



[REALISATION D'UN SMART METER]

Encadrants: *Xavier Redon, Alexandre Boé*

Industriel: *Guillaume Renault*

Table des matières

I)-Introduction	2
I.1)-Quel est le but du projet ?	2
I.2)-Pourquoi avons-nous choisi ce projet ?	2
II)-Cahier des charges	3
Etude électronique	4
I)-Comment calculer la puissance active?	5
II)-Etude de la première approche :	6
III)-Etude de la deuxième approche :	7
III.3)- Circuit intégré « Maxim 78M6610+LMU »	7
IV) -Bilan	7
V)-Calcul de la consommation du circuit.....	8
VI)-Etude du schématique	9
VI.1)-Etude de la fonction « Mesure de tension » :	10
VI.2)-Etude de la fonction « Mesure de courant » :	10
VII)-Réalisation du PCB :	12
Soudage de la carte électronique.....	14
Mise sous tension de la carte	15
VI.3)-Etude des choix de communication.....	16
Réalisation d'une communication UART	18
Configuration des registres avant calibration	20
Configuration des registres point de référence	22
Lancement de la calibration	23
Mesures des valeurs utiles et mise en forme.....	24
I)-Programmation et première approche.....	25
I.1)Le programme MODEM :	26
I.2)-Etude des bibliothèques :	26
Comportement émetteur:.....	28
Comportement récepteur:	29
Processus de calibration:.....	30
Conclusion	31
ANNEXES.....	32
Liste des composants :	32
Développement d'un programme de communication RF :	34
Filtrage des entrées analogiques ».....	36
Etude de la fonction « Mesure de courant ».....	38

I)-Introduction

Dans le cadre de notre dernière année d'études en école d'ingénieur Polytech Lille, il nous est demandé de choisir un projet comportant une étude, une conception et une réalisation sur un sujet préalablement défini. Ce projet a pour but de nous responsabiliser et nous confronter aux démarches et aux problématiques que peuvent rencontrer les ingénieurs dans leur travail. L'intérêt principal est aussi d'approfondir certains domaines de notre formation.

I.1)-Quel est le but du projet ?

La consommation d'énergie est un poste de dépense qui prend de l'importance et pèse de plus en plus sur les factures d'électricité des particuliers. Aujourd'hui, mesurer en continu ces consommations énergétiques est devenu indispensable. Créer un outil de mesure permettrait au particulier de visualiser l'impact de ses gestes, d'établir des prévisionnels financiers et d'adopter un comportement éco-responsable dans le but d'obtenir un bâtiment à basse consommation d'énergie.

Ce projet, proposé par Guillaume Renault, tuteur industriel, consiste en la réalisation d'un « Smart Meter ». Ce dispositif a pour but final de transmettre à une « base » la puissance consommée d'un appareil électrique présent en aval afin d'y établir des estimations de dépense énergétique.

I.2)-Pourquoi avons-nous choisi ce projet ?

Intéressés par les sciences de l'électronique et des systèmes embarqués, ce projet est pour nous un bon compromis entre le développement hardware et software. En effet, en partant du cahier des charges, il nous est demandé d'avoir une réflexion sur les composants choisis, de designer et router une carte mais également de configurer les contrôleurs et créer l'échange radio avec la base.

II)-Cahier des charges

Tout d'abord, le but du projet est de mesurer la tension instantanée, le courant instantané et la consommation instantanée d'un appareil connecté sur une prise secteur afin d'en extraire la puissance active de consommation.

La seconde partie est de transmettre ces mesures en RF via un module « Panstamp ». Pour atteindre ces objectifs, une carte électronique devra être réalisée et s'interfacier entre la prise secteur et l'appareil dont on souhaite mesurer la consommation. De plus, la carte électronique devra être conçue pour être alimentée par la prise secteur mais la consommation de cette carte doit être négligeable.

La plage de consommation mesurable sera de 1W à 8KW avec un pas au plus proche de 0.1W ceci permettra à l'utilisateur de mesurer la consommation des appareils en veille.

Points d'attention :

- Les mesures devront tenir compte du cos Phi
- La plage de consommation mesurable sera [1W - 8kW] avec un pas au plus proche de 0,1W
- S'agissant d'un circuit de mesure de consommation, sa consommation propre (hors module Panstamp) doit être négligeable.

Pour réaliser ce système, il est préférable d'utiliser les modules « Panstamp ». Un module contient un Atmega328p et une interface RF CC1101. Les modules « Panstamp » sont programmés comme n'importe quelle autre plateforme Arduino, avec la différence que le panstamp peut faire de la communication sans fil à basse puissance. (1uA en mode veille et 2.5mA pendant les transmissions)

Pour ce faire, le projet peut être réalisé de la manière suivante :

1. Choix des composants afin de respecter le cahier des charges
2. Réalisation d'une carte électronique de test
3. Programmation des « Panstamps » afin d'obtenir une communication sans fil
4. Calibration du capteur
5. Tests de la partie calibration
6. Tests du système dans son ensemble

Etude électronique

Comme dit précédemment, nous nous sommes consacrés dans un premier temps à la réalisation de la partie électronique. Ainsi, le module « Panstamp » comportant une interface RF et un microcontrôleur atmega328p étant déjà sélectionné par notre tuteur industriel (pour sa faible consommation), la première étape a été de choisir les composants nécessaires pour la mesure de la puissance et l'auto-alimentation de la carte.

Dans un premier temps, nous nous sommes concentrés sur le dispositif d'alimentation et nous en avons déduit qu'il est plus logique d'étudier le dispositif en aval, c'est-à-dire le circuit de mesures afin de déterminer le dimensionnement de l'alimentation. Ainsi, la première étape de la conception électronique consista en la sélection de la technologie employée et des composants utilisés.

Le dispositif à mettre en place a pour objectif de donner la consommation d'une charge en aval. Pour cela, il nous faut déterminer la puissance active en temps réel.

1)-Comment calculer la puissance active?

La puissance active ou puissance moyenne est la puissance réelle consommée par la charge. Elle est donnée par :

$$Pa = \frac{1}{T} \int u(t).i(t) d(t)$$

Ainsi, numériquement il suffit d'échantillonner la tension $u(t)$ et $i(t)$ en fonction du temps de manière à respecter le théorème de Shannon. Ainsi, il faut au minimum que :

$$fe > 2.f$$

Avec :

$fe = \text{fréquence d'échantillonnage}$

$f = \text{fréquence du signal à mesurer}$

PS : Plus fe est importante, plus l'erreur sur la puissance mesurée est faible.

Il faut donc un dispositif capable d'échantillonner à 100 Hz. Cependant, afin de réduire l'erreur de mesure, il est de convention de prendre $fe = 20 * f$ soit $fe = 2000 \text{ Hz}$.

Nous retenons alors deux possibilités à étudier :

- La conversion s'effectue directement à l'aide de l'ADC (« Analog to Digital Convertor ») du contrôleur « Panstamp ». Ainsi, grâce à l'échantillonnage et la numérisation des tensions renvoyées par la présence d'un capteur de courant et un capteur de tension, il est possible d'obtenir la puissance instantanée.
- La conversion, l'échantillonnage et le calcul de la puissance est effectuée via un circuit intégré spécialisé. La puissance est alors lue directement par le Panstamp.

II)-Etude de la première approche :

Le « Panstamp » embarque un ADC 10 bits disponible sur 8 voies analogiques. Comme décrit dans la datasheet, il est possible de configurer son fonctionnement. Par exemple, il peut être possible de régler la fréquence de son horloge (à l'aide d'un quartz externe) ou encore le nombre de voies analogiques converties par celui-ci.

Afin de déterminer la faisabilité de cette solution, nous allons effectuer les calculs suivants en faisant varier la fréquence de l'horloge de l'ADC dans le cadre d'une utilisation non optimum de celui-ci.

Le « Panstamp » permet de fonctionner à une fréquence comprise entre 50 et 200 kHz. Sachant que l'ADC met en moyenne 15 cycles pour effectuer une conversion, on en déduit le tableau suivant :

Horloge (kHz)	Fréquence de conversion si 1 voie utilisée (kHz)	Fréquence de conversion dans le pire des cas (Hz)
200	13	1625
50	3.3	416

Ainsi, on en déduit qu'il est possible de réaliser une conversion directement via le « Panstamp » seulement si celui-ci est bien configuré ou que la fréquence d'horloge de l'ADC est élevée. Cependant, plus la fréquence de l'horloge est élevée, plus l'imprécision sera grande (inertie) donc cette première approche ne correspond pas à notre application.

Figure 23-6. ADC Timing Diagram, Auto Triggered Conversion

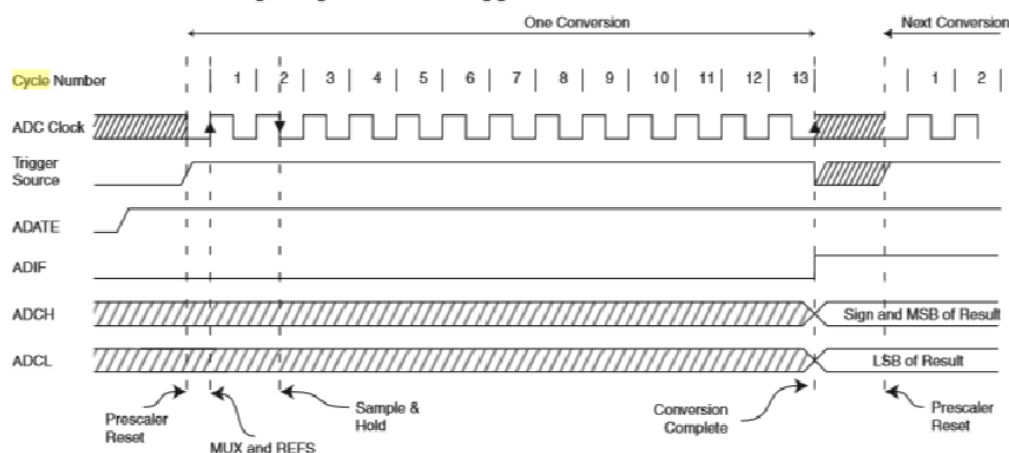


Figure 1 : chronogramme datasheet atmega328p

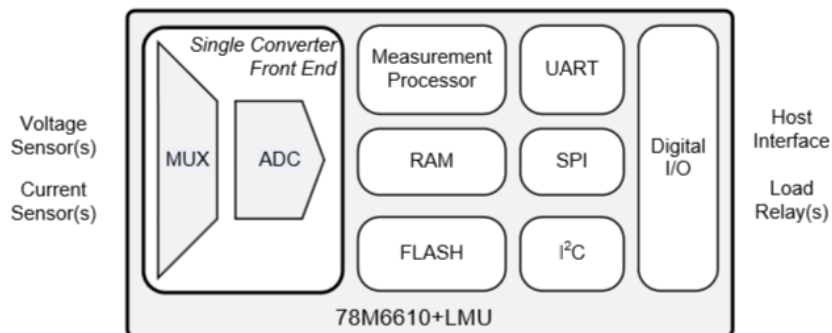
III)-Etude de la deuxième approche :

Tel que mentionné précédemment, la deuxième approche consiste à implanter sur la carte électronique un circuit intégré spécialisé permettant le calcul direct de la puissance instantanée. Après étude de plusieurs composants, le circuit intégré 78M6610+LMU de « Maxim Integreted » est retenu.

III.3)- Circuit intégré « Maxim 78M6610+LMU »

Le 78M6610+LMU est un circuit numérique spécialisé dans la mesure d'énergie. Il comporte un convertisseur Analogique-Numérique 24 bits multiplexé sur 4 voies. Son microcontrôleur intégré permet à la fois différentes mesures de puissance tel que la puissance active, la puissance réactive, la puissance apparente, le facteur de puissance. Le circuit intégré est capable de mesurer les valeurs instantanées de la tension, du courant et des puissances mais aussi de calculer l'énergie consommée. Le composant comporte une liaison série SPI, I²C ou UART. Nous pouvons donc communiquer en série entre le circuit intégré et notre microcontrôleur atmega328p.

De plus, le composant Maxim « 78M6610+LMU » permet d'obtenir un échantillonnage $f_e = 4000 \text{ Hz}$ par voies et répond donc de manière correcte à la contrainte d'échantillonnage. Ce composant est idéal pour notre application.



IV) -Bilan

De par les études effectuées précédemment, nous pouvons en déduire que la deuxième approche est plus cohérente puisqu'elle permet un taux d'échantillonnage plus important et donc une précision accrue. Enfin, cela libère des ressources conséquentes sur le « Panstamp » pour la réception et la transmission des données par radio fréquence. Cela permettra notamment de mettre en mode veille le « Panstamp » lorsque celui-ci n'est pas sollicité.

V)-Calcul de la consommation du circuit

L'ensemble de nos éléments de mesures étant choisi, il nous est possible de dimensionner l'alimentation en conséquence. Ainsi, afin de choisir une alimentation assez compacte, on sélectionne un montage hacheur type « Flyback ».

L'élément sélectionné est un dispositif intégrant à la fois le circuit de régulation et le transformateur:

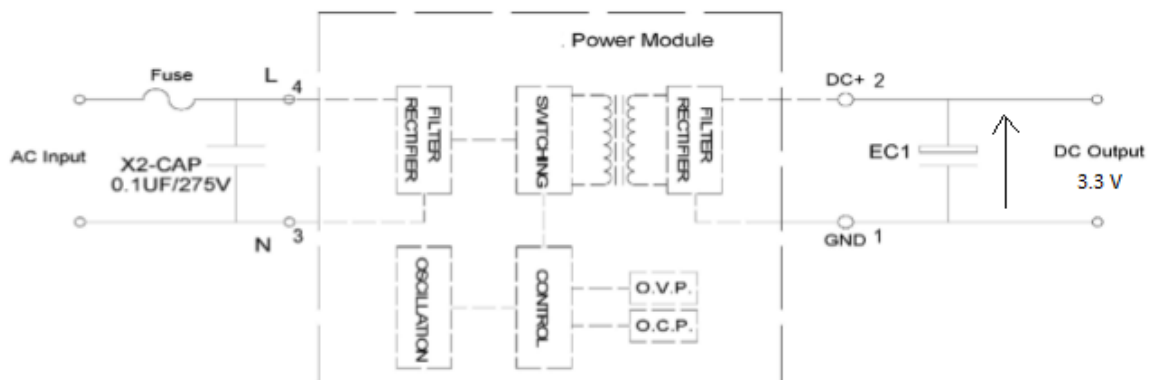
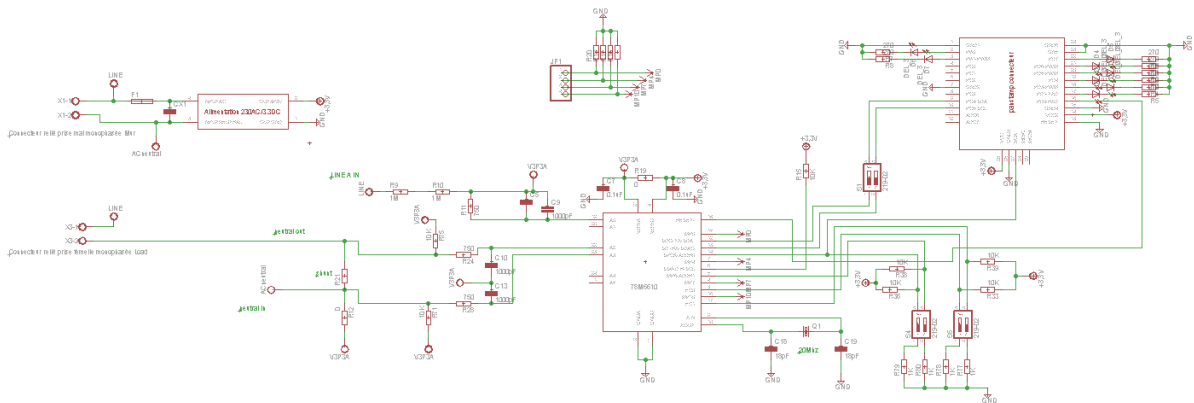


Figure2 : convertisseur AC/DC : VTX-214-001-103

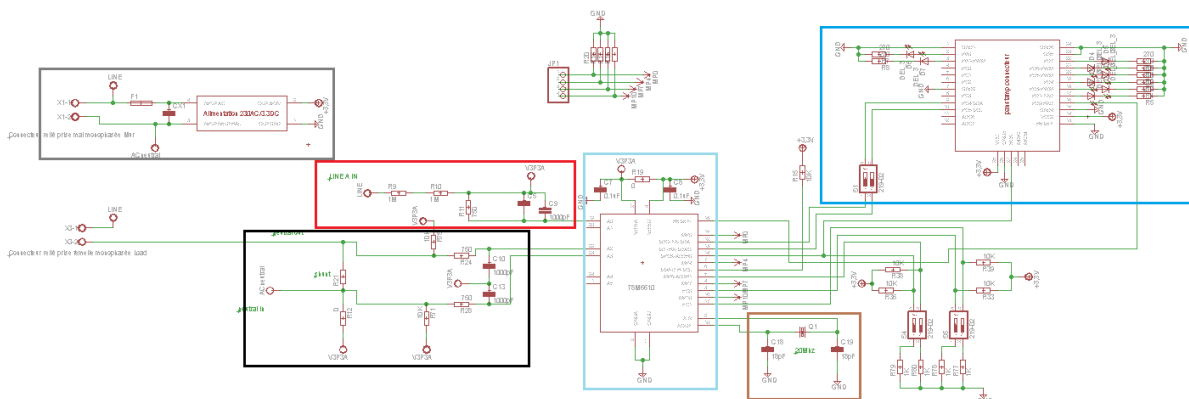
Sachant que le composant Maxim 78M6610+LMU à une consommation maximale de 10.3 mA et que celle du Panstamp est de 2.5 mA, on en déduit que la consommation totale est d'environ 12.8 mA. Ainsi, en prenant une marge supérieure, tout type d'alimentation 3.3V pouvant fournir une puissance de 43mW en sortie convient.

VI)-Etude du schématique

Après avoir étudié le composant Maxim 78M6610+LMU et son référence design, nous avons commencé à créer le schématique de notre circuit imprimé sur le logiciel EAGLE. Pour la réalisation de celui-ci nous nous sommes en grande partie inspirés du schématique du référence design et nous avons ajouté les éléments nécessaires pour une alimentation secteur 230V, l'intégration du « Panstamp » permettant d'envoyer les informations en liaison radio et des éléments supplémentaires pour faciliter le débogage lors de la calibration du circuit intégré (configuration des registres via des LED).

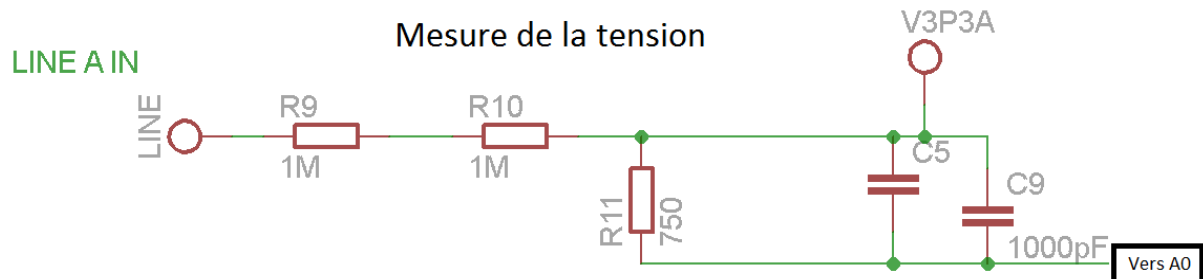


Nous pouvons découper notre schématique en plusieurs fonctions :



VI.1)-Etude de la fonction « Mesure de tension » :

Nous ne pouvons directement imposer une tension de 230V aux bornes du composant Maxim. Ainsi, il est nécessaire de passer par l'étage d'adaptation présent ci-dessous :



Les résistances série R9, R10, R11 limitent le courant et réalisent un diviseur de tension sur l'entrée tel que :

$$U(R11) = \frac{R11}{R11 + R10 + R9} \cdot Vline$$

De plus, l'alimentation V3P3A est placée afin d'effectuer un offset (de référence) tel que :

$$V(A0) = \frac{R11}{R11 + R10 + R9} \cdot Vline + V3P3A$$

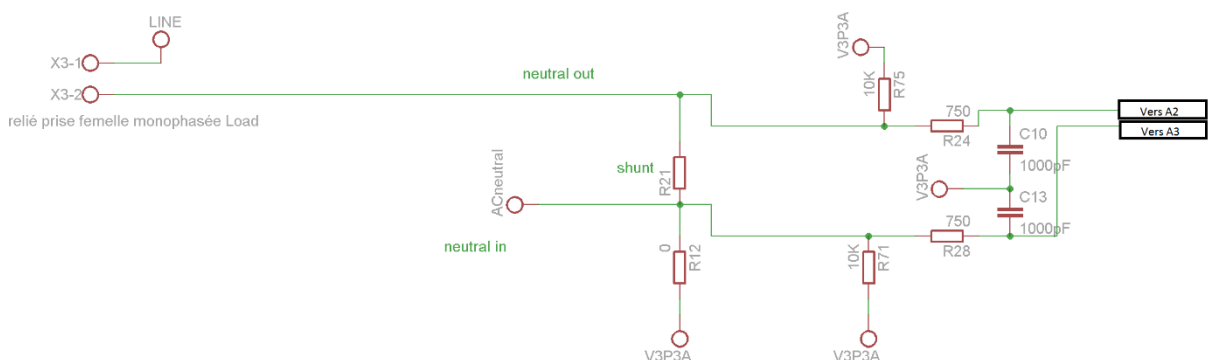
En remplaçant par les valeurs numériques, nous obtenons :

$$V(A0) = 3.75 \cdot 10^{-4} \cdot Vline + 3.3 V$$

On en déduit pour $-230\sqrt{2} < Vline < 230\sqrt{2}$:

$$3.178 < V(A0) < 3.422 V$$

VI.2)-Etude de la fonction « Mesure de courant » :



Ici, on mesure la différence de potentiel présente sur la résistance de shunt. Celle-ci nous donne une image du courant circulant dans celle-ci.

$$I(t) = \frac{U(t)}{R}$$

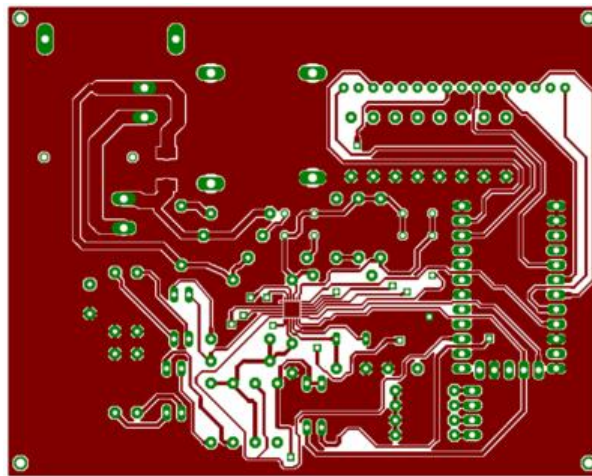
Lorsque la tension aux bornes du shunt est positive, c'est la sortie A2 qui mesure cette différence de potentiel, dans le cas contraire, c'est la sortie A3 qui relève la mesure. Comme pour la mesure de la tension, V3P3A sert de référence de mesure.

Remarque : Les capacités présentes aux bornes A2 et A3 du composant Maxim réalisent un filtrage des hautes fréquences. Veuillez-vous reporter à la section « Filtrage des entrées analogiques » de l'annexe pour plus d'informations.

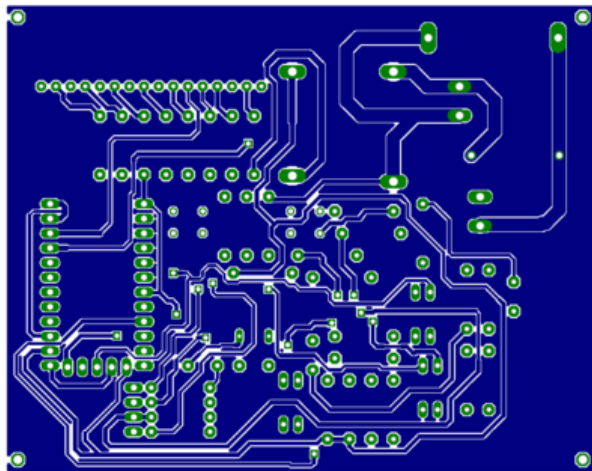
VII)-Réalisation du PCB :

Le PCB est la dernière étape avant le tirage de la carte. En effet, celui-ci permet le placement des composants et le routage des pistes électriques sur la carte afin d'en créer un typon. Cette étape demande donc beaucoup de minutie et d'expérience de jugé, c'est pourquoi nous avons donné trois semaines de notre temps sur cette partie.

Dans un premier temps, étant plus familiarisé aux composants traversants, nous avons entrepris une première version présente ci-dessous :

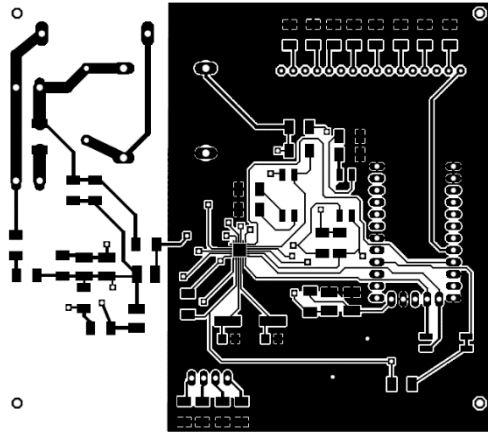


TOP

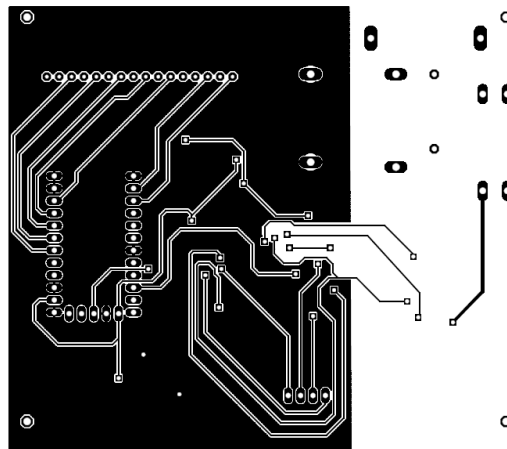


BOTTOM

Cependant, le fait de placer des composants traversant influe sur les lignes de champs et crée par la même occasion de nombreuses ruptures de masse. Ainsi, en remplaçant la technologie traversant par du CMS et en suivant les conseils de nos professeurs, nous avons supprimé les languettes et îlots de masse pour arriver à la carte réduite de prototype suivante :



TOP



BOTTOM

Une fois la carte tirée, nous commençâmes à travailler en parallèle. Une personne du binôme s'occupant de la partie software via la réalisation de la communication Radio Fréquence et du protocole UART entre deux "Panstamps" tandis que l'autre membre s'occupait du soudage de la carte et de la compréhension du protocole de calibration du Maxim 78M6610+LMU.

Soudage de la carte électronique

Nous avons soudé la carte en trois semaines. Certaines parties ont été recommencées plusieurs fois, notamment afin de respecter certaines règles de fixations nécessaires à la sécurité sur la partie haute tension. La carte a ensuite été testée au multimètre afin de vérifier la bonne soudure des composants et l'état des pistes.

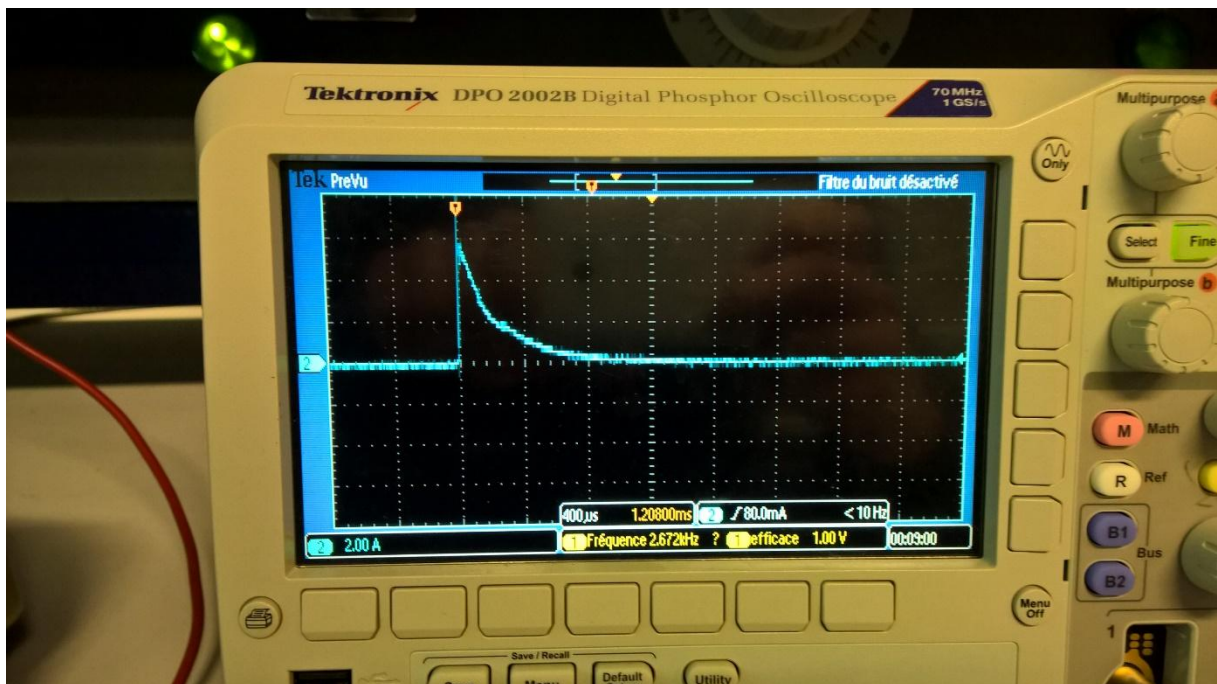


Mise sous tension de la carte

Lors de la première mise sous tension de la carte, le fusible de protection de l'alimentation de celle-ci a grillé. Nos intervenants nous ont donc guidés vers de possibles défaillances et les possibilités pour y remédier. Ainsi, après relecture de la "datasheet" de l'alimentation à découpage, nous avons changé l'ancien fusible (63 mA) par un fusible temporisé de valeur plus grande afin que celui-ci résiste aux courants d'appels invoqués par le circuit "selfique" du transformateur de l'alimentation AC-DC.

Afin de vérifier le bon fonctionnement de la carte, nous avons placé une résistance purement résistive permettant de faire passer un courant de 5A ampères sous 230 Vrms. Le fonctionnement de la partie haute tension est alors validé.

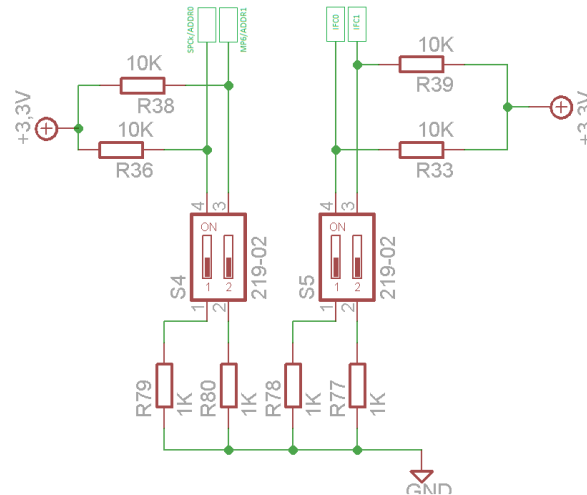
Ci-dessous, le relevé de l'appel de courant côté alimentation lors de mise sous tension. Nous obtenons un pic de courant de l'ordre de 6A. D'où la nécessité d'un fusible 150mA temporisé.



VI.3)-Etude des choix de communication

Sur la carte réalisée, les switches permettent de choisir le protocole de communication série (SPI ou I2C).

Le tableau suivant résume le protocole choisi en fonction de la position des commutateurs.

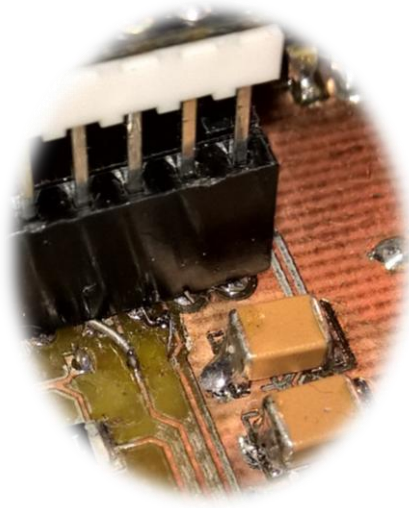


X = Ne pas tenir compte

	IFC0	IFC1
SPI	0	x
I ² C	1	1
UART	1	0

Nous souhaitions au début choisir la communication la plus simple et la plus efficace possible. Ainsi, nous nous sommes orientés sur une communication entre le "Panstamp" et le Maxim mettant en œuvre le protocole SPI. Cependant, il apparaît page 59 de la "datasheet" que le protocole SPI ne permet d'obtenir la valeur de la puissance instantanée et ce, dû à une plage d'adressage registre via le protocole SPI trop faible (<3F). L'I2C posant les mêmes problèmes, il est nécessaire de passer par la dernière solution, à savoir l'utilisation du protocole UART permettant d'accéder à l'ensemble des registres.

Nous avons donc modifié la carte en conséquence. Nous avons notamment relié les broches PD3 et PD4 respectivement aux broches Rx et Tx du Maxim afin de mettre en place une communication via port série logiciel.



Les soudures de modifications

Réalisation d'une communication UART

Le protocole UART permet de créer un réseau comportant jusqu'à 255 unités. Cependant, comme seul un composant Maxim 78M6610 est présent sur la carte, il n'est pas nécessaire de tenir compte de l'ensemble des travaux de configuration pour l'UART.

Nous choisissons la communication UART pour communiquer entre le "Panstamp" et le circuit intégré (78M6610). Nous choisissons également d'opérer suivant le fonctionnement suivant:

- 9 600 Bauds
- 8 Bits
- Pas de Parité
- Pas de « flow control »

Si nous souhaitons par la suite modifier le débit de la communication UART, il est nécessaire de l'effectuer via le registre CONFIG.

Une fois la configuration modifiée, enregistrer le tout dans la flash.

Les broches du 78M6610 doivent être configurées de la manière suivante :

- **IFC0 = 1**
- **IFC1 = 0**

Configuration de l'adressage du Maxim :

Comme la communication ne s'effectue pas en multipoints (un seul Maxim sur le réseau), les broches ADDR0 et ADDR1 doivent être selon la configuration suivante :

- **MP6/ADDR1 = 0**
- **SCK/ADDR0 = 0**

Paquet UART de communication pour le maître (Panstamp):

Le paquet nécessaire à la réalisation de la communication entre le Panstamp et le 78M6610+LMU est composé des trames suivantes :

- Header = 0xAA
- Byte Count = nombre d'octets total du paquet
- Payload contenant le registre pointé et la commande
- $CHECK_{SUM} = (256 - \sum Bits) \% 256$

Paquet UART de l'esclave (Maxim 78M6610) :

Voici la structure de la trame émise par le 78M6610 (page 56) :

ACKNOWLEDGE without data

ACKNOWLEDGE with data	BYTE COUNT	READ DATA	CHECK SUM
--------------------------	---------------	--------------	--------------

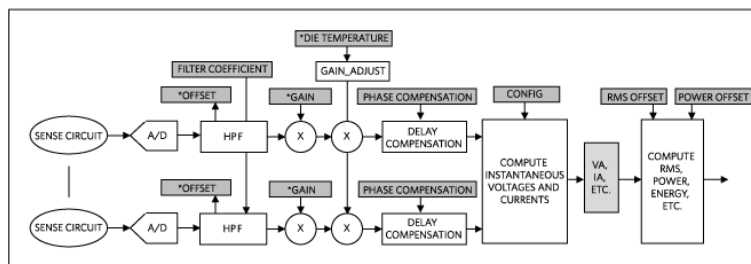
Ainsi que les codes entêtes associés :

Reply Code	Definition
0xAA	Acknowledge with data
0xAB	Acknowledge with data (half duplex)
0xAD	Acknowledge without data.
0xB0	Negative Acknowledge (NACK).
0xBC	Command not implemented.
0xBD	Checksum failed.
0xBF	Buffer overflow (or packet too long).
- timeout -	Any condition too difficult to handle with a reply.

Configuration des registres avant calibration

Afin d'obtenir une mesure cohérente de la puissance consommée par la source en aval, il est nécessaire d'étalonner le Maxim 78M6610+LMU via des registres de commandes.

Le Maxim 78M6610+LMU contient une routine interne permettant de simplifier la configuration de celui-ci. Nous allons donc passer par cette routine afin de simplifier les erreurs et le temps consacré à cette démarche. Cependant, il semble qu'il soit nécessaire de passer par une pré-phase de configuration de certains registres. Ainsi, cette partie va être découpée en deux. Etudions dans un premier temps la valeur des registres avant calibration puis, les registres nécessaires pendant la configuration.



Remarque : Avant de commencer cette partie, il est intéressant de récupérer la valeur de l'ensemble des registres usine car une modification de certains registres peut compromettre le fonctionnement du 78M6610. Ceci a été réalisé à l'aide de programmes (présent sur le Twiki).

La première étape consiste en la configuration des voix A0 (tension) et A2, A3 pour le courant. Ainsi, la tension est multiplexée sous le nom S0 et le courant sous le nom S1.

Configuration de référence de tension (voir page 44):

- **CONFIG[23 :22] = 00**

Pré-amplification:

- **CONFIG[21 :20] = 00**

Configuration de l'entrée de tension S0 (voir page 20) :

Comme nous n'avons qu'une tension directe et non inversée, nous devons multiplier S0 par 1. **CONFIG[19 :8] (0x25)** vaut donc :

CONFIG BITS	19:18	17 :16	15 :14	13 :12	11 :10	9 :8
Value	01	01	01	01	01	10

Configuration de l'entrée de courant S1 (voir page 23) :

Comme nous n'avons qu'une tension directe et non inversée, nous devons multiplier S0 par 1. **CONFIG[7 : 0]** (0x25) vaut donc :

CONFIG BITS	7 : 6	5 : 4	3 : 2	1 : 0
Value	01	01	01	10

Donc **CONFIG = 0x055656**

Configuration des échelles de mesure :

Lien : <http://www.maximintegrated.com/en/app-notes/index.mvp/id/5757>

Configuration de VFSCALE (0x189):

Au maximum, on a 250 mVpk aux bornes de S0 (soit la pleine échelle). Ainsi, au maximum, d'après le point diviseur effectué :

$$V(\text{line}) = 2667.66 \times V(S0).$$

On en conclut donc que :

$$FSV = 0.250 \times 2667.66 = 666.92 \text{ V}$$

Pour respecter les termes, la tension réseau sera donc de 666.92V. On en conclut :

$$\mathbf{VFSCALE = 10\#667 = 0x00029B}$$

Configuration de IFSCALE (0x18C) :

En prenant le même raisonnement que pour IFSCALE. On en déduit :

$$FSI = V(S1) / R_{\text{shunt}} = 0.250 / 0.04 = 62.5 \text{ Apk}$$

Donc :

$$\mathbf{IFSCALE = 10\#63 = 0x00003F}$$

Sauvegarde des informations dans la flash :

Utilisation du registre **COMMAND (0x00)** pour sauvegarder les informations non volatiles en flash.

- **COMMAND = 0xACC200**

Configuration des registres point de référence

La calibration doit se faire en présence **d'une tension AC et une charge stable** dans le temps. De plus, la **tension**, le **courant** et la **température** doivent être connus afin de remplir les registres **VTARGET** (0x72), **ITARGET** (0xB4), **T_TARGET** (0x13E) de leurs valeurs.

- $VTARGET = V_{refrms} / VFSCALE$
- $ITARGET = I_{refrms} / IFSCALE$
- T_TARGET

Exemple1 :

- $V_{refrms} = 230 \text{ V}$
- $I_{refrms} = 20 \text{ A}$
- Température ambiante = 25°C

Donc :

- $VTARGET = 230 / 667 = 0.3448 = \mathbf{0x2C2268}$
- $ITARGET = 20 / 62.5 = 0.32 = \mathbf{0x28F5C2}$
- $T_TARGET = 25 = \mathbf{0x006400}$

Exemple2 :

- $V_{redrms} = 230 \text{ V}$
- $ITARGET = 10 \text{ A}$
- Température ambiante = 22 °C

Donc :

- $VTARGET = 230/667 = 0.3448 = 0x2C2268$
- $ITARGET = 10/62.5 = 0.16 = 0x147AE1$
- $T_TARGET = 22^\circ\text{C} = 0x005800$

Sauvegarde des informations dans la flash :

Utilisation du registre **COMMAND (0x00)** pour sauvegarder les informations non volatiles en flash.

- **COMMAND = 0xACC200**

Il faut alors placer la broche RESET du Maxim à la masse afin que les valeurs soient prises en compte.

Lancement de la calibration

Une fois l'ensemble des registres enregistré, on peut lancer la routine de calibration. Celle-ci se fait par la commande suivante :

- **COMMAND (0x00) = 0xCA2C20** (Calibration du Gain - voir page 44)
- **COMMAND (0x00) = 0xCA2A20** (Calibration de l'offset - voir page 44)

Si tout se passe correctement, le registre COMMAND doit passer à 0x0000XX. Sinon, les erreurs sont sur les bits à 1.

Ainsi, de manière chronologique, nous devons envoyer les trames suivantes pour la configuration du Maxim :

Registre										
CONFIG	AA	A0	A3	6F	00	D3	56	56	05	B6
VFSCALE	AA	0A	A3	89	01	D3	9B	02	00	AF
IFSCALE	AA	0A	A3	8C	01	D3	3F	00	00	0A
COMMAND	AA	0A	A3	00	00	D3	20	22	CA	CA
VTARGET (230V)	AA	0A	A3	72	00	D3	68	22	2C	AE
ITARGET (10A)	AA	0A	A3	B4	00	D3	E1	7A	14	B3
T_TARGET	AA	0A	A3	3E	00	D3	00	58	00	40
COMMAND (Gain)	AA	09	A1	00	00	D3	20	2C	CA	C0
COMMAND (Offset)	AA	0A	A3	00	00	D3	20	2A	CA	CC
READ COMMAND	AA	07	A3	00	00	E3	C9			
COMMAND	AA	0A	A3	00	00	D3	20	22	CA	

Mesures des valeurs utiles et mise en forme

Les registres de tension, courant, puissance nous donnent généralement un ratio de la valeur réelle. Il faut donc multiplier la valeur obtenue par un coefficient pour obtenir la valeur réelle. Notre tuteur industriel souhaite accéder aux valeurs suivantes :

- Tension instantanée aux bornes de la charge
- Courant instantané aux bornes de la charge
- Puissance active
- Puissance réactive
- Puissance apparente
- Facteur de puissance
- Température du composant Maxim

Afin d'accéder aux valeurs réelles, il est donc nécessaire d'effectuer les opérations suivantes :

Mise à l'échelle de la tension instantanée VA_RMS (0x81) :

$$Vr = \frac{VA_RMS \times FSV}{2^{23}}$$

Mise à l'échelle du courant instantané IA_RMS (0xBA) :

$$Ir = \frac{IA_RMS \times FSI}{2^{23}}$$

Mise à l'échelle de la puissance active WATT_A (0xE1) :

$$Pa = \frac{WATT_A \times FSV \times FSI}{2^{23}}$$

Mise à l'échelle de la puissance réactive VAR_A (0xF3) :

$$Qa = \frac{VAR_A \times FSV \times FSI}{2^{23}}$$

Mise à l'échelle de la puissance apparente VA_A (0xEA) :

$$Sa = \frac{VA_A \times FSV \times FSI}{2^{23}}$$

Mise à l'échelle du facteur de puissance PFA (0x12F) :

$$Vr = \frac{PFA \times FSV \times FSI}{2^{23}}$$

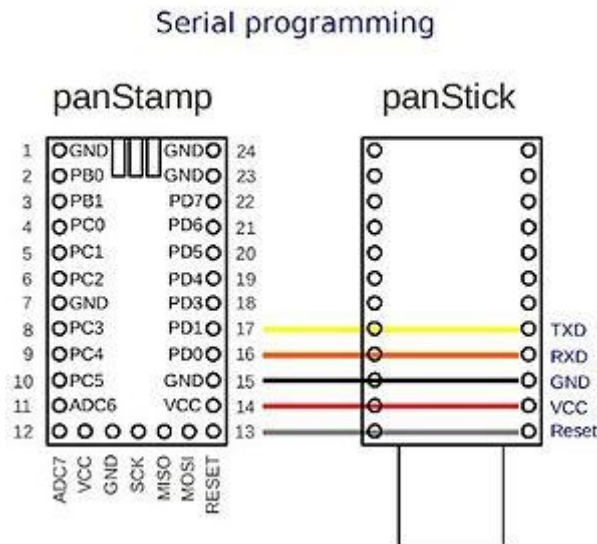
Mise à l'échelle de la température TEMPC (0x13B) :

$$TempChip = \frac{TEMPC}{2^{10}}$$

Il en résulte le programme en C présent [sur le drive](#).

l)-Programmation et première approche

Afin de programmer nos « Panstamps », nous avons câblé une communication série à l'aide d'un adaptateur USB FTDI. Nous pouvons notamment utiliser un Panstick pour réaliser ce processus. Les connections à réaliser sont visibles sur l'image ci-dessous.



Pour émettre ou recevoir une communication RF nous devons d'abord équiper notre Panstamp d'une antenne ou d'un connecteur SMA (type de connecteur coaxial dont l'impédance caractéristique est de 50 Ohms). Pour notre application nous avons choisi d'utiliser la solution par antenne car celle-ci est moins couteuse. L'antenne peut être réalisée à l'aide d'un simple fil, la longueur de notre fil permettra de définir notre fréquence porteuse donc notre longueur d'onde. Les fréquences indiquées sur le site de « Panstamp » sont les suivantes :

Frequency = 868 MHz

Quarter Wavelength: 82.2 mm

Half Wavelength: 164.3 mm

Frequency = 915 MHz

Quarter Wavelength: 77.9 mm

Half Wavelength: 155.9 mm

Nous avons choisi d'utiliser la fréquence de 868 Mhz car par défaut le Panstamp émet à cette fréquence. Ceci nous évitera de paramétrer la fréquence porteuse dans nos programmes.

I.1)Le programme MODEM :

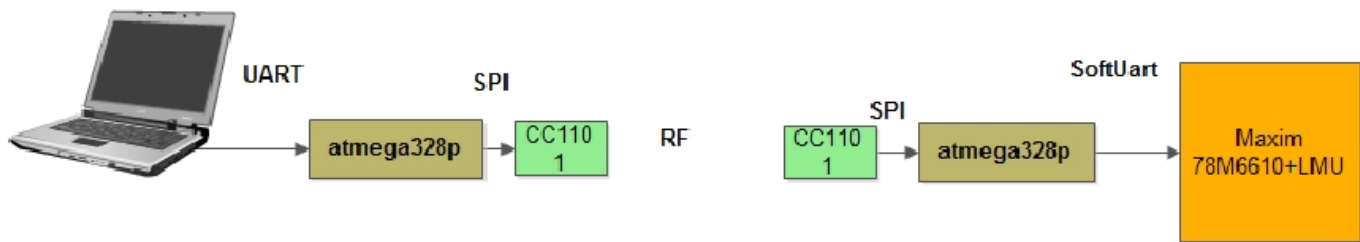
Dans un premier temps, afin de nous familiariser avec le développement de communication RF sur « Panstamp », nous avons commencé à étudier le programme « modem » fourni dans la bibliothèque « Panstamp ». Le programme joue le rôle de passerelle entre le réseau de communication RF utilisé par les « Panstamps » et le port série de l'ordinateur ou le « Panstamp » est connecté. Le programme possède 2 modes de fonctionnement qui sont :

- Le mode data (permet d'écouter le réseau RF et de convertir en ASCII les paquets entrant et de les afficher sur un terminal)
- Le mode commande (permet de configurer cette « passerelle » par le biais de commandes AT sans prendre en compte la réception/transmission des paquets).

I.2)-Etude des bibliothèques :

Pour développer nos propres communications RF, nous avons étudié attentivement les différentes bibliothèques fournies sur le site de panstamp (bibliothèque « Arduino », bibliothèque « Panstamp », bibliothèque swap). Nous avons pu différencier les rôles de chacune.

- La partie « Arduino » comme nous avons vu lors de nos précédents projets, permet de contrôler les I/O (analogique et digital), les ADC, les PWM etc...
- La partie « Panstamp » permet de contrôler les communications RF (transmission/réception), la synchronisation, la gestion de l'alimentation (mode veille/mode transmission)... On notera que la partie « Panstamp » permet de réaliser des communications RF simples mais elle ne possède pas de protocole de communication.

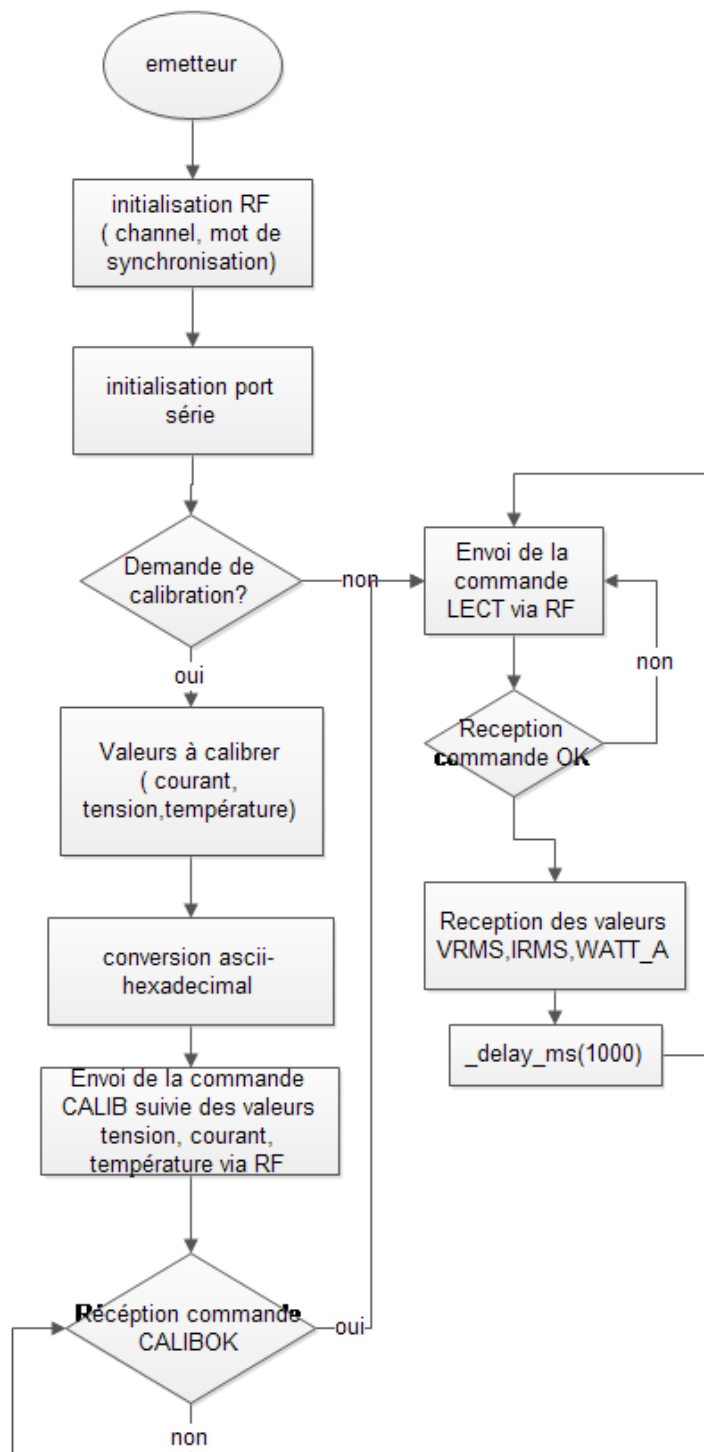


Afin de pouvoir créer un programme beaucoup plus complexe et intelligent. Nous avons retranscrit les fichiers sources C++ de la librairie « Panstamp » en langage C. Ceci représente 23 fichiers dont 9 fichiers .c et 14 fichiers .h.

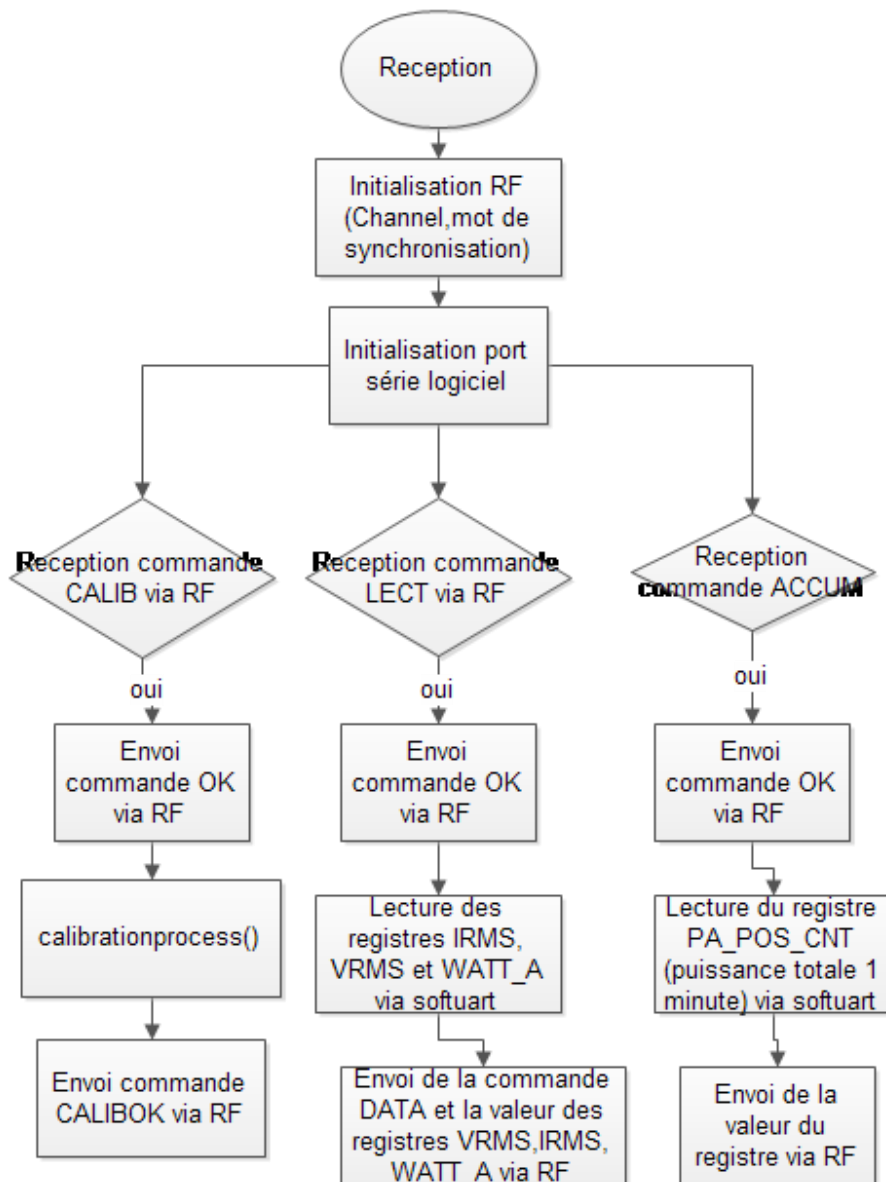
En termes de chiffres la retranscription de la librairie représente environ 3000 lignes de code à retranscrire ou à vérifier. Pour cette réalisation l'utilisation du manuel de la librairie « avr-libc » a été un élément essentiel. Nous avons ensuite implanté les fonctions nécessaires pour une communication série entre notre « Panstamp » émetteur et notre ordinateur et une communication série logiciel entre notre « Panstamp » récepteur et le composant MAX 78M6610+LMU.

Pour réaliser cette « chaîne de communication », nous avons été contraints de reconfigurer le registre BAUD du Maxim à 9600 bauds car l'utilisation d'un port série logiciel ne nous permettait pas d'établir une communication à 38 400 bauds (valeur d'usine du maxim).

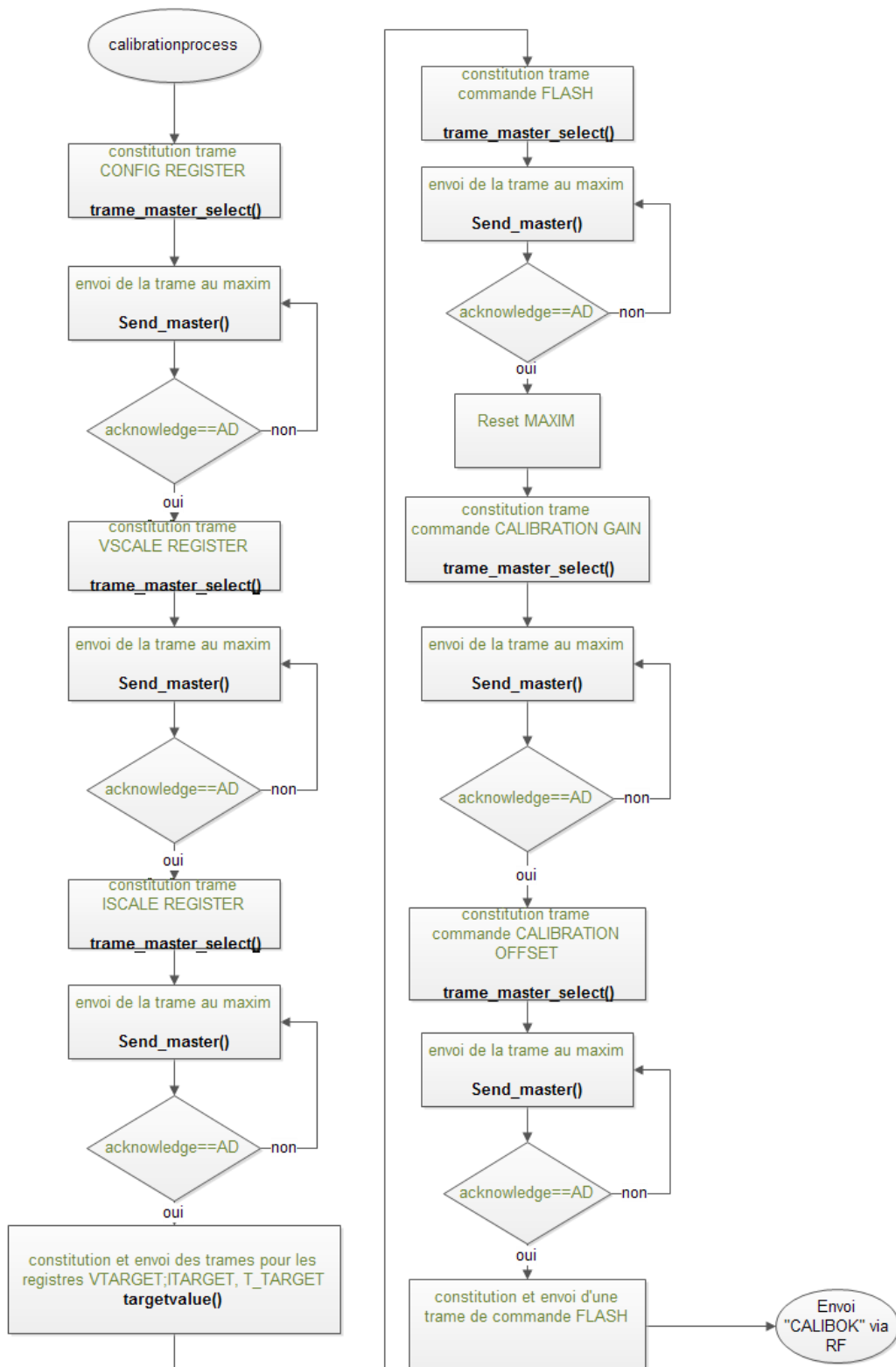
Comportement émetteur:



Comportement récepteur:



Processus de calibration:



Conclusion

Ce projet de fin d'étude est pour nous un réel défi qui nous tiens à cœur de relever en le menant à terme avant février autrement dit en réalisant sa mise en fonctionnement et ce, dans les délais qui nous sont impartis.

Ce projet ambitieux qui nous a été confié nous permet d'accroître considérablement les connaissances acquises jusqu'alors dans le cursus universitaire et ce, principalement dans le domaine des systèmes embarqués tout en prenant en considération les contraintes liées au cahier des charges et des normes de sécurité inhérentes à Polytech'Lille. De plus, sur le plan humain ce fut pour nous, une expérience très enrichissante car, nous avons dû partager et comprendre nos idées mais aussi prendre en compte les contraintes et impératifs définies afin de réaliser ce système. Autrement dit, ce projet d'étude est un travail d'équipe.

De plus, nous pensons avoir amélioré notre autonomie et méthodologie de par les recherches et démarches personnelles effectuées tout au long de ce projet.

En conclusion, ce début de projet nous a permis de réaliser une véritable étude de faisabilité à travers une réflexion autour des choix des composants, une étude technique mais aussi une étude de circuit.

ANNEXES

Liste des composants :

Composant	Réf schématique	Valeur	Quantité	Prix unitaire	Coût total
Résistance de Shunt	R26	0.004 ohm	3	0,546	1,638
Boitier			1	21,11	21,11
Microrupteurs de sécurité			1	6,88	6,88
Bornier de puissance			2	0,84	1,68
Alimentation AC/DC 220/3.3V			1	6,26	6,26
Résistances	R9, R10	1 M	6	0,9	5,4
Résistances	R75, R71, R38, R39, R36, R33, R14, R3	10 K	24	0,285	6,84
Résistances	R11, R24, R28,	750 ohm	9	0,285	2,565
Résistances	R79, R80, R78, R77	1 K	12	0,19	2,28
Résistances	R7, R8, R6, R5, R4, R3, R2, R1	100 ohm	24	0,242	5,808
DEL Rouge	D8, D7, D6, D5, D4, D3, D2, D1		16	0,355	5,68
Switch	S1, S4, S5		6	1,12	6,72
Capacités	C9, C10, C13	1000 nF	9	0,977	8,793

Capacités	C7, C6, C11	0.1 uF	9	0,744	6,696
Capacités	C18, C19	18 pF	6	0,078	0,468
Capacité d'alimentation			2	0,209	0,418
Quartz 20 MHz	Y1	20 MHz	2	0,355	0,71
Barrettes 12 pôles			10	0,432	4,32
Barrettes 5 pôles			10	0,42	4,2
Entretoises plastique			10	0,186	1,86
Visses M3			4	0,164	0,656
Fusible 250V 63 mA			5	2,19	10,95
Ecrou			1	3,73	3,73
Rallonge M/F			1	9,52	9,52
				125,182	€

Développement d'un programme de communication RF :

Afin de prendre en main la communication entre deux panstamps, nous avons tout d'abord développé un programme sans l'utilisation du protocole SWAP et sur un environnement Arduino. Le fonctionnement de ce programme de test est le suivant:

Le Panstamp N°1 transmet un paquet au Panstamp N°2. Lors de la transmission du paquet, le Panstamp N°1 fait clignoter une LED verte et affiche le paquet transmit sur le port série. Le Panstamp N°2 écoute le réseau RF, il possède une LED rouge qu'il va allumer jusqu'à la réception d'un paquet. Quand le Panstamp N°2 reçoit un paquet, il éteint la LED rouge et affiche sur le port série le paquet reçu.

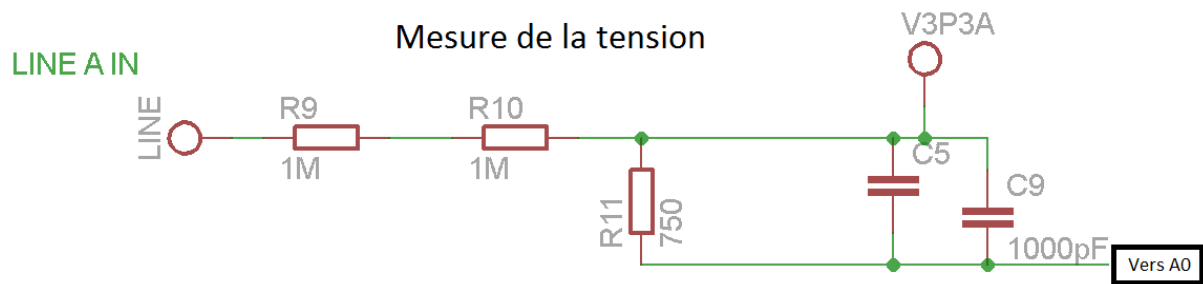
Programme source du transmetteur :

```
7 #include "HardwareSerial.h"
8
9 #define RFCHANNEL 0 // utilisation du channel 0
10 #define SYNCHORD1 0x05 // mot de synchronisation, octet haut
11 #define SYNCHORD0 0x07 // mot de synchronisation, low byte
12 #define SOURCE_ADDR 6 // adresse du panstamp emetteur
13 #define DESTINATION_ADDR 5 // adresse du panstamp qui recoit
14 #define LED 3 //led pin B
15
16
17 CCPACKET packet; // objet paquet
18 byte i;
19
20 void setup() {
21
22
23 // Setup LED output pin
24 pinMode(3, OUTPUT);
25 digitalWrite(3, LOW);
26
27 panstamp.radio.setChannel(RFCHANNEL); //configure le channel
28 panstamp.radio.setSynchord(SYNCHORD1, SYNCHORD0); // configure le mot de synchronisation RF
29 panstamp.radio.setDevAddress(SOURCE_ADDR); //configure adresse du panstamp
30 panstamp.radio.setCcregs(); // permet de prendre en compte les configurations ci-dessus
31 panstamp.setHighTxPower(); // permet de transmettre en puissance haute
32
33 packet.length = 10; // defini la longueur des paquets
34 packet.data[0] = DESTINATION_ADDR; // premier paquet de la transmission correspond à l'adresse destinataire
35
36 for(i=1; i<packet.length; i++)
37 packet.data[i] = i; // remplissage des données
38
39
40 Serial.begin(9600); // initialisation du port série à 9600bauds
41 }
42
43 void loop() {
44 digitalWrite(3, HIGH);
45 delay(1000); // delay ajouté pour visualiser la LED
46 panstamp.radio.sendData(packet); // transmission de la trame
47 digitalWrite(3, LOW);
48 delay(100);
49 for(i=0; i<packet.length; i++)
50 Serial.print(packet.data[i]); // affichage du paquet transmit
51
52 Serial.println(" ");
53 delay(100);
54 panstamp.sleepSec(5); // entre en mode veille pendant 5 secondes (courant consommé 1uA)
55 }
56
```

Programme source du récepteur :

```
8 #include "HardwareSerial.h"
9 #include <avr/wdt.h>
10 #include "TimerOne.h"
11
12 #define RFCHANNEL      0 // utilisation du channel 0
13 #define SYNCWORD1     0xB5 // mot de synchronisation, octet haut
14 #define SYNCWORD0     0x47 // mot de synchronisation, low byte
15 #define SOURCE_ADDR   5 // adresse du panstamp
16
17
18 CCPACKET txPacket; // packet object
19 byte i;
20
21 void rfPacketReceived(CCPACKET *packet)
22 {
23
24     digitalWrite(3, LOW); // paquet reçu donc la led s'éteint
25
26     if (packet->length > 1)
27     {
28         Serial.print("le paquet reçu est: ");
29
30         for(i=0 ; i< packet->length ; i++) // affichage du paquet reçu sur le port série
31             Serial.print(packet->data[i]);
32
33         Serial.println(" ");
34         delay(100);
35
36     }
37     delay(1000);
38 }
39
40
41 void setup() {
42
43     pinMode(3, OUTPUT);
44     digitalWrite(3, HIGH);
45     panstamp.radio.setChannel(RFCHANNEL); //configure le channel
46     panstamp.radio.setSyncWord(SYNCWORD1, SYNCWORD0); // configure le mot de synchronisation RF
47     panstamp.radio.setDevAddress(SOURCE_ADDR); //configure adresse du panstamp
48     panstamp.radio.setCCregs(); // permet de prendre en compte les configurations ci-dessus
49
50     panstamp.radio.enableAddressCheck(); //permet de recevoir seulement les paquets qui
51     // correspondent à l'adresse source et l'adresse de broadcast
52     panstamp.setPacketRxCallback(rfPacketReceived); //permet d'appeler la fonction rfPacketReceived
53     //lors de la réception d'un paquet
54     Serial.begin(9600); // initialisation du port série à 9600bauds
55 }
56
57
58 void loop() {
59     digitalWrite(3, HIGH);
60 }
61
```

Filtrage des entrées analogiques



Les résistances série R9, R10, R11 limitent le courant et réalisent un diviseur de tension sur l'entrée tel que :

$$U(R11) = \frac{R11}{R11 + R10 + R9} \cdot Vline$$

Aussi, les capacités C5 et C9 permettent de réaliser un filtre passe-bas.

$$H(jw) = \frac{R}{1 + jRCw}$$

$$H(jw) = \frac{R}{1 + j \cdot \frac{w}{W_0}}$$

Avec :

$$W_0 = \frac{1}{RC}$$

Si on considère le circuit d'adaptation dans sa globalité :

$$G(jw) = \frac{Z_{eq}}{R_{eq} + Z_{eq}}$$

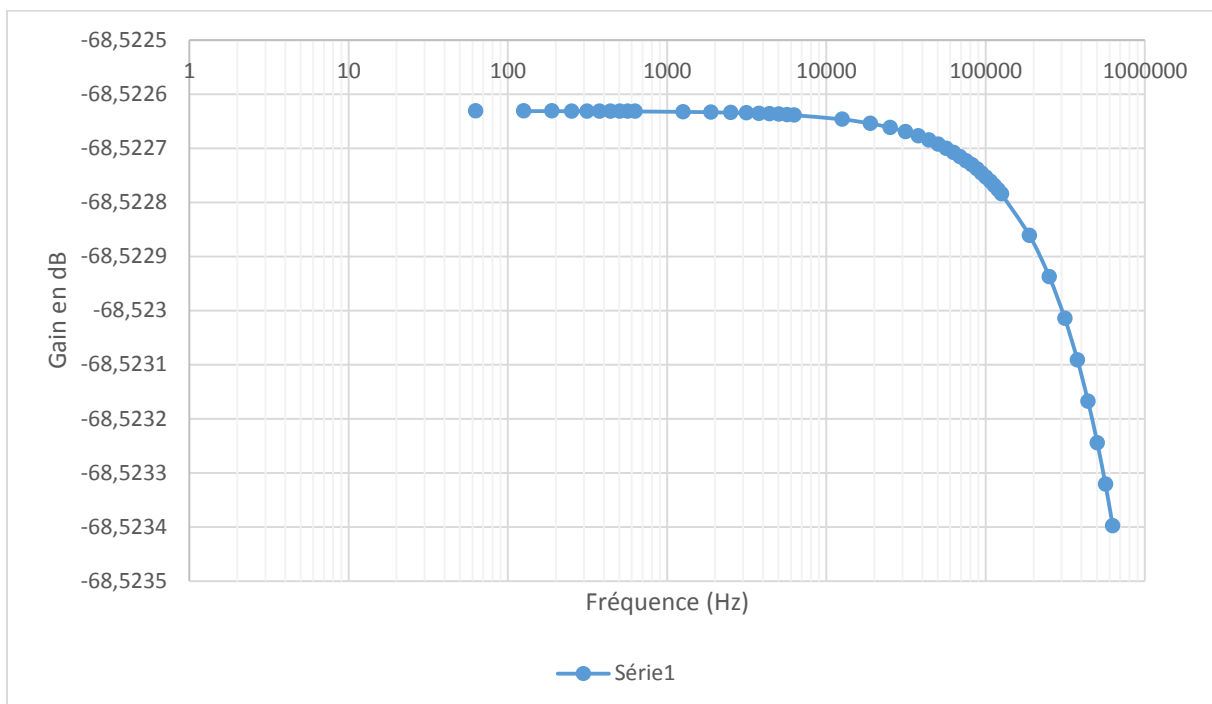
$$G(jw) = \frac{\frac{R}{1 + j\frac{w}{W_0}}}{R_{eq} + \left(\frac{R}{1 + j\frac{w}{W_0}}\right)}$$

$$G(jw) = \frac{1}{\left(1 + \frac{R_{eq}}{R}\right) + j \cdot R_{eq} \cdot C \cdot w}$$

$$G(jw) = \frac{1}{1 + \frac{R_{eq}}{R} + j \cdot \frac{w}{W_0}}$$

Avec $W_0 = \frac{1}{R_{eq} \cdot C}$

PS : $R_{eq} = R_9 + R_{10} = 2 \text{ M}\Omega$. On obtient alors la réponse fréquentielle suivante :



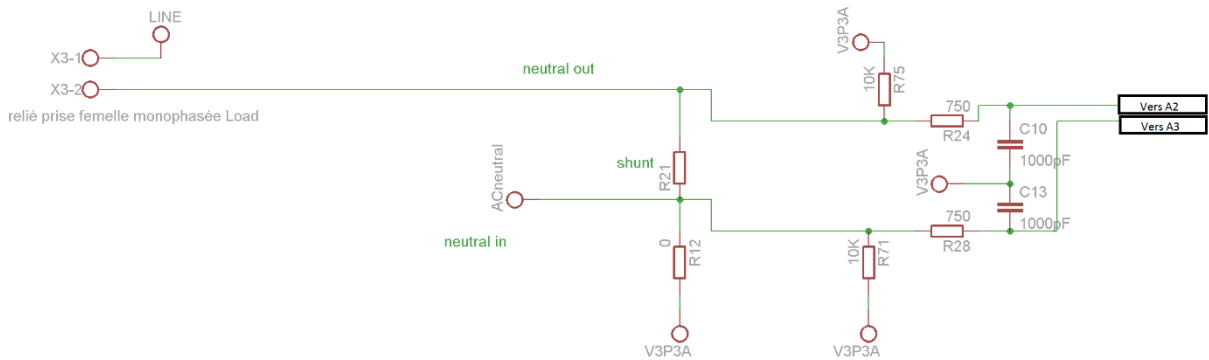
- Lorsque $w \ll W_0$

$$|H(jw)| = \left| \frac{V_s}{V_e} \right| = 20 \cdot \log(1) - 20 \cdot \log\left(1 + \frac{R_{eq}}{R}\right) \cong -68 \text{ dB}$$

- Lorsque $w \gg W_0$

$$|H(jw)| = \left| \frac{V_s}{V_e} \right| \cong -20 \cdot \log(w)$$

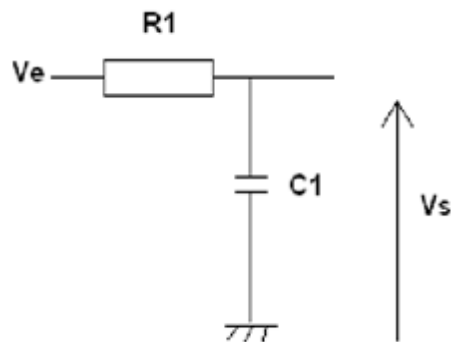
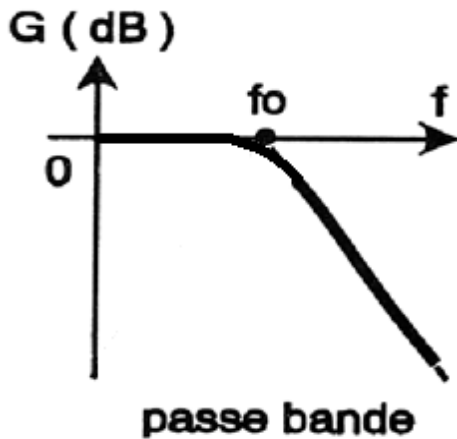
Etude de la fonction « Mesure de courant »



Ici, on mesure la différence de potentiel présente sur la résistance de shunt. Celle-ci nous donne une image du courant circulant dans celle-ci.

$$I(t) = \frac{U(t)}{R}$$

Lorsque la tension aux bornes du shunt est positive, c'est la sortie A2 qui mesure cette différence de potentiel, dans le cas contraire, c'est la sortie A3 qui relève la mesure. Comme pour la mesure de la tension, V3P3A sert de référence de mesure.



Les capacités présentes aux bornes A2 et A3 du composant Maxim réalise un filtrage des hautes fréquences.