



\*\*\*\*\*

## RAPPORT DE STAGE- PLASTRI\_FRUGAL

ENTREPRISE : PLASTIC ODYSSEY

TUTEURS : Mejdî Nicri, Bob Vrignaud, Simon  
Bernard

TUTEUR SCOLAIRE : M. Portal

STAGIAIRE : DIENG IBRAHIMA

DUREE : 3 mois

\*\*\*\*\*

## Les remerciements

\*\*\*\*\*

Je tiens à remercier chaleureusement mon tuteur de stage Mejdî Nicri de par sa disponibilité, son envie de partager et sa pédagogie, m'a permis d'apprendre énormément sur la démarche scientifique à suivre pour mener à bien un projet et de mettre en pratique les connaissances acquises au cours de ma formation. Je remercie Bob Vrignaud de m'avoir tutoré durant ce stage en tant que responsable R&D. Je remercie aussi Benjamin DaMolliens qui m'a permis de découvrir plastic Odyssey de même Simon Bernard, Alexandre Dechelotte et Maité Abos d'avoir eu confiance en moi pour cette mission. Mes remerciements vont aussi vers l'ensemble de l'équipe de P.O de leur sympathie, leur esprit d'équipe et leur cohésion.

Ce stage était ma deuxième et la plus longue expérience en entreprise, il m'a permis de découvrir le fonctionnement d'une entreprise, de faire face à la rigueur au travail, la ponctualité, le respect du cahier des charges et surtout l'efficacité et l'aptitude à trouver des solutions face à des problèmes contraignants. Grâce à une équipe compréhensive et à l'écoute, j'ai pu effectuer mon stage dans un bon environnement de travail propice à la productivité.

# SOMMAIRE

INTRODUCTION .....	4
1. CAPTEUR DE TRI .....	5
1.1. CONTEXTE.....	5
1.2. CHOIX TECHNOLOGIQUE .....	6
1.3. FONCTIONNEMENT DU CAPTEUR..	7
2. TECHNOLOGIE DÉVELOPPÉE.....	8
2.1. LE SPECTROMÈTRE.....	8
2.2. LE MICROCONTRÔLEUR.....	9
2.2.1. ALGORITHME DE GOERTZEL.....	9
2.2.2. ACQUISITION DE DONNÉES.....	11
2.2.3. CODE DE CONTRÔLE LEDS & MOTEURS.....	15
2.2.4. ALGORITHME SYSTÈME COMPLET.....	18
2.3. RÉSULTATS OBTENUS CONCLUSION.....	19
3.LA BIBLIOGRAPHIE.....	20
4.LES ANNEXES.....	21

## Introduction

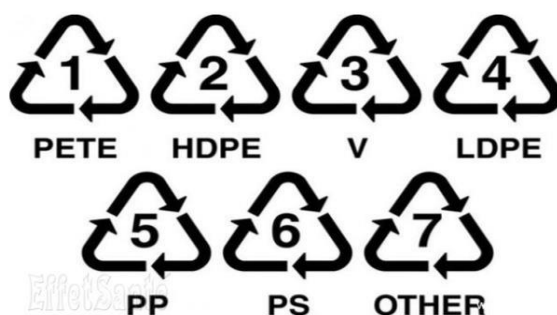
Ce stage s'inscrit dans le cadre de la formation à l'école Polytech Marseille. Il est d'une durée de 3 mois et s'articule comme suit : Programmation d'une chaîne de modulation et d'acquisition synchrone sur l'ATmega32U4 + un ADC externe. J'ai été accueilli au bureau de Plastic Odyssey à 4 Place Cardi Sanot, à Marseille, Colbert. Je collabore avec Mejdi Nicri qui se trouve à Paris et qui travaille sur le capteur de tri de même que Bob Vrignaud qui est le responsable R&D.

Ce rapport a pour but de présenter de rappeler les objectifs du stage et l'état d'avancement du projet. Permettant, par de même, de faire une présentation de l'avancement du projet aux responsables de Plastic Odyssey pour qu'ils puissent mieux décider de la suite.

Ce projet s'inscrit dans un contexte de lutte contre la pollution plastique. En effet, Plastic Odyssey est une entreprise qui œuvre dans la valorisation des déchets plastique avec comme idée maitresse la transformation de ces déchets grâce à une pyrolyse Low Tech qui permettra de faire avancer un bateau en mer qui aura pour mission de sillonner le monde et de mettre à disposition la technologie développée, de pyrolyse, de tri, de recyclage en produits dérivés comme des objets aux populations les plus concernées par la pollution plastique.

Chaque jour, 19 tonnes de de déchets plastique sont versés dans les océans, ce qui est colossal en termes de quantité de matière. La réutilisation de ces déchets est une source de génération d'emplois et permet de mettre en place une économie circulaire autour du plastique. Ainsi, Plastic Odyssey a décidé d'investir dans la R&D pour mettre en place des solutions viables dans le processus de valorisation. C'est là qu'intervient le capteur de Tri qui permet de différencier les différents types de plastique et leur famille grâce à un système que nous avons conçu au cours de ce stage et qui permettra de reconnaître à partir d'un spectromètre n'importe quel type de plastique.

L'importance du capteur de tri réside dans le fait qu'il permettra de savoir quel plastique utilisé pour la pyrolyse et lequel pour le recyclage en objet. En effet, il y'a trois grandes familles de plastique à savoir : **les thermoplastiques, les thermodurcissables et les élastomères**. Les premiers sont les plus utilisés car étant plus facile à recycler. Ils se divisent en petites familles :



# 1. CAPTEUR DE TRI

Le module de capteur de tri sur lequel j'ai effectué mon stage est la continuité d'un des modèles de capteur déjà proposé. Ce modèle est celui de Qing (Qing1) qui avait participé à la conception au cours de son stage effectué au sein de Plastic Odyssey. Cependant, le modèle proposé lors de mon stage est certes en continuité avec le travail de Qing (expliqué dans la partie 1.2) mais présente quelques différences. Dans ce modèle, nous avons pris en compte des éléments comme le temps d'acquisition, le choix de l'algorithme pour la FFT, le choix du microcontrôleur et des convertisseurs analogiques numériques.

## 1.1. CONTEXTE

Ce capteur s'inscrit dans un contexte de formation et de démocratisation de la technologie de tri plastique. On s'est rendu compte qu'il n'est pas évident d'identifier tous les types de plastique à l'œil nu et que le tri du plastique est très important pour savoir lequel est utilisable pour faire du recyclage en objet ou en carburant utilisé par les bateaux dans notre cas. Ce capteur se veut donc low tech.

En effet, Plastic Odyssey est une entreprise qui lutte contre la pollution et promeut la réutilisation et le recyclage des déchets plastiques, d'où la raison du choix Low Tech. Ce capteur de tri permettra donc de remplir une double fonction :

- Faciliter la formation : en permettant aux populations les plus exposées à la pollution plastique et qui n'ont pas accès à la technologie de se la procurer.
- La démocratisation : en effet, sur le marché il existe des capteurs de tri utilisés par les grosses entreprises industrielles. Ce capteur, à l'opposé de ce qui existe déjà sur le marché, sera bas prix, facile d'usage et facilement transportable, il sera accessible à tout le monde.

## 1.2 CHOIX TECHNOLOGIQUE

Le choix technologique a été au cœur de ce projet. En effet, pour le capteur de tri, on a choisi la **technologie optique**. Ce choix se justifie par le fait que pour notre modèle, l'objectif étant d'avoir un capteur fonctionnel traitant des échantillons de plastique, la technologie optique convient parfaitement. D'autant plus, ce capteur ne rentre pas dans le même registre que ceux des industrielles comme la machine VARISORT de Sesotec capable de trier des tonnes de déchets en quelques heures, il est plutôt axé pour l'éducation et la formation. Ensuite, pour la conception du capteur de tri, on a travaillé avec le microcontrôleur de la famille ATmega32U4 Arduino Micro. Ce choix se justifie par le fait qu'il est open source et permet aussi de travailler avec les bandes de fréquences voulues et aussi il est facile de l'intégrer dans un système grâce sa taille relativement petite. Les autres choix technologiques sont les suivants :

-**Les Leds choisies sont des SMD** (single mounted device) car permettant d'avoir une puissance d'émission et consommant le moins possible.

-**Les moteurs sont des micromoteurs** que l'on peut retrouver dans les lecteurs CD et ils sont pas à pas parce qu'ils permettent d'avoir une précision de positionnement qui est cruciale pour les photodiodes.

-Les **photodiodes** choisies sont des photodiodes puissantes qui correspondent à la taille du spectre du signal.

-Le choix de l'ADC a été crucial pour l'avancement de ce projet. Celui retenu est **l'ADS8586** (Annexe 2) car permettant d'avoir une fréquence d'acquisition de 100ksps, un PGA permettant de contrôler le gain manuellement, il est 16 bits et comporte 8 chanel qui peuvent être simultanés (voir Datasheet).

-**Le dev Board ADS8588** a été choisi parce qu'il nous permet d'effectuer des tests de conversion pour vérifier le fonctionnement de l'ADC et d'effectuer l'acquisition et la FFT sur le microcontrôleur via un adaptateur ZX-100 QSH.

-**L'adaptateur ZX-100 QSH** : permettant de relier le dev Board au microcontrôleur a été choisi pour faciliter la connexion entre les deux.

Lien vers le Bill of materials (BOM) pour ce capteur de tri : [BOM](#)

(Schéma des différents composants en **ANNEXE 1 et 2**)

### 1.3 FONCTIONNEMENT DU CAPTEUR DE TRI

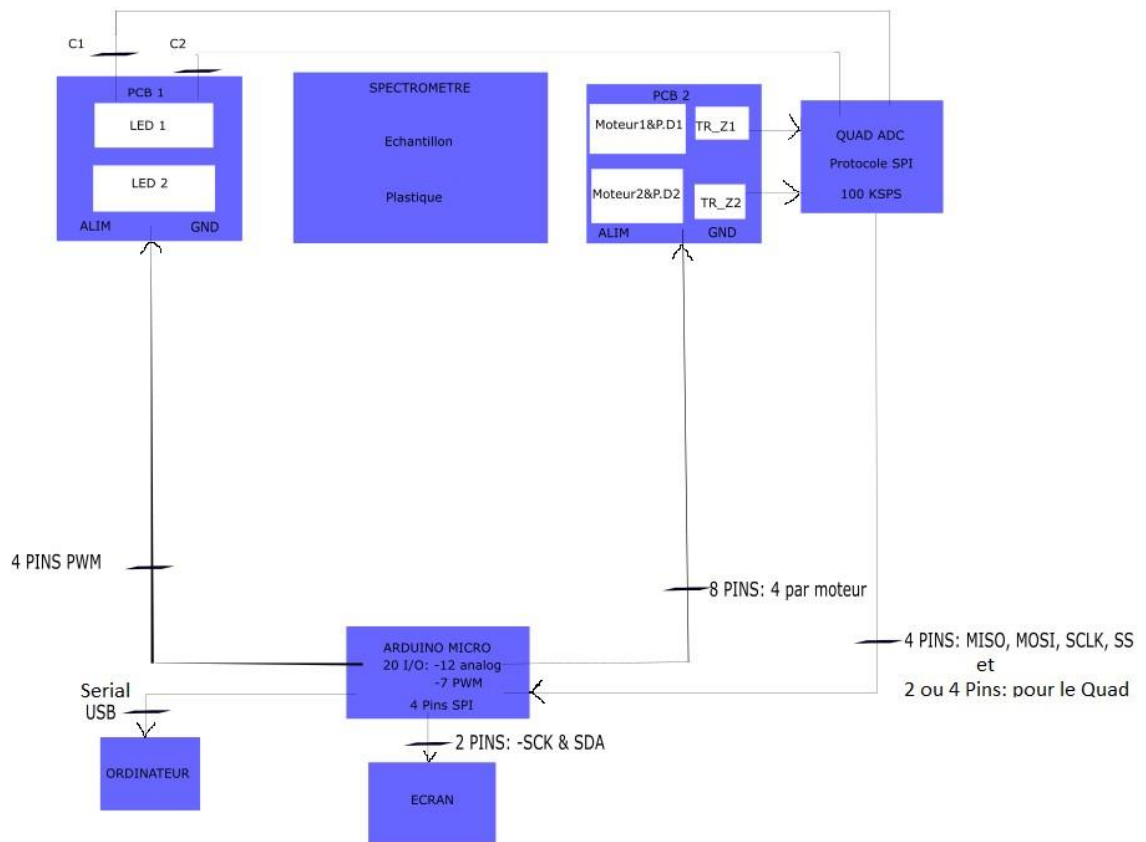


Schéma bloc général du capteur de tri plastique frugal

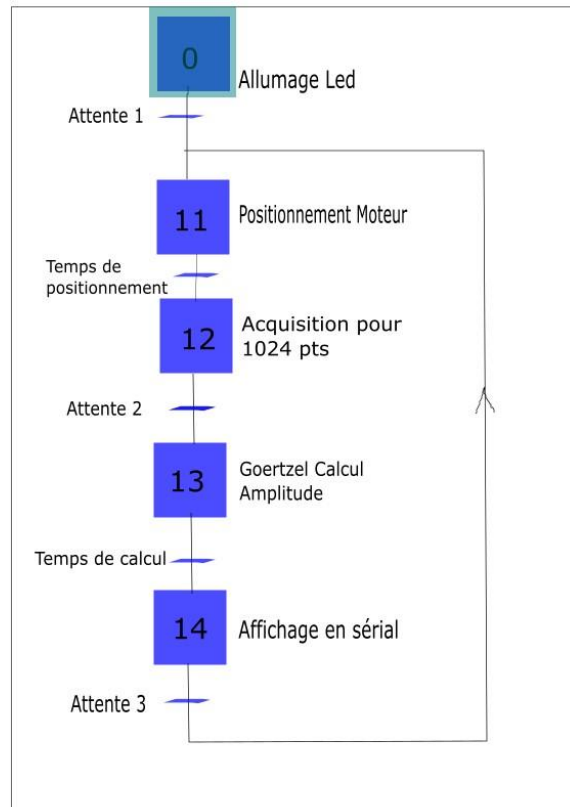
Ce capteur de tri est composé de deux grandes parties : **le spectromètre** et **le microcontrôleur**. La partie spectromètre sera développée par Mejdî Nicri et la partie Microcontrôleur par moi-même.

Le capteur de tri fonctionne de la manière suivante :

- Le microcontrôleur contrôle le système en permettant l'allumage des Leds, deux pour le moment, qui serviront à émettre l'infrarouge sur l'échantillon de plastique (on suppose transparent). Le choix des deux Leds s'explique par la nécessité d'avoir deux bandes de longueurs d'ondes différentes : 1100-1250nm et 1600-1700nm.
- Ensuite, il pilote le moteur qui est couplé avec une photodiode pour permettre la réception du signal transmis à travers l'échantillon de plastique qui sera par la suite amplifié et transmis au microcontrôleur.

- Ce dernier fait l'acquisition des données par une méthode de FFT dite de **l'algorithme de Goertzel**, permet de déterminer l'énergie transmise que l'on comparera à une base de données qui permettra de définir le spectre du plastique détecté et ainsi de déterminer le type de plastique recherché.

## • GRAFCET PROPOSE POUR LA CONCEPTION



Grafcet pour une led et un moteur

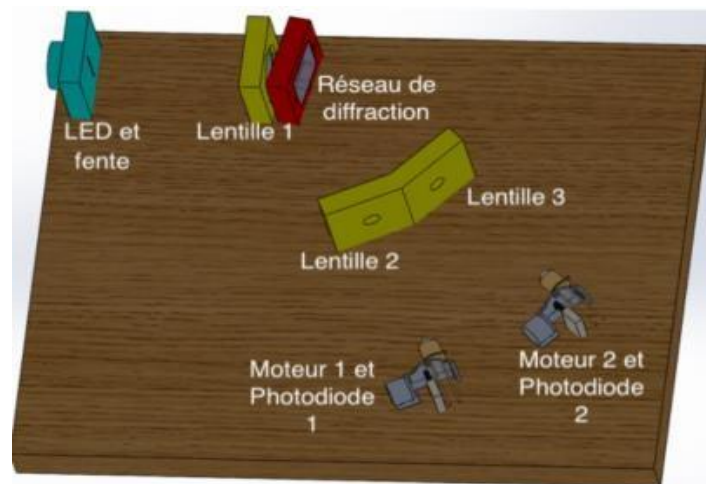
## 2. TECHNOLOGIE DÉVELOPPÉE

La technologie développée au cours de mon stage peut être présentée comme suit :

### 2.1. LE SPECTROMÈTRE

La partie spectromètre est développée par Mejdî Nicri. Dans cette partie, l'objectif est, à travers un système optique précis, de concentrer le spectre dans une zone précise en sortie du spectre pour que le moteur puisse balayer autour et permettre à la photodiode de détecter les bandes.





Modèle de spectromètre développé

## **2.2. LE MICROCONTRÔLEUR**

Le microcontrôleur utilisé est de la famille ATmega324U+, c'est l'Arduino Micro. Cette partie peut être subdivisée en 3 grosses sous parties : la FFT grâce à l'algorithme de Goertzel, l'acquisition de données et le contrôle de Leds et moteurs de même que l'affichage.

### **2.2.1 ALGORITHME DE GOERTZEL**

Dans cette partie, l'algorithme développé s'inspire du site de [embedded.com](http://embedded.com) et est développé par Kévin Banks. Le lien Github de l'algorithme est le suivant : [Code Goertzel](#).

L'algorithme permet de faire de la FFT sous une autre manière en permettant de détecter une fréquence dans une séquence d'échantillon. Il est aussi utilisé en traitement du signal pour la détection de DTMF (dual tone multiple frequency). Il a cette particularité d'être efficace en temps de calcul et d'être simple d'usage.

On appliquera l'algorithme sur le signal numérique qu'enverra la photodiode au microcontrôleur c'est-à-dire les échantillons de ce signal. Ce qui permet d'avoir l'enveloppe de l'énergie du signal détecté qui nous permettra par la suite à l'aide d'une base de données de plastique déjà existante de déterminer le type de plastique présent dans le spectromètre.

Les premiers tests de l'algorithme de Goertzel, en utilisant le logiciel Arduino nous donne les résultats suivants :

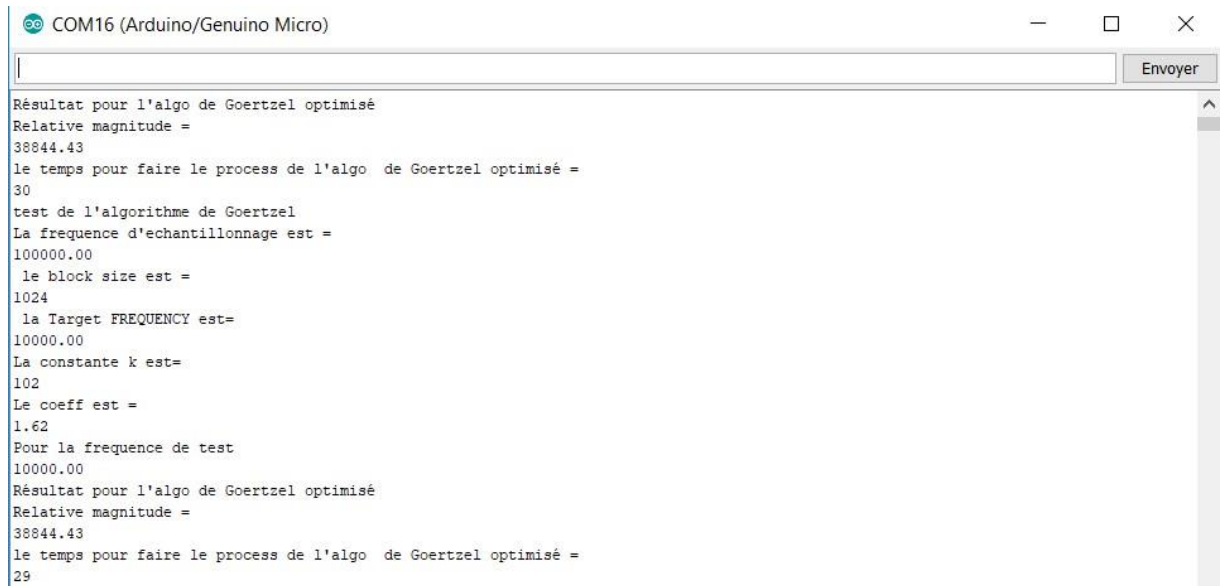


```
COM11 (Arduino/Genuino Uno)
La fréquence d'échantillonnage est =
100000.00
le block size est =
1024
la Target FREQUENCY est=
10000.00
La constante k est=
102
Le coeff est =
1.62
Pour la fréquence de test
10000.00
La partie réel est=
22976.15
la partie imaginaire est
-31320.71
Relative magnitude squared =
1508890112.00
Relative magnitude =
38844.43
le temps pour faire le process de l'algo de Goertzel basique =
113
```

**Titre** : Résultat de l'algo de Goertzel pour une sampling freq de 100kps,  
N=1024, freq=10khz et K=102 et coeff=1.621286

**Analyse** : dans ce cas, on a déclaré un tableau de 1024 échantillons (censé représenter le signal numérique) sur lequel on applique l'algorithme de Goertzel. On voit bien qu'on arrive à déterminer l'énergie du signal que l'on utilisera pour la base de données qui nous permettra de déterminer le type de plastique. Cependant le temps mis pour déterminer l'énergie est trop importante (113 ms pour 1024 échantillons) d'où la nécessité d'optimiser le code.

Donc pour la première optimisation du code, la première solution consistait à éviter d'utiliser des Serialprintln dans l'algo de calcul du temps d'exécution. Ce qui nous a permis de descendre jusqu'à 29 ms de temps d'exécution pour Goertzel, voir schéma ci-dessous :



```
Résultat pour l'algo de Goertzel optimisé
Relative magnitude =
38844.43
le temps pour faire le process de l'algo de Goertzel optimisé =
30
test de l'algorithme de Goertzel
La fréquence d'échantillonnage est =
100000.00
le block size est =
1024
la Target FREQUENCY est=
10000.00
La constante k est=
102
Le coeff est =
1.62
Pour la fréquence de test
10000.00
Résultat pour l'algo de Goertzel optimisé
Relative magnitude =
38844.43
le temps pour faire le process de l'algo de Goertzel optimisé =
29
```

### **Titre : Résultat du code de Goertzel optimisé**

**Analyse** : cependant, le résultat pour le code de Goertzel optimisé ne remplit pas le cahier des charges. En effet, le temps d'exécution de Goertzel doit, au pire des cas, tourné autour de 5 ms alors qu'ici nous avons 35 ms.

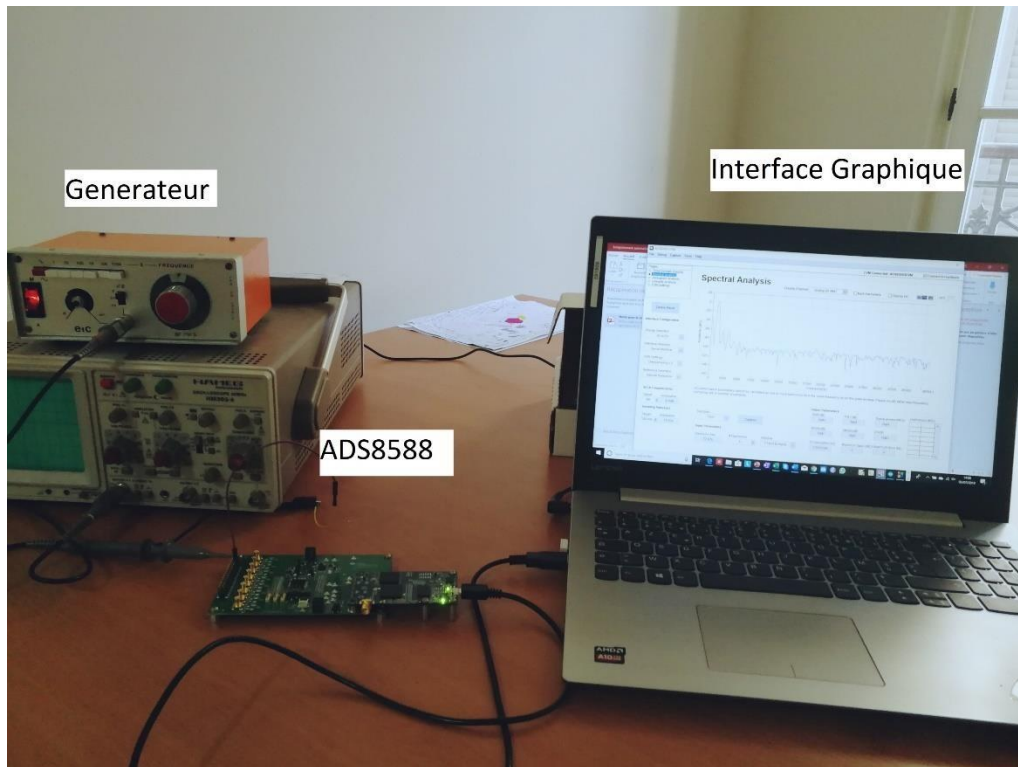
**Solution proposée** : il faut encore optimiser le code en jouant sur les types de données. En effet, le temps d'exécution de certaines variables est plus important que d'autre, donc l'idée ce sera de faire l'optimisation suivant ce choix. Cependant, à ce stade du stage, on a préféré mettre l'optimisation de coté pour se concentrer sur les autres parties à savoir l'acquisition de données et le test pour le système complet.

## **2.2.2 ACQUISITION DE DONNÉES**

L'algorithme d'acquisition de données permet de faire la réception de datas qui vont remplir un buffer de taille fixe que l'on définira dans l'algorithme sur lequel on appliquera ensuite l'algo de Goertzel. Ce buffer prendra les échantillons qui provienne du convertisseur Analogique/Numérique de l'ADS8588.

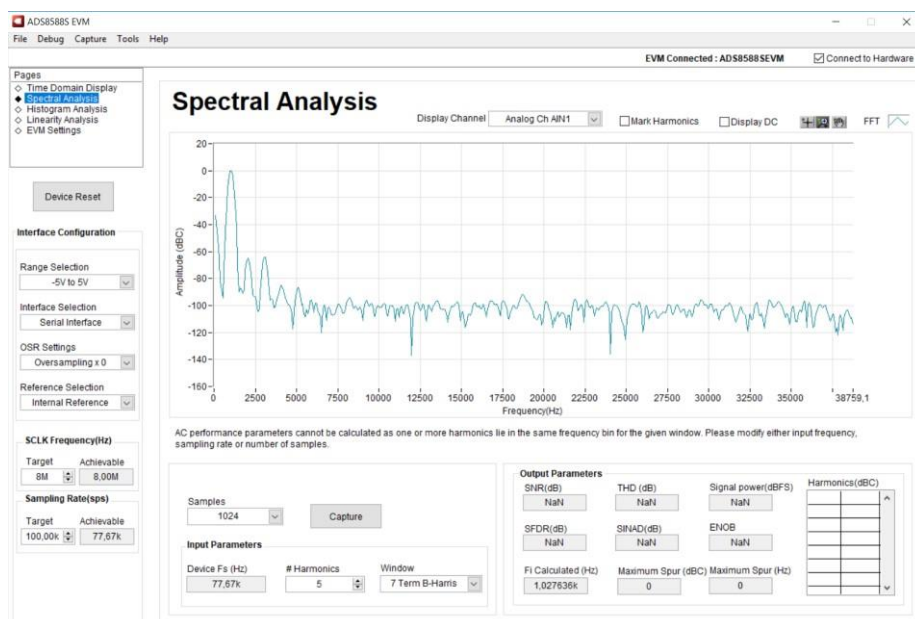
### **• TEST DU CONVERTISSEUR A/N ADS8588**

Pour faire le test de l'ADS8588 on utilise le montage suivant :



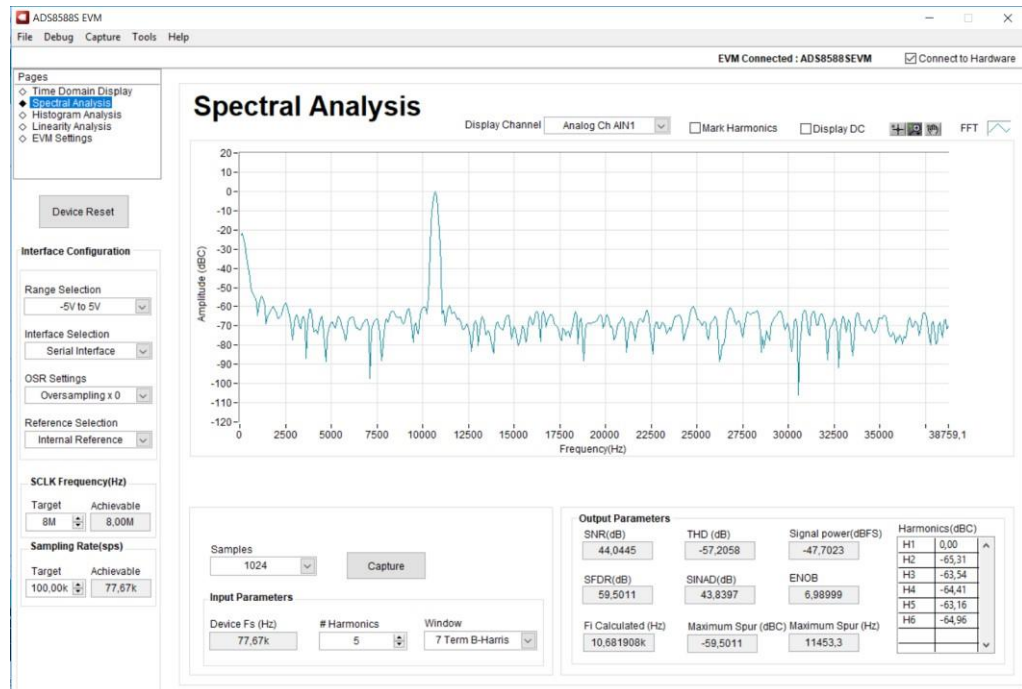
**Titre : Schéma de montage pour le test de l'ADS8588**

- En utilisant ensuite en entrée un signal de 4V crête à crête et de fréquence 1Khz, on obtient alors en sortie un spectre ou l'on voit la raie à 1Khz avec une amplitude maximale correspondante à celle du signal analogique converti.



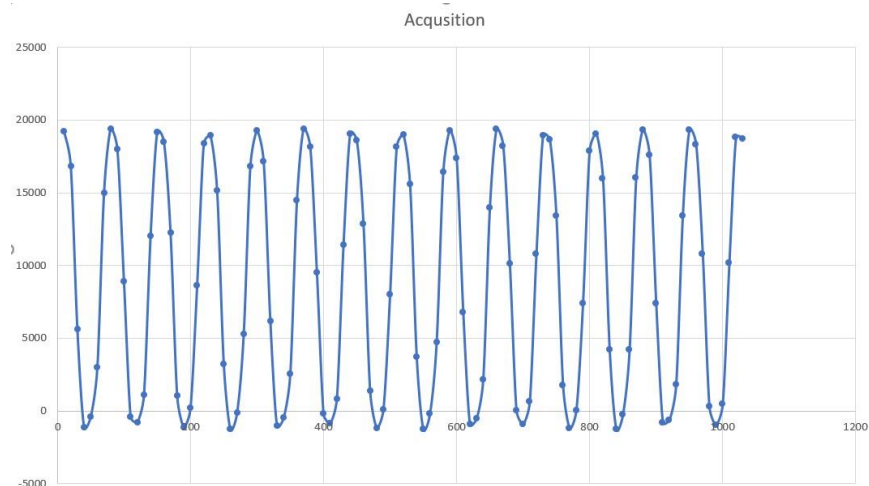
**Titre : Spectre de sortie de l'ADS8588**

- En utilisant ensuite en entrée un signal de 4V crête à crête et de fréquence 10Khz, on obtient alors en sortie un spectre où l'on voit la raie à 10Khz avec une amplitude maximale correspondante à celle du signal analogique converti.



**Titre : Spectre de l'ADS8588 pour une fréquence de 10Khz**

On peut récupérer les échantillons en sortie de l'ADS8588 sous forme de fichier csv. Ensuite, on retrace le signal analogique de départ sous excel pour vérifier si la conversion s'est bien effectuée. Donc pour un signal de 4V crête à crête à 10Khz, on a obtenu le signal suivant qui confirme bien la conversion.



**Titre : Signal récupérer à partir des échantillons de la conversion**

## • ALGORITHME D'ACQUISITION

(Lien Github pour l'algorithme de Goertzel : [ICI](#))

Dans cet algo, on a utilisé la bibliothèque digitalWriteFast pour optimiser l'écriture ceci nous permettant de gagner en rapidité.

On a utilisé une fonction **readAdc** qui permet de faire la lecture des datas numérique qui viennent de la conversion analogique, ensuite de les stocker sur un buffer de taille défini que l'on va utiliser pour le Goertzel.

### Résultat de l'acquisition

```
COM17 (Arduino/Genuino Micro)
27 20272852 8 3072
28 20272904 12 3072
29 20272916 12 3072
30 20272924 8 3072
31 20272936 12 3072
0.00
N t[us] dt[us] Value B
0 20296176 - 3072
1 20296188 12 3072
2 20296196 8 3072
3 20296208 12 3072
4 20296216 8 3072
5 20296228 12 3072
6 20296240 12 3072
7 20296248 8 3072
8 20296260 12 3072
9 20296272 12 3072
10 20296280 8 3072
11 20296292 12 3072
12 20296300 8 3072
13 20296312 12 3072
14 20296324 12 3072
15 20296332 8 3072
16 20296344 12 3072
17 20296356 12 3072
18 20296364 8 3072
19 20296376 12 3072
20 20296384 8 3072
21 20296396 12 3072
22 20296408 12 3072
23 20296416 8 3072
24 20296428 12 3072
25 20296440 12 3072
26 20296448 8 3072
27 20296460 12 3072
28 20296468 8 3072
29 20296480 12 3072
30 20296492 12 3072
31 20296500 8 3072
N t[us] dt[us] Value B
0 20319420 - 3072
1 20319432 12 3072
2 20319440 8 3072
3 20319452 12 3072
4 20319460 8 3072
5 20319472 12 3072
6 20319484 12 3072
7 20319492 8 3072
8 20319504 12 3072
9 20319516 12 3072
10 20319524 8 3072
11 20319536 12 3072
12 20319544 8 3072
13 20319556 12 3072
14 20319568 12 3072
15 20319576 8 3072
16 20319588 12 3072
17 20319600 12 3072
☐ Défilement automatique ☐ Show timestamp Les deux, NL et CR 9600 baud Effacer la sortie
```

**Analyse** : on constate que l'algorithme d'acquisition s'exécute autour de 8 et 12 us, ce qui est peu souhaitable. Nous voulons obtenir une valeur du temps d'exécution de 10us fixe parce que nous faisons l'acquisition à 100ksps et notre signal sera autour de 10 khz. Ce qui pourrait expliquer le fait que le temps ne soit



pas stable est le temps que mets la fonction micros pour calculer le temps d'exécution qui est de 4us.

A ce stade du projet, ce problème n'est pas encore résolu. Tout de même une des solutions serait de jouer avec les variables pour gagner plus en temps d'acquisition.

### **2.2.3. CODE DE CONTRÔLE LEDS & MOTEURS**

Pour rappel, les Leds doivent être modulées à une fréquence de 10khz (ou autour) et les moteurs doivent nous permettre de récupérer le signal en sortie du spectromètre grâce à la photodiode.

Pour le contrôle des Leds et des moteurs, on a écrit un code qui permet de le faire en utilisant l'I2C et le scan port nous permettant de vérifier si le branchement s'est bien effectué (**voir annexe 3**). Pour le moteur, la première partie consiste à réaliser le positionnement mécanique s'ensuit le déplacement pas à pas qui permet la détection des différentes bandes grâce à la photodiode. Pour les Leds, on va moduler une des quatre à une fréquence de 10Khz, de préférence ce sera la Led à l'input D2.

- Le code pour le contrôle des moteurs est un code assez basique qui est composé de deux parties : une première pour la setup et le positionnement mécanique du moteur et une deuxième pour le déplacement forward et backward du moteur.

( Lien du code [ici](#) )

- Le code pour la modulation des Leds est un code assez simple aussi. La particularité de ce code est que l'on travaille en SPI et que l'on doit moduler manuellement la fréquence autour de 10Khz.

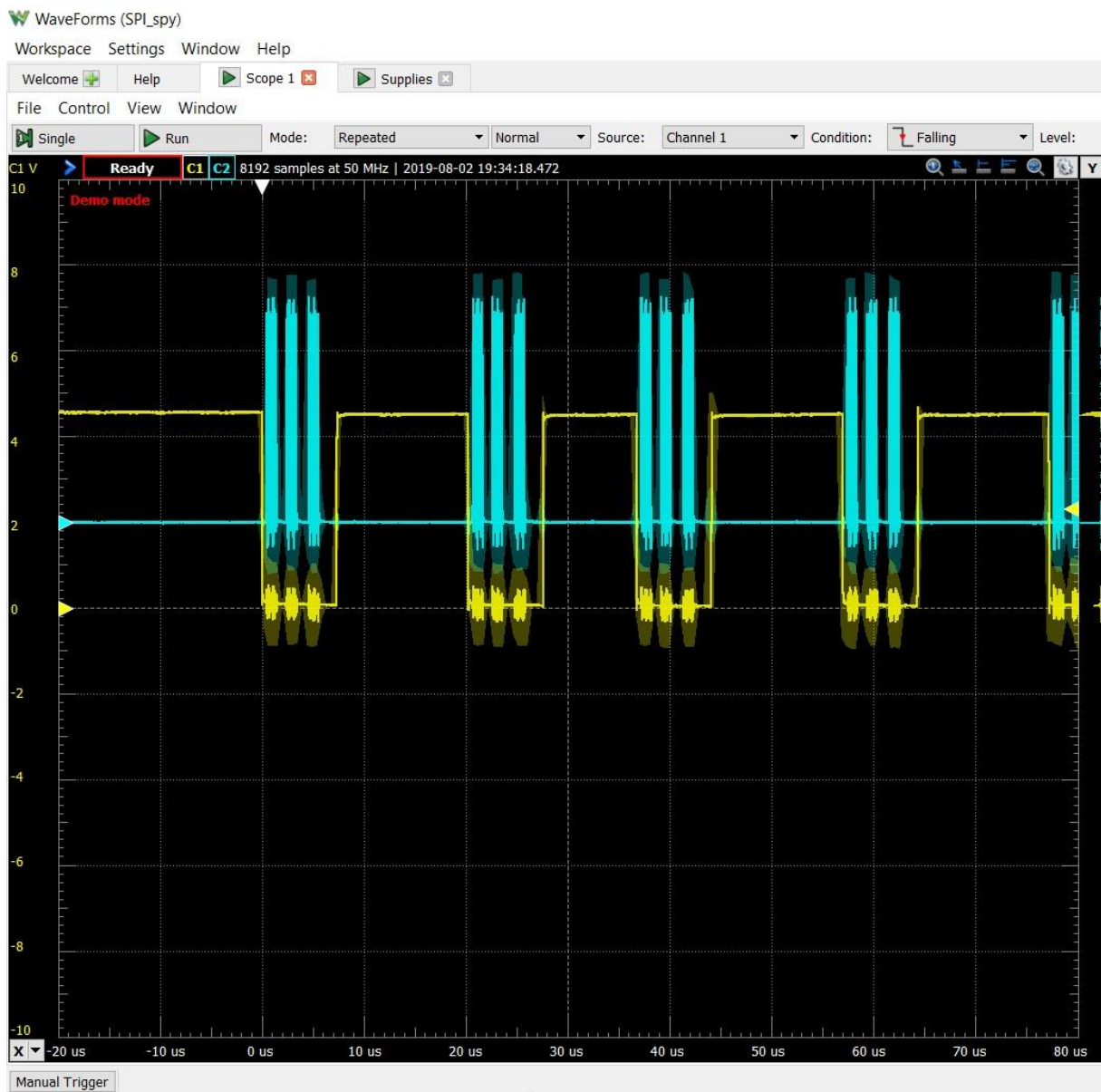
( Lien du code [ici](#) )

**Résultats** : le code pour le contrôle des moteurs marche bien, le test avec le PCB moteur montre bien le moteur qui effectue un déplacement mécanique avant de commencer à balayer pour récupérer le signal en sortie.

Pour la Led, nous avons réussi à moduler la D2 à une fréquence de 10Khz en jouant sur les Timers interrupts de l'ATMega32U4 précisément le Timer 2 (voir code).

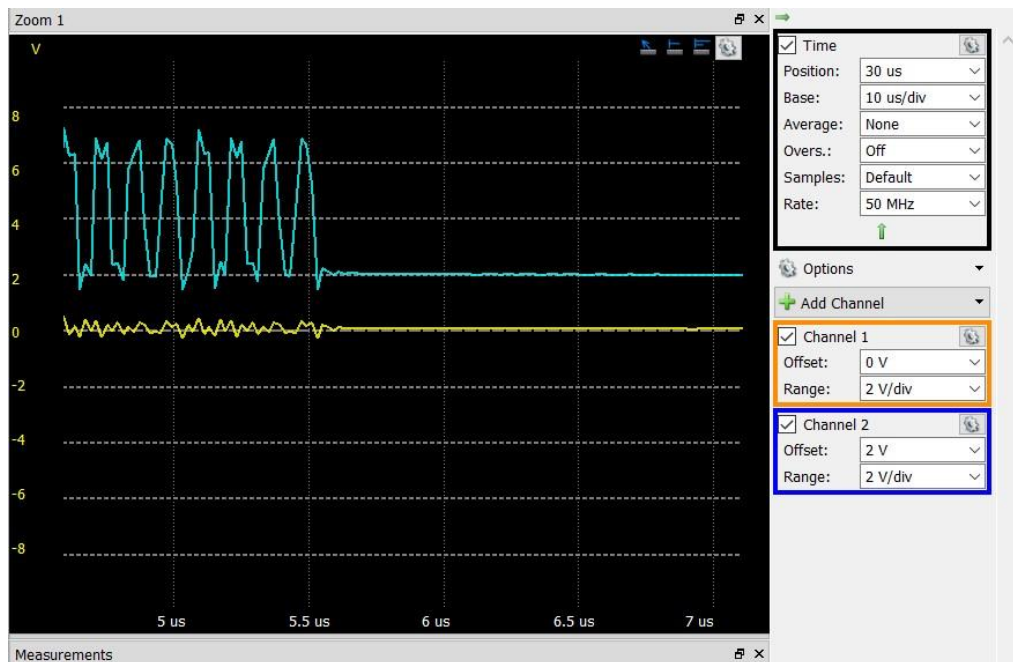
## • TEST PROTOCOLE SPI ENTRE ARDUINO ET MCP3204

Pour faire les tests pour vérifier le fonctionnement du protocole SPI entre l'Arduino et le convertisseur analogique numérique, on a utilisé l'Analog Discovery qui nous a donné le résultat suivant :



**Titre : résultat pour le protocole SPI entre l'Arduino et le MCP3204**





**Titre : Zoom sur le premier bloc de samples**

**Analyse :** On constate bien que l'on a 3 slots de 8 samples pour chaque trame ce qui correspond bien au 32 samples qu'on avait déclaré dans notre boucle for pour l'acquisition de données. Donc, on peut en conclure que l'on a bien le protocole SPI qui fonctionne bien entre l'Arduino et le convertisseur analogique numérique MCP3204.

## **2.2.4. ALGORITHME SYSTÈME COMPLET**

Pour l'algorithme du système complet, nous avons mis tous les codes ensemble dans un sketch [ici](#) pour faire un test général du système. Dans cette partie, tous les setups sont initialisés dans la fonction setup générale et on a défini toutes les fonctions de chaque partie et enfin la boucle loop() générale.

Les premiers tests montrent que le système est quasi complet. En effet, on arrive à avoir la modulation des Leds à 10Khz, le control des moteurs en même temps que l'acquisition même si on n'a toujours pas trouvé de solution pour avoir les 10 us pour l'acquisition. Les tests de FFT avec l'algorithme de Goertzel n'ont pas abouti à ce stade du stage. Donc on a intégré cette partie dans le système complet.

## **2.3. RÉSULTATS GLOBAUX OBTENUS**

Vu que le spectromètre n'est pas encore totalement fonctionnel, il est difficile d'évaluer les résultats finaux. En effet, l'ensemble des tests des codes se sont basés sur des tests avec des datas que l'on a généré nous-même. Ces tests marchent bien et vu les résultats obtenus, nous considérons qu'une fois le spectromètre fonctionnel, avec la photodiode, on pourra récupérer le signal, faire la conversion avec l'algo et enfin appliquer le Goertzel. Cependant la partie de gestion de la base de données plastique n'a pas été étudié durant ce stage.

## **CONCLUSION**

Ce stage de développement de capteur de tri frugal m'a permis de comprendre les étapes de fonctionnement d'un projet au sein d'une entreprise. A travers ce projet, j'ai énormément appris sur la démarche à suivre pour mener à bien un projet de la conception jusqu'à la réalisation, le respect du cahier des charges, et surtout j'ai pu mettre en pratique les connaissances acquises au cours de ma formation.

A ce stade, le capteur n'est pas encore fonctionnel, il reste une partie sur l'acquisition de données et une autre sur le code final avec l'algo de Goertzel de même que le test final avec le spectromètre. Tout de même, avec les codes que l'on a, on peut supposer que l'on n'est pas loin du prototype fonctionnel final qui pourra reconnaître un type de plastique une fois la chaîne complète.

### 3.BIBLIOGRAPHIE

Pour ce projet, les sources utilisées sont les suivantes :

- <https://cms.edn.com/uploads/SourceCode/09banks.txt>
- <https://www.embedded.com/design/configurablesystems/4024443/The-Goertzel-Algorithm>
- <https://github.com/jacobrosenthal/Goertzel>
- <https://f-leb.developpez.com/traductions/arduino-spi/>
- <http://www.ti.com/lit/ug/sbau278b/sbau278b.pdf>

## 4. ANNEXES

(Ci joint l'ensemble des annexes pour ce projet)

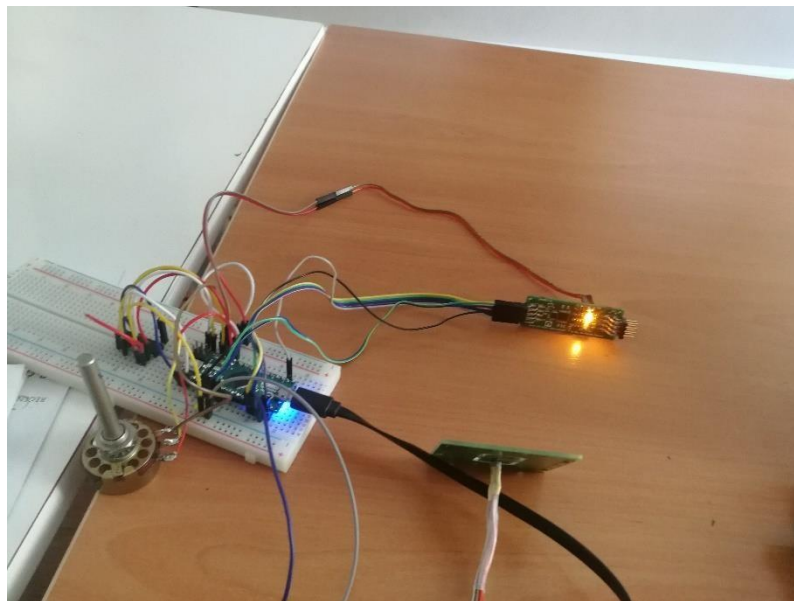
- **Annexe 1** : l'ensemble des composants utilisés dans le cadre de ce stage.



PCB Photodiode qui sera couplé avec le moteur



PCB leds pour la modulation



Test Photodiode avec les Leds

## Annexe 2 : Le convertisseur analogique numérique ADS8588

### 7.3 Schematics

Figure 22 through Figure 24 illustrate the EVM schematics.

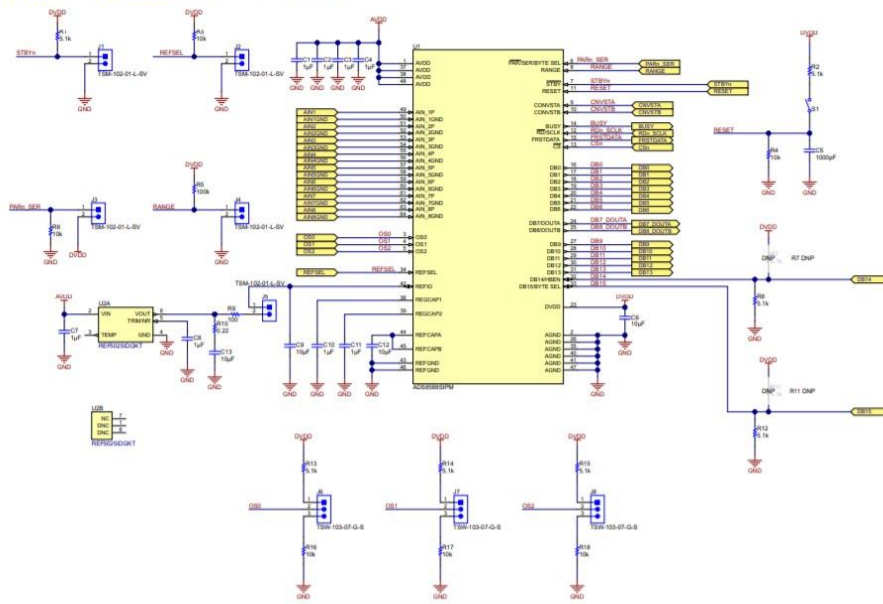
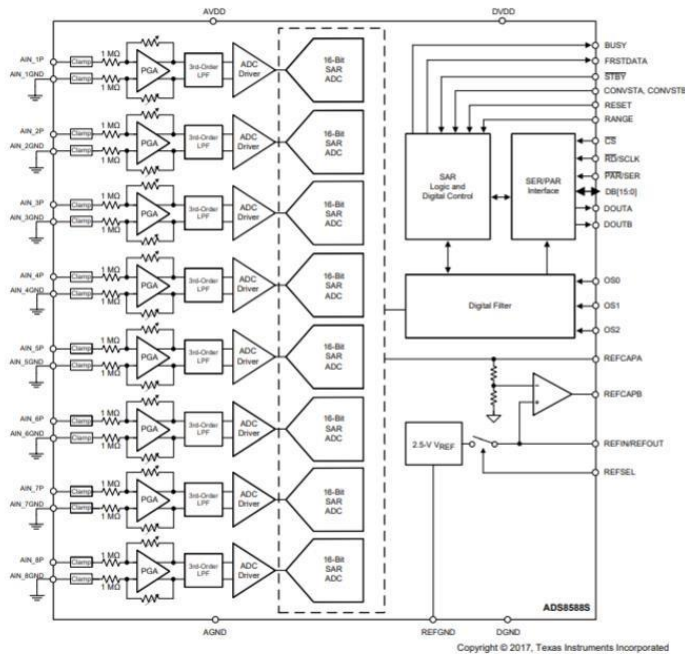


Figure 22. ADS8588EVM-PDK Schematic

## Schéma électrique de l'EVM ADS8588

### 8.2 Functional Block Diagram



Copyright © 2016–2017, Texas Instruments Incorporated

Submit Documentation Feedback

25

Product Folder Links: [ADS8588S](#)

## Schéma bloc de l'ADC 16 bits

**Product Name:** ZX100-QSH030 Samtec Breakout Adapter

**Product Description:** 30pins x 2 rows, 60 pin Samtec breakout adapter test board – Single Ended. Designed in 4 layer PCBoard with no vias on any signal traces, improved signal integrity and crosstalk.

- Designed in 4 layer PCBoard
- No via on any signal traces
- Improved signal integrity and crosstalk.
- Matching connector's 50Ω impedance – Reference plane impedance 50Ω for DC to 8GHz bandwidth applications

GND test points for easy access  
GND test point is tied to 2 inner GND planes

**Application:** Bringup, testing, emulation, development, modular design evaluations

**Mates with :** Samtec QTH030 QTH020(DP) - Also mates with ANY one bank of QTH060, QTH090, QTH120 : QTH020(DP) QTH040(DP) QTH060(DP) QTH080(DP) HQCD, HQDP, HFHM2  
Compatible with – differential Pair (DP), unused signal can be left unconnected or Grounded for improved noise immunity.

**Pitch:** 0.50mm (0.0197") High Speed connector

**Headers:** 0.1" (2.54mm) center, 0.025" SQ with 0.32" (8.13mm) post height

**ZX100-QSH030 pin configuration**

Layer	Header	Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Bottom	J4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
Top	J2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	

Note  
All ZEBAX products are RoHS compliant and Lead Free unless otherwise indicate

**Titre : Adaptateur ZX100 QSH030 utilisé pour l'interface Arduino ADS8588**



**Titre : ADS 8588 convertisseur Analogique/Numérique**