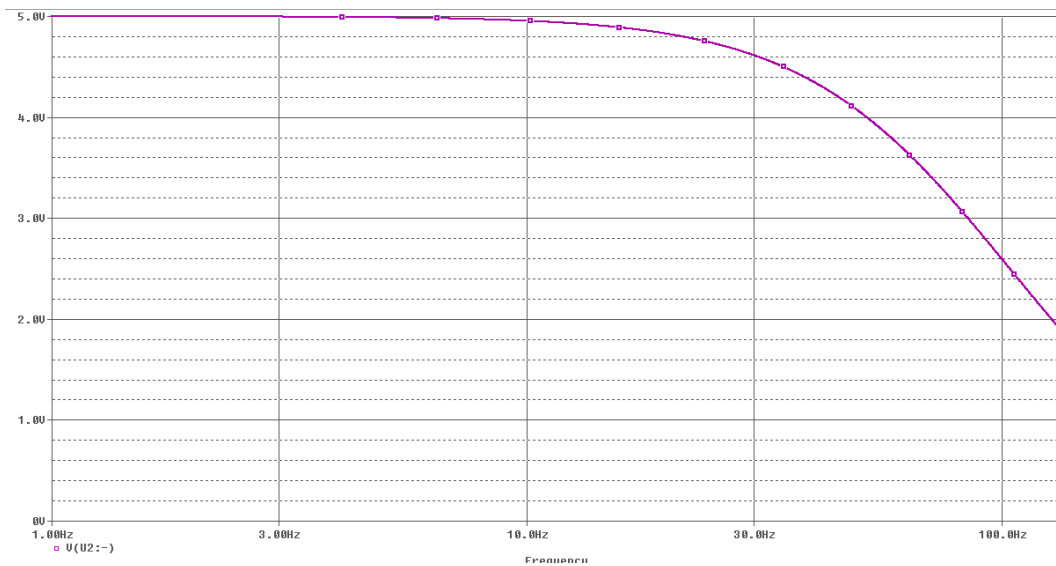


FIGURE 83 – Réponse en fréquence du filtre, dans les basses fréquences.

La figure suivante nous montre la réponse du filtre lorsque la fréquence augmente jusqu'à 1kHz.

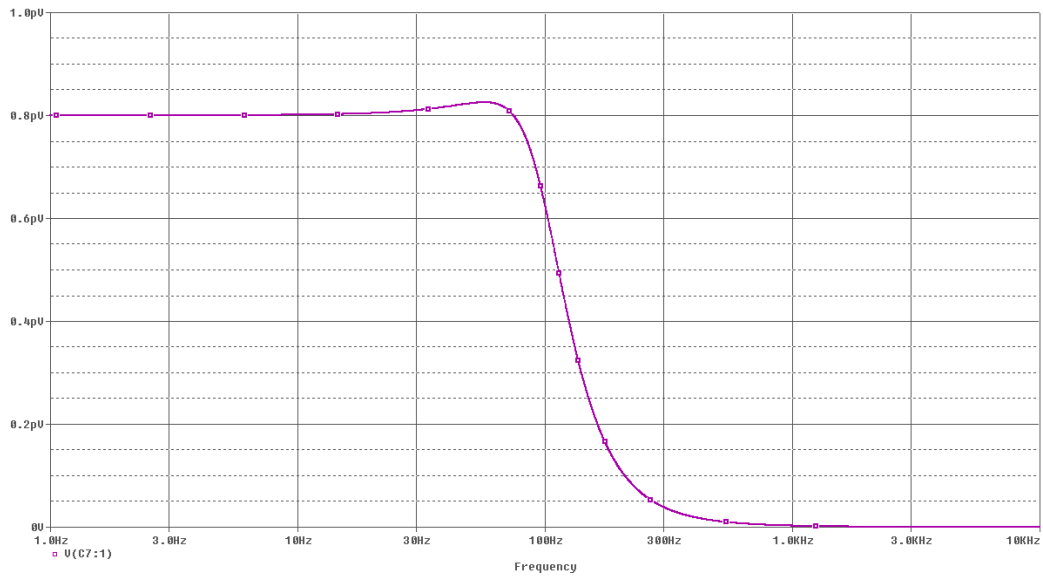
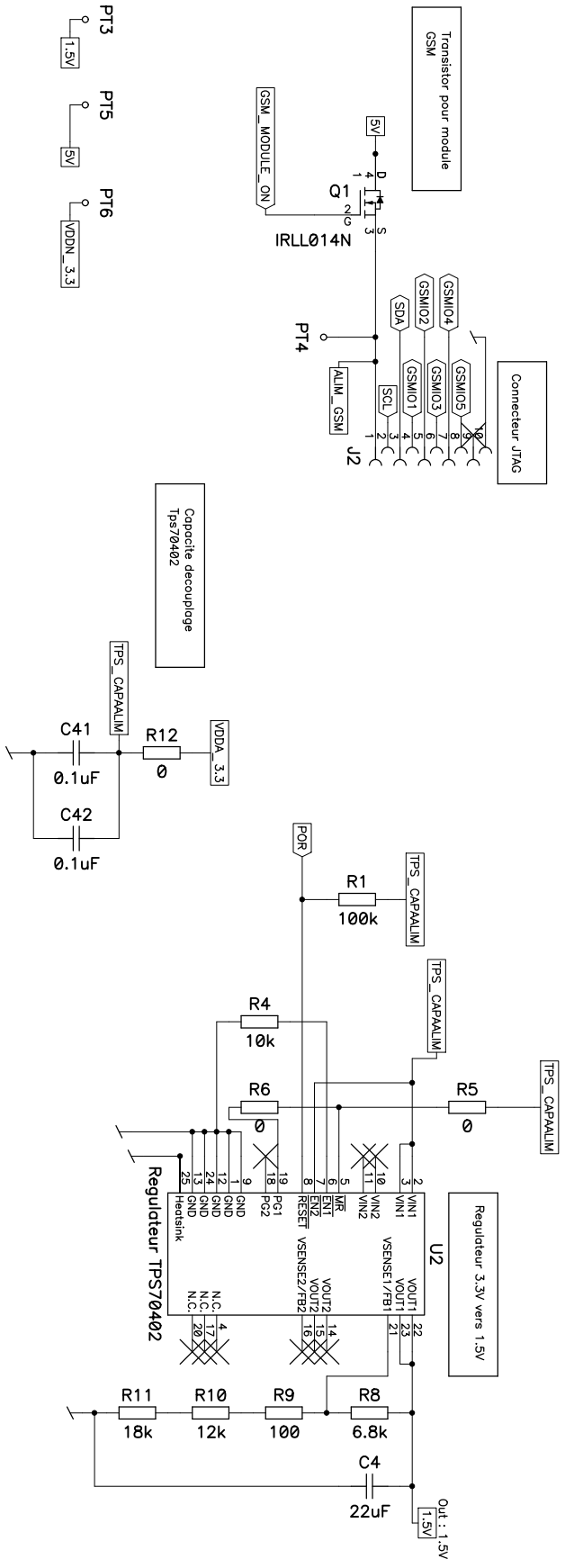
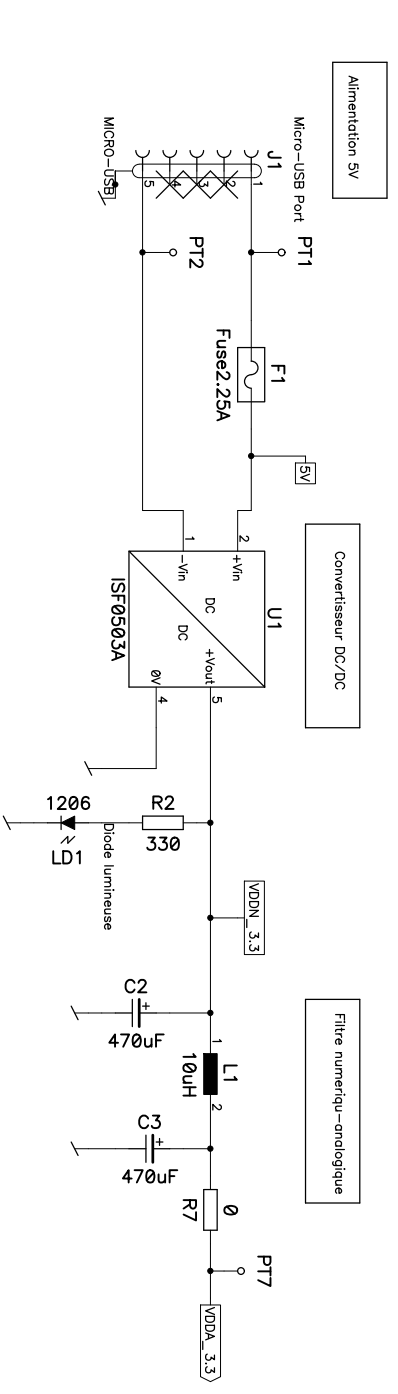
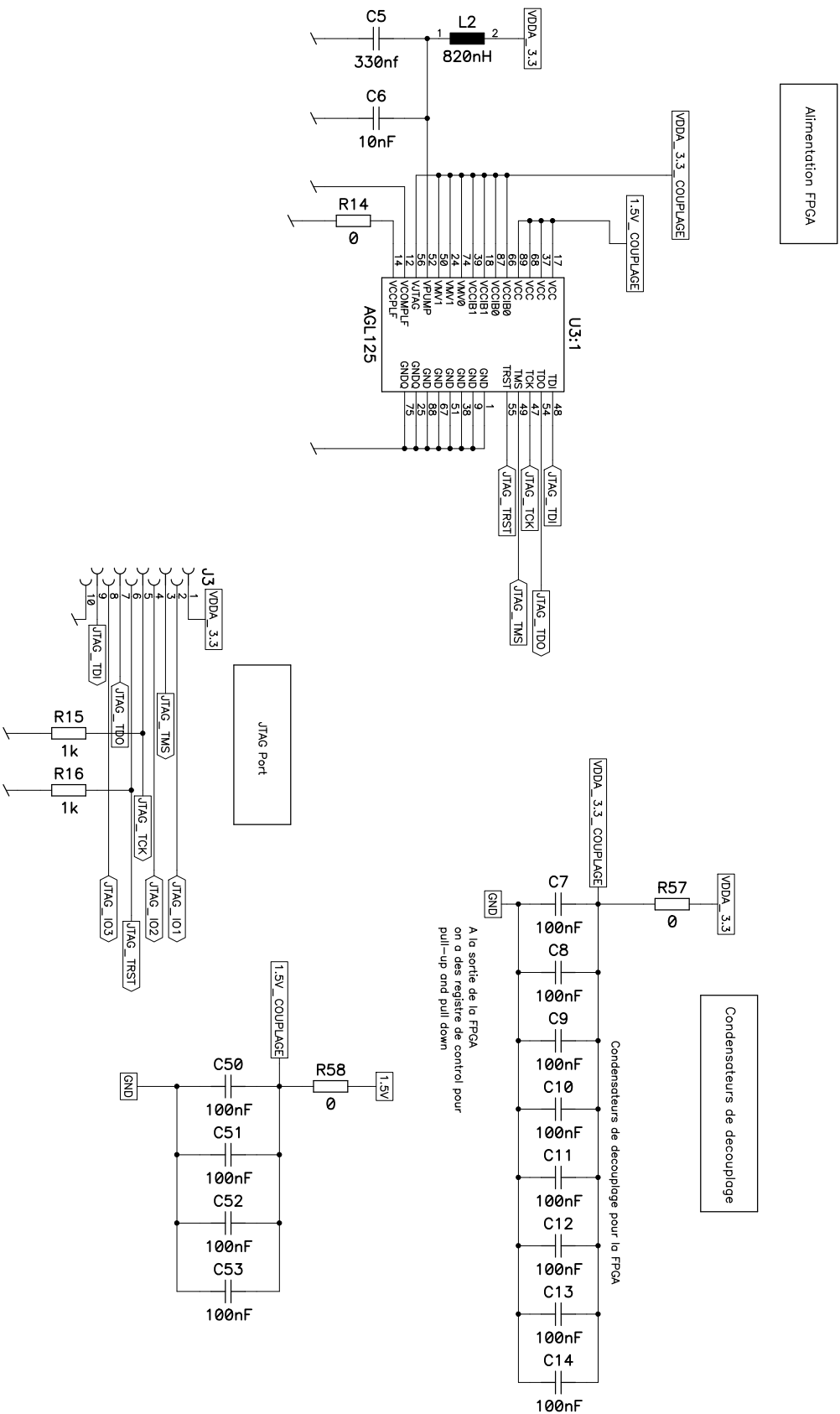


FIGURE 84 – Réponse en fréquence du filtre sur une grande plage de fréquence.



<p>Titre</p> <p>Titre 2</p> <p>HAUTE ECOLE VALAISANNE</p>		DES	{Date}	{Drawn By}
		REV	V1.0	
<p>1/10</p> <p>Monitoring_v1_1.sch</p>				



Alimentation FPGA

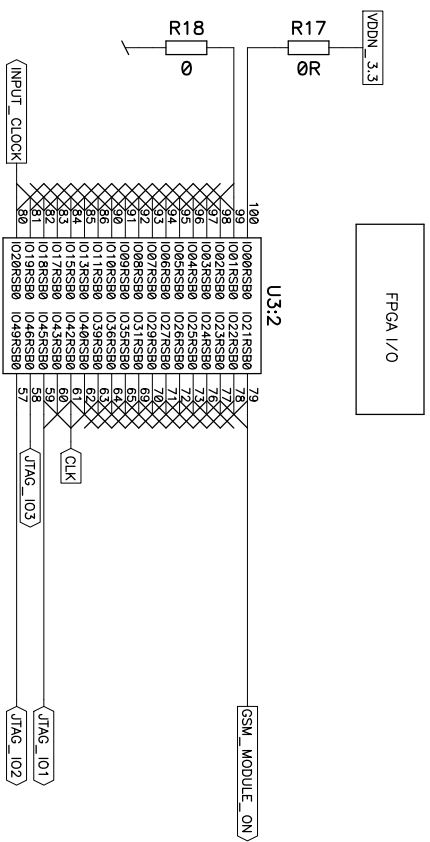
Condensateurs de decouplage

Condensateurs de decouplage pour la FPGA

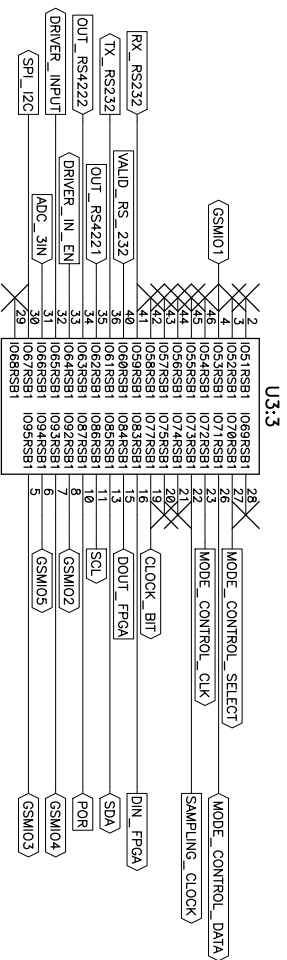
A la sortie de la FPGA  
on a des registre de control pour  
pull-up and pull down

JTAG Port

<pre> {Title} {Title 2}         </pre>		<pre> DES {Date} REV V1.0         </pre>		<pre> FPGA_Alim         </pre>		<pre> {Path} Monitoring_v1_1.sch         </pre>	
<pre> HAUTE ECOLE VALAISANNE         </pre>							



AGL 125

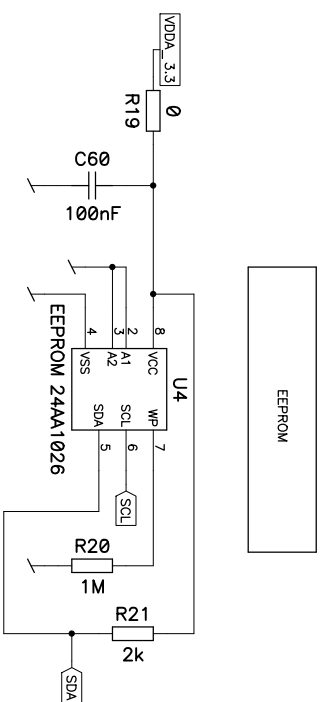


AGL 125

HAUTE ECOLE VALAISANNE

FPGA\_10

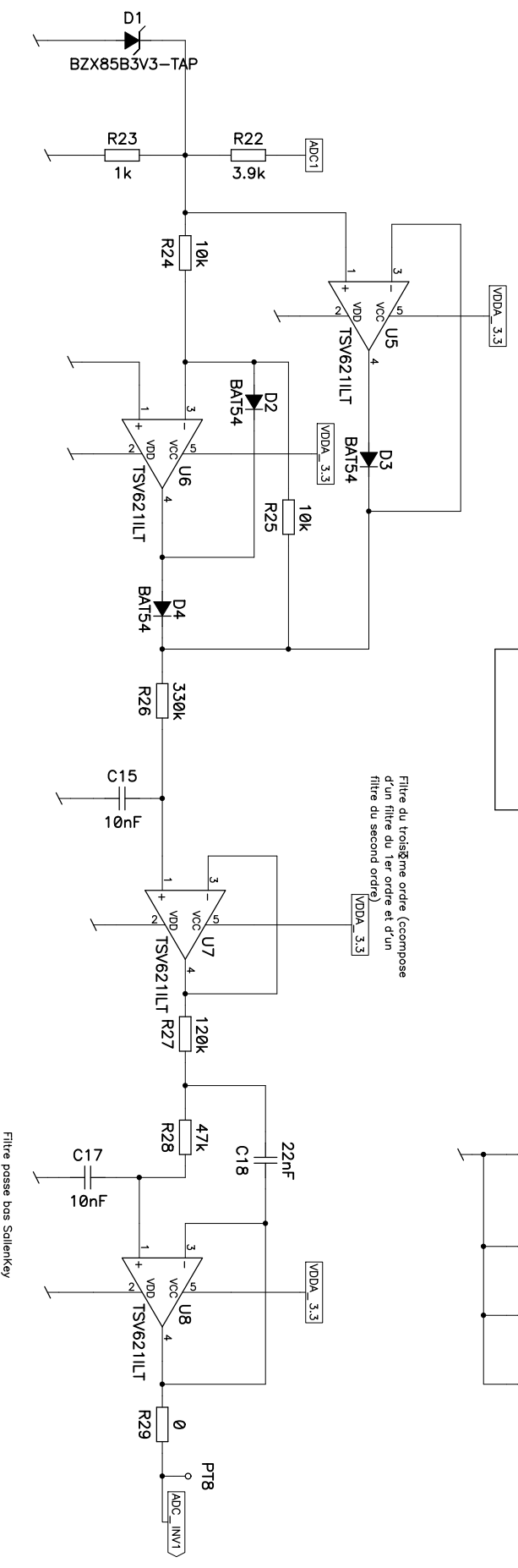
DES	{Date}	{Drawn By}
REV	V1.0	
3/10	{Path}	Monitoring_v1_1.sch



EEPROM

{Title} {Title 2}		EEPROM
HAUTE ECOLE VALAISANNE		

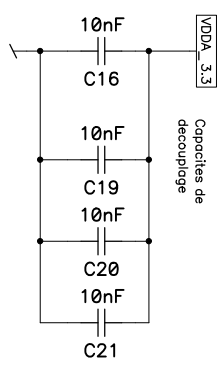
DES	{Date}	{Drawn By}
REV	V1.0	
4/10	{Path} Monitoring_v1_1.sch	



Filtere

Filtere du troisieme ordre (compose  
d'un filtre du 1er ordre et d'un  
filtre du second ordre)

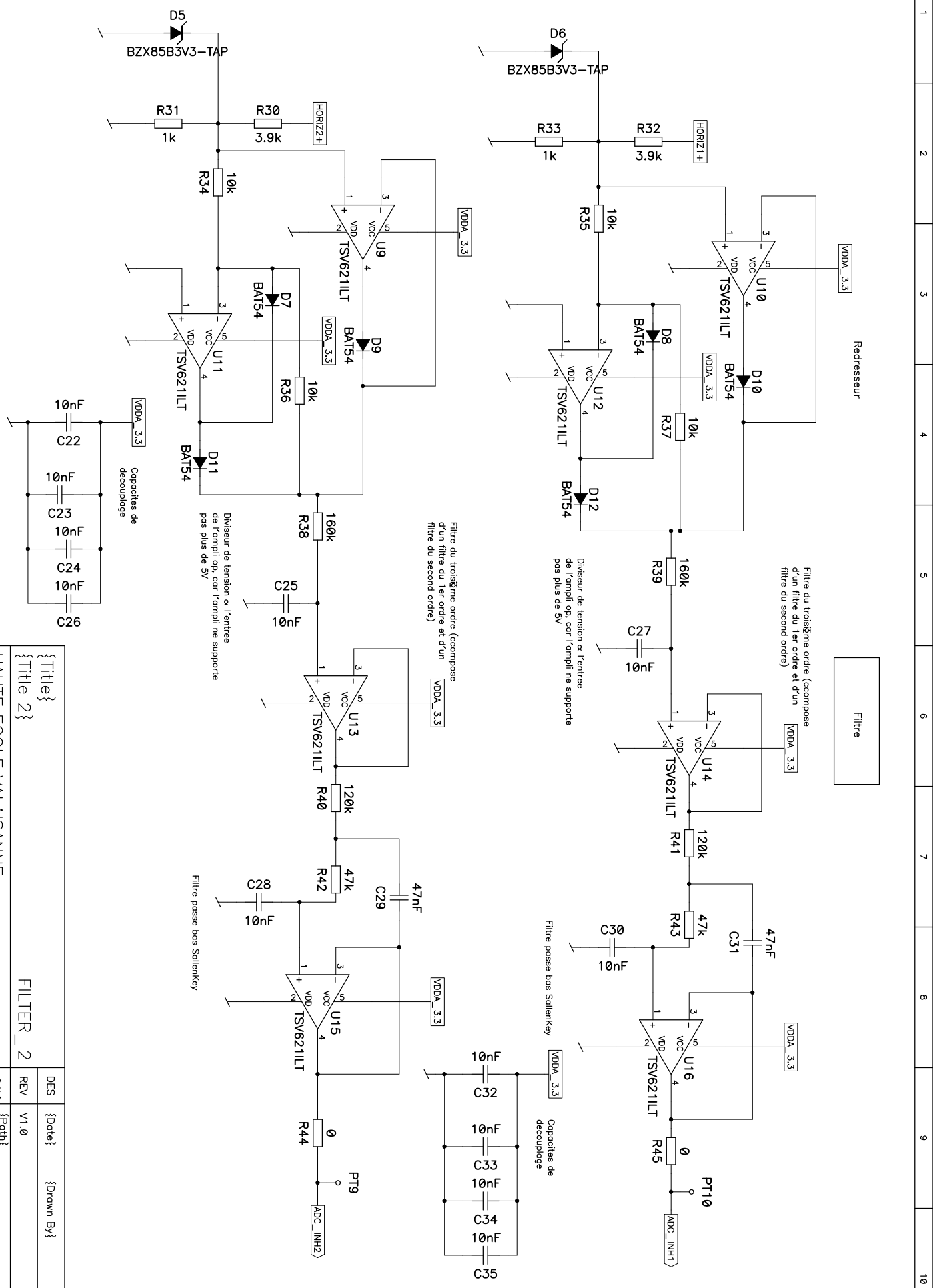
Filtere passe bas Sallenkey



Capacites de  
decouplage

Diviseur de tension α l'entree  
de l'ampli op, car l'ampli ne supporte  
pas plus de 5V

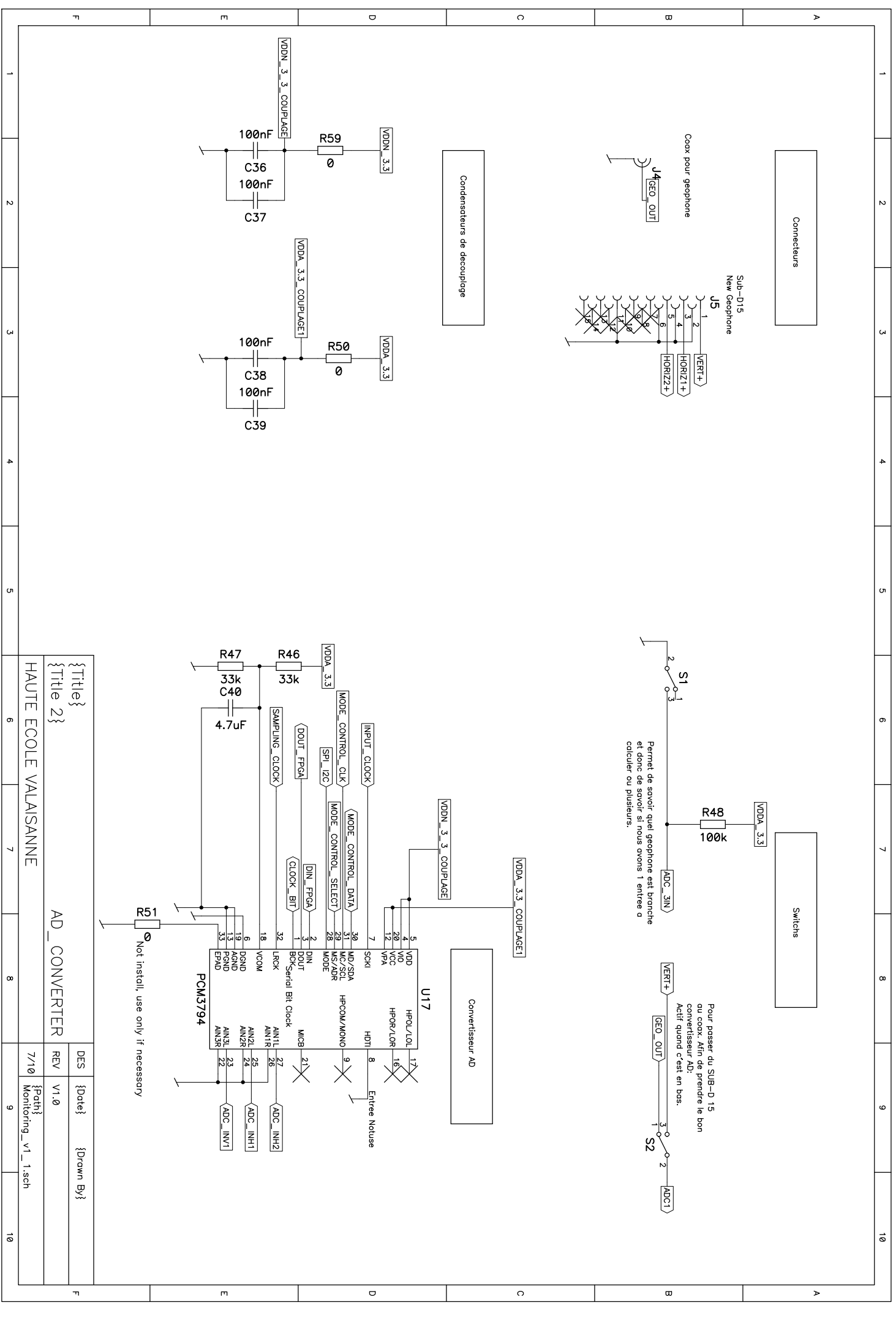
<p>HAUTE ECOLE VALAISANNE</p>		<p>DES {Date} Drawn By}</p>	
<p>HAUTE ECOLE VALAISANNE</p>		<p>REV V1.0</p>	
<p>HAUTE ECOLE VALAISANNE</p>		<p>5/10 {Path} Monitoring_v1_1.sch</p>	
<p>HAUTE ECOLE VALAISANNE</p>		<p>5/10</p>	



{Title} {Title 2}		DES	{Date}	{Drawn By}
HAUTE ECOLE VALAISANNE		REV	V1.0	
		6/10	{Path}	
			Monitoring_v1_1.sch	

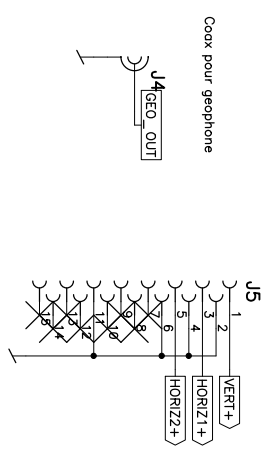
FILTER\_2





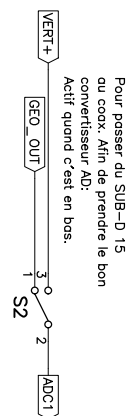
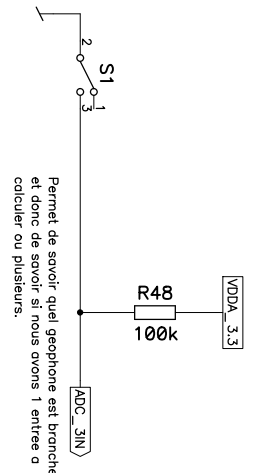
Connecteurs

Sub-D15  
New Geophone

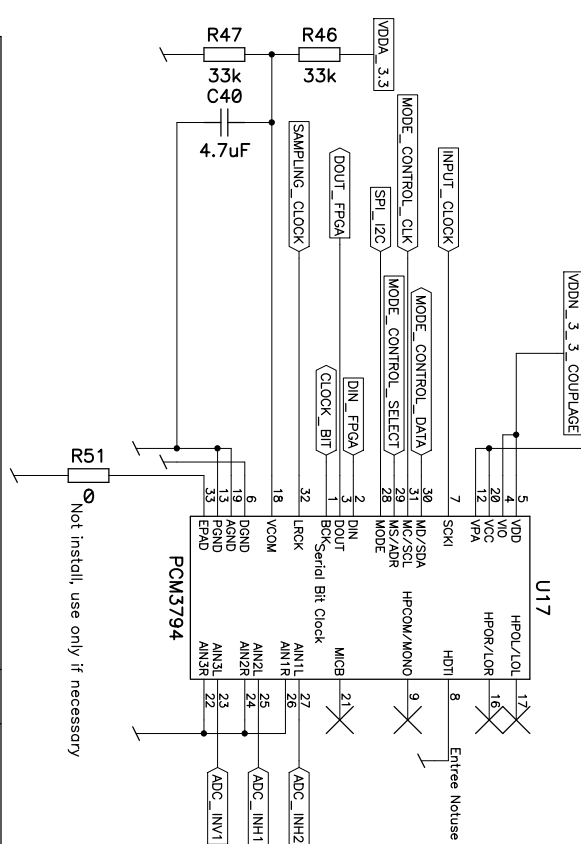


Condensateurs de decouplage

Switchs

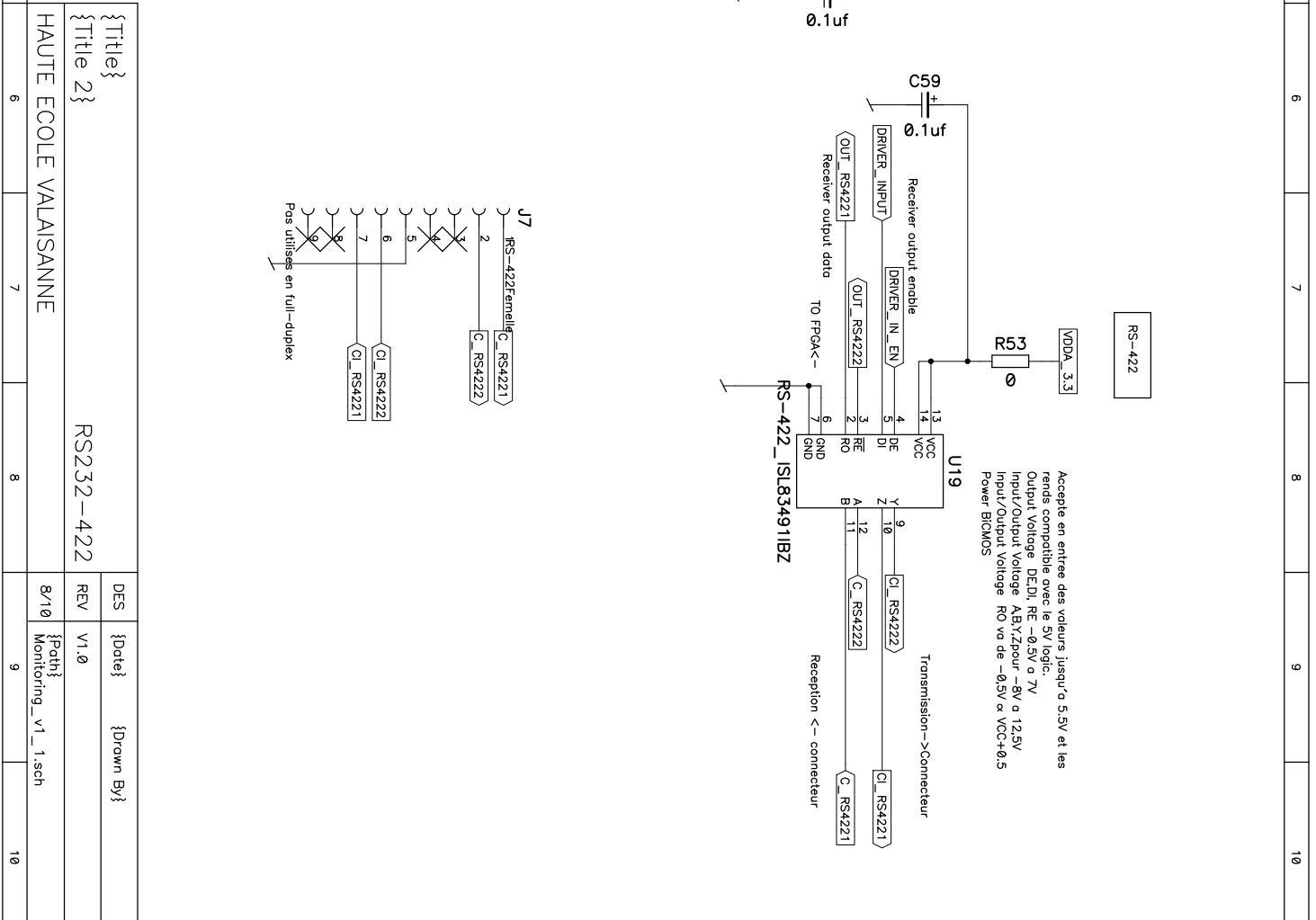
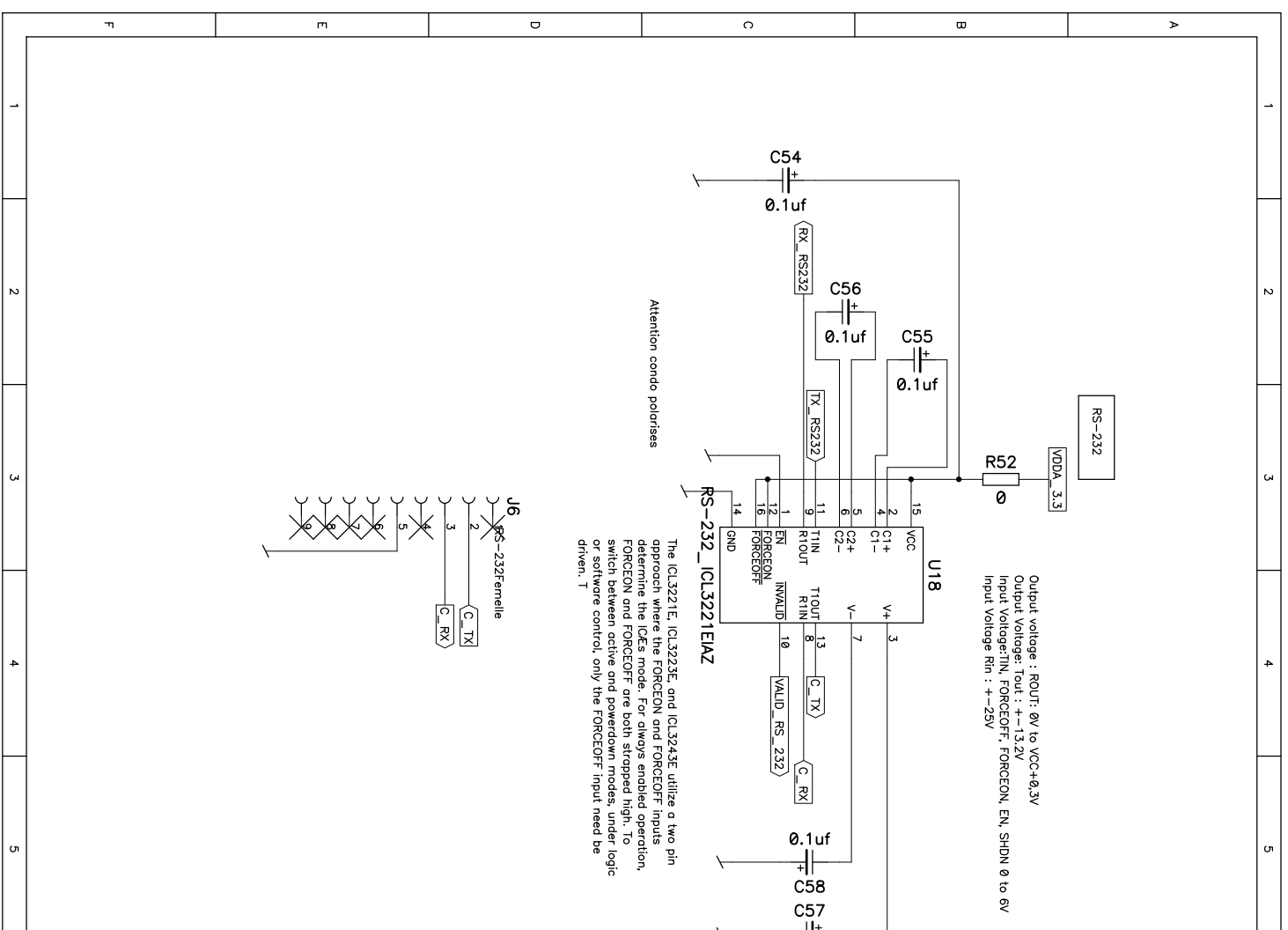


Convertisseur AD

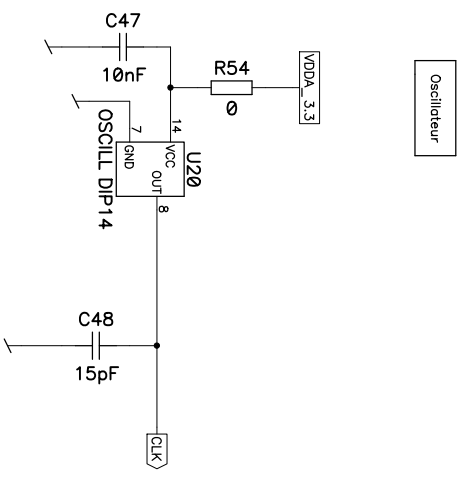


Not install, use only if necessary

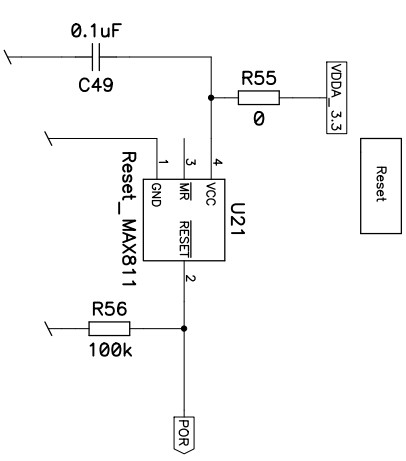
<p>{Title}</p> <p>{Title 2}</p> <p>HAUTE ECOLE VALAISANNE</p>		<p>AD_CONVERTIER</p>	
DES	{Date}	{Drawn By}	
REV	V1.0		
7/10	{Path}	Monitoring_v1_1.sch	



<p>HAUTE ECOLE VALAISANNE</p>		<p>DES {Date}</p>	<p>REV V1.0</p>	<p>8/10</p>	<p>{Path}</p>
<p>HAUTE ECOLE VALAISANNE</p>		<p>DES {Date}</p>	<p>REV V1.0</p>	<p>8/10</p>	<p>{Path}</p>
<p>HAUTE ECOLE VALAISANNE</p>		<p>DES {Date}</p>	<p>REV V1.0</p>	<p>8/10</p>	<p>{Path}</p>



Oscillateur

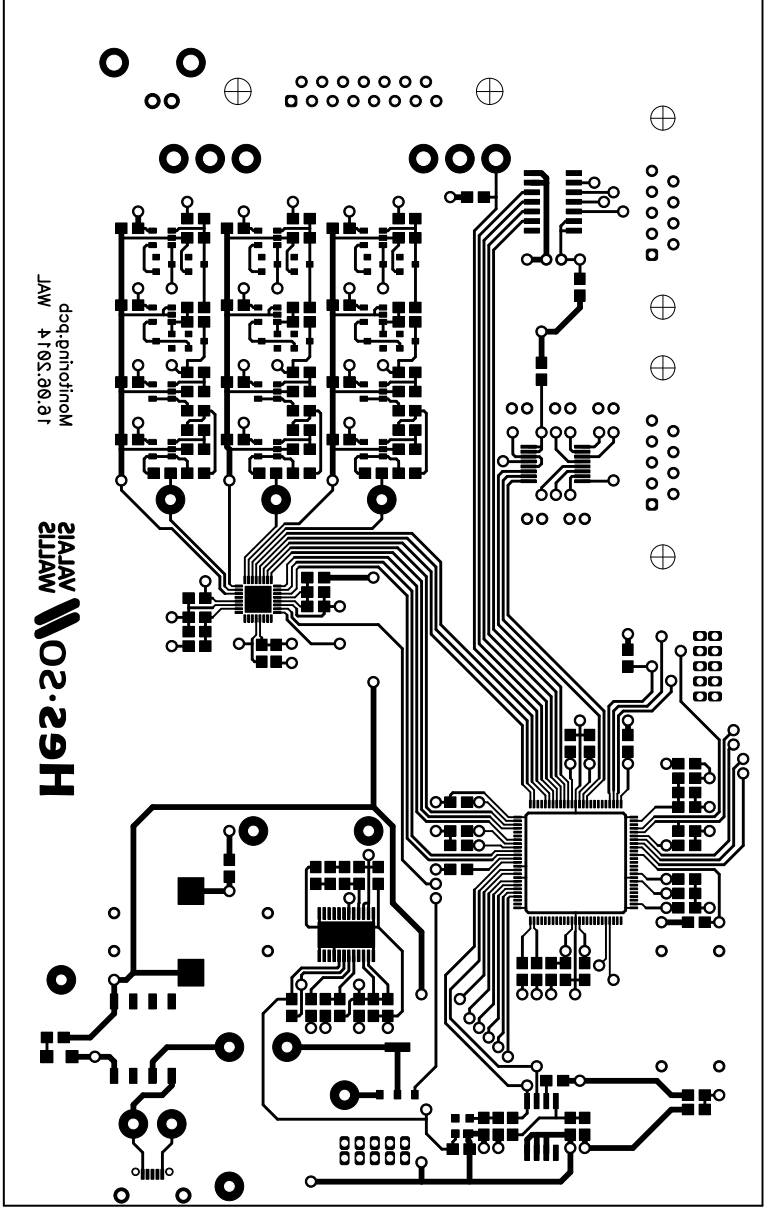


Reset

<p>HAUTE ECOLE VALAISANNE</p>		<p>OSCILLATORS</p>	
DES	{Date}	REV	{Drawn By}
REV	V1.0	9/10	{Path} Monitoring_v1_1.sch

1	2	3	4	5	6	7	8	9	10
A	B	C	D	E	F				





Drill Table	Quantity	Ø Dia (mm)	Notes
01	+	0.200	
S	X	0.220	
CT	Y	0.800	
AST	T	0.800	
4	Σ	0.001	
ST	Μ	0.001	
0	□	0.001	
5	◇	0.005	
4	◇	0.005	
S	◇	0.200	

REV	V1.0
DATE	10.05.2011
DES	JAW

Моїї ініціалі  
 БСБ\

ГАУЛЕ ЕСОГЕ VAGI2AMIE  
 Моїї ініціалі  
 ID Flansen Roman

