

CERN
Organisation Européenne pour
la Recherche Nucléaire
F-01631 CERN Cedex
France
<http://www.cern.ch/>



Kévin Raymond
IUT Anecy
Département GEII
Année 2007



RAPPORT DE STAGE

Mesure d'une température
avec précision et transmission
par liaison sans fil

Maîtres de stage
JAVIER SERRANO (Chef de section)
PHILIPPE NOUCHI

Enseignants
JEAN-LOUIS JORDA
MARCEL PEYRET

Remerciements

Tout d'abord, j'exprime ma profonde gratitude envers Javier SERRANO pour m'avoir pris en stage, ainsi qu'à Philippe NOUCHI sans qui il aurait été plus difficile d'entrer au CERN. Bien sûr, je les remercie également pour leurs nombreux conseils. Ils m'ont permis d'organiser mon travail d'un point de vue professionnel.

Je me dois également de remercier Nicolas DE METZ-NOBLAT, un administrateur qui s'est occupé de la réinstallation de mon système et Christine GAYRAUD pour le suivi des différentes commandes.

J'adresse aussi mes remerciements aux autres collègues pour la bonne ambiance et les quelques conseils toujours bienvenus.

Plus particulièrement à Matthieu CATTIN et Olivier BARRIÈRE pour la réalisation du prototype. Merci à Paul KENNERLEY pour sa précieuse aide anglophone.

C'est un réel plaisir de travailler dans une telle équipe.

Note

◇
Ce
projet est
réalisé à base
de Pic 18F2550
mais peut facilement
être adapté pour d'autres
microcontrôleurs. De plus,
ce rapport a été rédigé en L^AT_EX
en accord avec le monde du logiciel
libre. Ce langage étant compilé, il per-
met également une mise en page de qualité
et normalisée. Je vous recommande d'ailleurs
la classe Beamer de L^AT_EX permettant de
réaliser rapidement des présentations
et des transparents de très grandes
qualités. « La connaissance
s'acquiert par l'expérience,
tout le reste n'est
que de l'informa-
tion » - Al-
bert Ein-
stein
◇

Sommaire

| | |
|---|-----------|
| Introduction | 1 |
| 1 Le CERN | 2 |
| 1.1 Présentation | 2 |
| 1.2 Situation actuelle | 3 |
| 1.3 Groupe AB-CO | 4 |
| 1.3.1 Salle de Contrôle | 5 |
| 1.3.2 Section AB-CO-HT | 5 |
| 2 Les outils de développements | 6 |
| 2.1 Schéma et Circuit imprimé | 7 |
| 2.1.1 KiCaD | 7 |
| 2.1.2 gEDA | 8 |
| 2.2 Les Microcontrôleurs | 11 |
| 2.2.1 Les PIC sous Linux | 11 |
| 2.2.2 Piklab | 11 |
| 3 Mesure d'une température | 13 |
| 3.1 Choix d'un capteur | 13 |
| 3.1.1 Capteurs actifs | 13 |
| 3.1.2 Capteurs passifs | 14 |
| 3.2 Pt100 : Circuit conditionneur | 16 |
| 3.2.1 Les circuits intégrés | 16 |
| 3.2.2 Un circuit adapté | 17 |
| 4 Transmission sans fil | 22 |
| 4.1 Choix du protocole | 22 |
| 4.2 Le module Xbee | 22 |
| 5 L'USB | 24 |
| 5.1 Human Interface Device | 24 |
| 5.2 Exploitation | 25 |
| 6 Impressions personnelles | 26 |
| Conclusion | 27 |
| Glossaire | 28 |
| Bibliographie | 29 |

ANNEXES

| | | |
|----------|-----------------------------|-----------|
| A | Tables Pt100 | 1 |
| B | Calcul d'incertitude | 4 |
| | B.1 Données | 4 |
| | B.2 Les étapes | 4 |
| C | Cartes | 10 |
| | C.1 Emetteur | 10 |
| | C.2 Récepteur | 10 |

ARTICLE CONNEXE

Utilisation de gEDA pour la Conception électronique Assistée par Ordinateur.
Utilisation de Piklab pour programmer les PIC sous Linux avec l'ICD2. Utilisation
de la carte Pt100 et Xbee. Page **20** des annexes.

Table des figures

| | | |
|-----|--|----|
| 1.1 | Organisation des accélérateurs | 3 |
| 2.1 | Synoptique de la suite gEDA | 9 |
| 3.1 | Structure interne de l'ADT70 | 17 |
| 3.2 | Source de courant stable | 18 |
| 3.3 | Pont de Wheatstone | 19 |
| A.1 | Table Pt100 | 3 |
| C.1 | Schéma Émetteur | 11 |
| C.2 | Schéma Récepteur | 15 |

Introduction

Généralités

Etudiant en deuxième année à l'Institut Universitaire et Technologique (IUT) d'Annecy en Génie Electrique et Informatique Industrielle (GEII), j'ai été amené à réaliser un stage d'une dizaine de semaines pour découvrir le monde industriel et un domaine d'activités. Je l'ai effectué au sein du CERN du 2 avril au 22 juin 2007. Ce stage a été une opportunité pour moi de travailler dans un centre de recherche. Le but étant non pas le profit mais l'avancée scientifique, cela crée une réelle motivation.

C'est dans ce cadre que j'ai mis en pratique mes connaissances acquises ces dernières années, mais également appris à m'adapter pour optimiser et non plus appliquer pour réaliser.

Cahier des charges

Ne s'agissant pas d'une tâche critique, j'avais pour but de réaliser un projet pouvant facilement être repris pour mieux l'adapter aux besoins du CERN.

Il consiste en deux cartes : l'une mesurant la température avec le plus de précision possible (0,1 °C), la seconde recevant les températures pour les enregistrer sur un ordinateur qui devra les traiter. La carte de mesure de température devant être alimentée par une pile (9V), sa consommation doit être minimale. Ces deux cartes doivent être en liaison sans fil avec une portée raisonnable.

Le choix du capteur est libre, tout en respectant la précision sur une plage mesure de 0 °C à +30 °C. Le protocole utilisé pour la transmission sans fil est libre mais doit être fiable et d'une faible consommation. Pour ce qui est de la réception sur l'ordinateur, le protocole est également libre.

Il doit être possible d'avoir plusieurs satellites (carte de mesure pour la température) à divers endroits émettant les informations vers l'ordinateur.

Une contrainte plus restrictive est d'explorer le monde du logiciel libre. Pour ce qui est de la réalisation de la carte et du développement logiciel, j'ai exploré ce domaine sur une machine GNU/Linux.

Cela m'a également permis de me familiariser avec Linux.

Après avoir présenté le CERN, nous nous intéresserons à mon activité de stagiaire ainsi qu'aux travaux réalisés, et enfin nous ferons le bilan de ce stage avant de conclure sur sa globalité.

1 Le CERN

1.1 Présentation

Le Centre Européen pour la Recherche Nucléaire (CERN) est l'un des plus prestigieux centres de recherche fondamentale au monde.

Créé en 1954 pour rassembler les plus grands chercheurs Européens, il est l'investigateur de grandes technologies telles que le World Wide Web ou la supraconductivité. La Cryogénie est également un domaine dans lequel le CERN innove.

Le CERN est devenu l'Organisation Européenne pour la Recherche Nucléaire. Le but étant de fournir les outils dont les physiciens du monde entier ont besoin. Ceci par des accélérateurs de particules associés à leurs détecteurs, ils analysent les collisions entre les particules élémentaires.

La mise en commun des données à exploiter par les physiciens du monde entier est le point de départ du développement d'Internet. Maintenant, un nouveau système se met en place : la Grille.

« La grille nous permet de combiner la puissance de centres de calculs scientifiques où qu'ils soient, pour fournir la ressource informatique la plus puissante que puisse offrir le monde » indique Ian Bird, chef du déploiement de la grille au CERN. Grâce au réseau de fibres optiques, il n'y aura pas de temps de latence pour les transferts de données entre ordinateurs.

Cette infrastructure permettra de traiter les 14 pétaoctets générés chaque année par le nouvel accélérateur (LHC). Cela équivaut à plus de vingt millions de cédéroms. Il est prévu pour être opérationnel fin 2007. Pour exploiter toutes ces données, il faut une puissance de calcul équivalente à 70 000 ordinateurs personnels, les plus rapides de l'époque actuelle.

Ce laboratoire européen pour la physique des particules s'est ouvert à beaucoup de pays. Ses États membres sont les suivants : Allemagne, Autriche, Belgique, Danemark, Espagne, Finlande, France, Grèce, Hongrie, Italie, Norvège, Pays-Bas, Pologne, Portugal, la République slovaque, la République tchèque, Royaume-Uni, Suède et Suisse. La Fédération de Russie, Israël, la Turquie, la Yougoslavie (le statut d'observateur est suspendu après l'embargo de l'ONU, juin 1992), la Commission des Communautés européennes et l'UNESCO ont le statut d'observateur.



1.2 Situation actuelle

Pour accélérer les particules, il existe des accélérateurs linéaires et circulaires. À technologie égale, plus ils sont longs, plus ils sont puissants. C'est pourquoi le CERN dispose de quatre accélérateurs circulaires et quelques accélérateurs linéaires. Ces derniers servent à lancer les particules dans un premier accélérateur de particules circulaire, pour que sa vitesse et son énergie augmentent. Le faisceau de particules effectue quelques tours, de plus en plus rapidement. Une fois atteint l'énergie limite de l'accélérateur, il est injecté dans l'accélérateur suivant. Ainsi, en cascasant plusieurs accélérateurs, le faisceau pourra atteindre une énergie de 7 TeV¹ à 0,999 999 991 fois la vitesse de la lumière ! En accélérant deux faisceaux à sens contraires, une énergie de 14 TeV sera donc disponible.

Il y a principalement trois accélérateurs circulaires :

Le PS pour Proton Synchrotron amène les particules à 25 GeV. Ensuite le SPS (Super Proton Synchrotron) produit une énergie de 450 GeV. Pour enfin aboutir au LHC et ses 7 TeV. Les particules accélérées peuvent être des protons et des anti-protons. Mais aussi des ions de plomb. À l'heure actuelle, l'énergie maximale atteinte dans un accélérateur de particules est de 2 TeV (collisionneur proton anti-proton installé au laboratoire Fermilab près de Chicago).

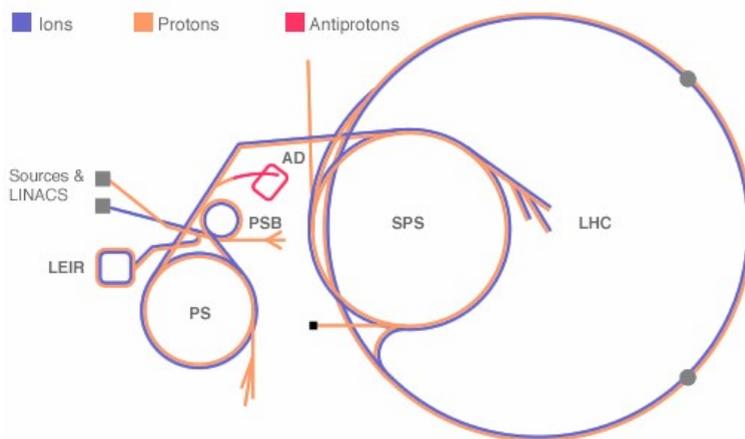


FIG. 1.1 – Organisation des accélérateurs

L'ancien plus grand accélérateur du CERN, le LEP a été entièrement démonté. Actuellement, le montage du nouveau est bientôt achevé. Il s'agit du LHC, pour Large Hadron Collider, l'entreprise scientifique la plus ambitieuse que le monde n'ait jamais connue. Il va permettre des collisions frontales de faisceaux de protons à une énergie de 14 TeV (contre seulement 200 GeV pour le LEP et ses faisceaux

¹ 1 Électron-Volt correspond à $1,602\ 176\ 53 \times 10^{-19}$ Joules



d'électrons). En effet, la masse d'un proton est presque deux-mille fois plus élevée que celle d'un électron. Ce qui augmente considérablement son énergie. Il est composé de 1 200 aimants bipolaires supraconducteurs de 15 m de long, produisant un champ magnétique de 8,4 Teslas. S'y ajoute 500 aimants quadripolaires supraconducteurs dont le but est de refocaliser les faisceaux pour mieux contrôler les collisions.

Les collisions sont programmées pour être produites à des zones spécifiques dans l'accélérateur. Là sont situés les différents détecteurs, organisés par expériences. Tous sont uniques, ils sont conçus dans le but de satisfaire les exigences expérimentales. Par exemple, le puits ATLAS sur le LHC est composé d'un détecteur de 22 m de haut, de 44 m de long pour un poids de 7 000 tonnes. Celui-ci enfoui à près de 100 m de profondeur, au niveau de l'accélérateur.

Pas moins de 7 000 scientifiques d'universités et de laboratoires du monde entier s'en serviront pour apporter des réponses aux questions les plus fondamentales de la science.

Pour augmenter ainsi la puissance tout en gardant l'anneau d'une même circonférence (27 km) il leur faut plonger les aimants dans un bain d'hélium liquide. De cette manière ils atteignent des températures de refroidissement de l'ordre de 1.9 K (soit environ -271 °C). C'est ainsi que le secteur 7-8 (un huitième de l'accélérateur) a atteint une température inférieure à celle de l'espace intersidéral le 10 avril 2007. C'est la plus grande installation supraconductrice du monde. De plus, tous les supraconducteurs sont plongés dans de l'azote liquide.

Le CERN, avec plus de 3 000 employés et près du double l'été, est une très grande organisation. Il est Divisé en plusieurs départements, les différents domaines d'activités sont regroupés par expériences (CMS², ATLAS . . .) ou par compétences. Par exemple HR³ ou AB⁴.

Mon stage s'est déroulé chez AB-CO-HT (Accelerator and Beams Control – Hardware and Timing).

1.3 Groupe AB-CO

Au sein du département AB, le groupe AB-CO s'occupe de la partie contrôle de l'accélérateur. Une nouvelle salle vient d'être inaugurée. Elle centralise les opérations de tous les accélérateurs.

² Compact Muon Solenoid

³ Ressources Humaines

⁴ Faisceaux et Accélérateurs



1.3.1 Salle de Contrôle



Le contrôle du bon déroulement des opérations s'effectue en continu.

Le coût de l'installation du LHC est actuellement de 2 100 Millions d'euros. Toute perte du faisceau engendrerait des dommages irréversibles sur les aimants à coûts élevés et plusieurs mois d'interruption pour réparations. Pour les particules ayant pratiquement la vitesse de la lumière, il faut être très précis pour synchroniser toutes ces opérations. La section AB-CO-HT s'en charge.

1.3.2 Section AB-CO-HT

Le Timing permet de synchroniser toutes les opérations au sein du CERN, pour ainsi être de plus en plus précis. Puisqu'il faut 90 μ s aux particules pour effectuer ces 27 km, les équipements électroniques doivent être très rapides.

Toutes les opérations sont gérées par des horloges atomiques au Césium (ou Bohrium) avec une précision de 10^{-14} secondes. La référence de l'heure UTC (Universal Time Coordinated) est prise par GPS pour la reproduire localement. De cette manière, le CERN reste synchronisé même en perdant le signal GPS. Avec une dérive inférieure à 1 ns par jour.

Cela permet également de lancer des expériences de plus grandes envergures, par exemple avec Rome. En effet, le CERN émet des neutrino en Italie pour une autre expérience en physique des particules. Il est important que les deux sites aient la même heure pour être certains de recevoir les particules émises par le CERN et non venues de l'espace.

L'électronique étant de plus en plus précise, elle devient de plus en plus sensible à la température. Une mesure précise de la température présente donc un avantage pour éviter un certain nombre de perturbations. Notamment la dérive thermique des horloges atomiques qui serait plus faible avec un meilleur contrôle de température.



2 Les outils de développements

Windows associé aux logiciels propriétaires est une bonne solution pour démarrer simplement en CAO (Conception électronique Assistée par Ordinateur) ou la programmation des microcontrôleurs. Hormis le coût des licences, le problème est le bridage de ces logiciels. On les utilise comme ils sont conçus. C'est-à-dire qu'aucune amélioration n'est possible.



La solution est de passer aux logiciels libres qui eux sont gratuits et non bridés. De ce fait, si une fonctionnalité lui manque, il nous est possible de la rajouter ou d'en faire la demande. C'est pourquoi on arrive à trouver des logiciels gratuits et très évolués. Dans ce domaine qui est celui du logiciel libre, il existe de nombreux logiciels pour un électronicien. Tant au niveau CAO que programmation des microcontrôleurs, les possibilités ne cessent d'évoluer.

Logiciel libre se dit d'un logiciel donnant le droit à toute personne en possédant une copie de l'utiliser et de le redistribuer librement, voire de le modifier. On retrouve en général ces logiciels avec une licence « copyleft » (©).

Cependant les logiciels libres les plus appréciés font partie du projet GNU (GNU's Not Unix). Il s'agit en fait de la FSF (Free Software Foundation). Tout logiciel de ce projet est basé sur quatre principales libertés :

- Liberté 1 : La liberté d'exécuter le programme pour tous les usages.
- Liberté 2 : La liberté d'étudier le fonctionnement du programme.
- Liberté 3 : La liberté de redistribuer des copies.
- Liberté 4 : La liberté d'améliorer le programme.

Avec l'accès au code source, il est aisé d'améliorer le programme : c'est ce qui encourage la création d'une communauté de développeurs visant à améliorer le logiciel. Ce mode de distribution encourage également la concurrence, ce qui amène à un développement de plus en plus poussé pour ainsi aboutir à un projet utile et fonctionnel.





2.1 Schéma et Circuit imprimé

Parmi la multitude d'outils Open Source¹ permettant de dessiner un schéma et un circuit imprimé, les plus connus et les plus utilisés sont KiCad et gEDA.

2.1.1 KiCaD

KiCaD est une suite open source (GPL) pour la réalisation de schémas électroniques et de circuits imprimés programmés par Jean-Pierre Charras. GPL signifie General Public License. Cela ressemble fortement au projet GNU (ou GNU GPL).

Cette suite comprend cinq programmes :

- **Eeschema**, l'éditeur de schéma et de symboles,
- **PcbNew**, un outil de dessins de circuits imprimés et de création d'empreintes,
- **Gerbview**, un générateur de fichier GERBER (documents de phototraçages),
- **Cvpcb**, pour choisir les empreintes de chaque composant (ou symbole),
- **Kicad**, le gestionnaire de projets.

KiCaD est une suite très conviviale facile à prendre en main. Un des ses points forts est son aide accessible directement dans la barre d'outils avec de nombreuses captures d'écran. Il existe beaucoup de fichiers de langue dont la principale est le Français. Également disponible en Anglais, Portugais et Espagnol il s'agit d'un outil à la portée de tous.

Ecrits en C++ et basés sur WXWIDGETS² ces différents logiciels sont multi-plateformes, c'est-à-dire qu'ils tournent sur Macintosh, GNU/Linux et Unix, Windows.

Les bibliothèques existantes sont insuffisantes, mais il est aisé d'en créer. De plus il y a la possibilité de générer une représentation de la carte en 3Dimension à l'aide de Wings3D. KiCaD gère le circuit imprimé sur douze couches. Par ailleurs, il existe un moyen d'importer des bibliothèques d'Orcad.

En ce qui concerne son installation sous windows, cela ne devrait pas poser de problème, il existe même une version ne nécessitant pas d'installation (pour les disques amovibles notamment). Pour Linux tout dépend de la distribution. Le plus simple est d'essayer avec les paquets synaptic.

Penser à mettre à jour la liste avant de commencer l'installation (choisir en fonction du système ...) :

```

| $ sudo yum update          || $ sudo apt-get update      |
| $ sudo yum install kicad   || $ sudo apt-get install kicad |

```

¹ Open Source signifie Logiciel Libre

² Logiciel libre aidant à la création d'interface multi-plateformes



Si les paquets nécessaires sont introuvables (ce qui serait étonnant vu la popularité de cette suite logicielle), télécharger les fichiers binaires de la dernière version sur ce site : <ftp://iut-tice.ujf-grenoble.fr/cao/>. Suivre la procédure habituelle (décompresser l'archive puis lancer « kicad » se trouvant dans kicad/linux/). Toute la documentation nécessaire est disponible dans l'archive. À noter qu'il est préférable de l'extraire dans /usr/local/, et d'effectuer un « `chmod -R 644 /usr/local/kicad/-library` » puis « `chmod -R 644 /usr/local/kicad/modules` » pour donner les droits d'accès aux bibliothèques, aux utilisateurs normaux.

Les différentes bibliothèques disponibles le sont sur <http://www.kicadlib.org/>

Eeschema est l'outil qui nous met en confiance avec KiCaD. Sa convivialité assure une prise en main rapide. En première approche ce qui peut perturber est le déplacement des composants. En effet, lorsque l'on déplace un composant, ses connexions ne suivent pas. Il est donc impossible de vérifier une bonne connexion sans exécuter le script nécessaire. Néanmoins cela reste une suite de bonne qualité, utilisée par exemple pour l'enseignement.

Pour plus d'informations :

http://www.lis.inpg.fr/realise_au_lis/kicad/, le site officiel.

http://kicadlib.org/Fichiers/Tutorial_Kicad_FR.pdf, pour un tutoriel très complet.

<http://tech.groups.yahoo.com/group/kicad-users/>, pour le forum.

Le deuxième outil permettant de concevoir des cartes électroniques est gEDA. Il est la référence sous Linux. une véritable communauté de programmeurs y travaille. De plus, il inclut un logiciel de simulation.

2.1.2 gEDA

gEDA pour GNU Electronic Design Automation est une suite de logiciels CAO et de simulation sous Linux, destinée tant au prototypage que pour la production. Les sources sont également distribuées, ce qui en fait un projet très régulièrement mis à jour. En contre-partie il faut noter que beaucoup de tutoriaux sont dépassés. Il faut donc faire la part des choses en préférant lire seulement celles de gschem et Pcb. Lors de la création de la netlist par exemple, des conseils avisés sont donnés.

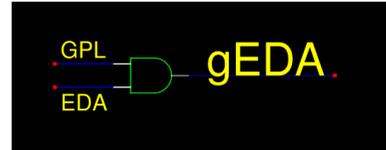
Cette suite d'outils est principalement composée de :

- **gschem**, l'éditeur de schémas et de symboles,
- **gnetlist**, un translateur vers d'autres utilitaires,
- **Pcb**, un outil de dessins de circuits imprimés et de création d'empreintes,
- **ngspice**, un clone de spice avec des fonctions étendues,
- **gnucap**, un simulateur original avec compilateur de modèles,
- **geda**, le gestionnaire de projet (non actuel).



Attention à l'utilisation de geda, le gestionnaire de projet. Il n'a pas suivi l'évolution des autres programmes, il est donc préférable de ne pas l'utiliser. Un terminal suffit pour relier les différents outils.

Comment l'installer ? La dernière version de gEDA nécessite python 2.5 ou supérieur. Il est possible que gEDA ainsi que toutes ses dépendances soient dans les dépôts de votre distribution (donc même étape que pour KiCad, voir page 7, section 2.1.1) ; mais ce ne sera peut-être pas la dernière version. Pour cette suite en constante évolution, il est important d'avoir la version la plus à jour, pour éviter certains bogues.



Une autre possibilité est de télécharger l'image iso du CD. Elle inclut la dernière version de tous les programmes, ainsi que les dépendances nécessaires. Voir dans la partie « download » du site officiel : <http://www.geda.seul.org/>. La solution la plus simple est de monter le cd sur un lecteur virtuel. Très simple sous linux.

```
$ sudo mount -o loop <image>.iso /mnt
```

Pour poursuivre l'installation l'aide nécessaire se trouve sur le CD. En ce qui concerne les tutoriaux, attention à bien lire les dernières versions.

Une fois l'installation terminée, il faut avoir le synoptique de la suite logicielle en tête pour commencer une carte (figure 2.1).

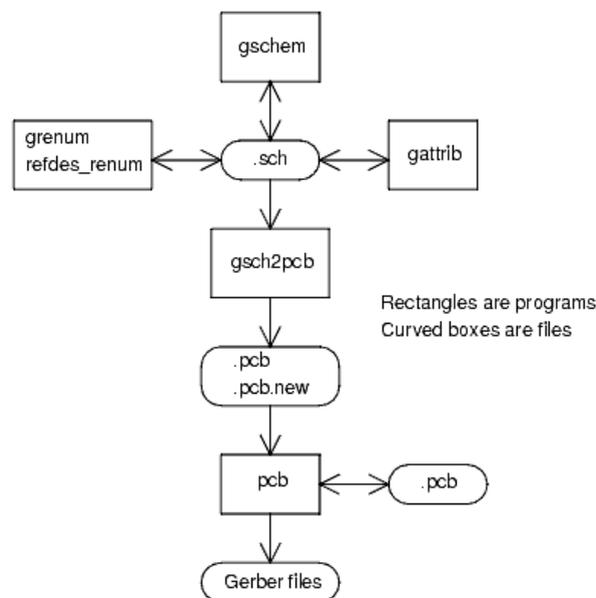


FIG. 2.1 – Synoptique de la suite gEDA

Tout commence avec gschem. Une fois le schéma réalisé et vérifié avec le contrôleur DRC, il faut créer la netlist. Grenum sert à assigner une référence à tous les



composants, ce qui peut être réalisé automatiquement avec `gschem`. De même `gattrib` permet d'assigner des empreintes aux composants, mais il est plus évident au début de le faire sous `gschem`. `gsch2pcb` permet de créer la netlist et le fichier *.pcb. `Pcb` permet de placer les composants et de router les connexions générées par la netlist. Enfin, les fichiers Gerber ou PostScript sont exportés par `Pcb`.

gschem

`gschem` est un logiciel permettant de rapidement créer des schémas une fois passée la phase d'apprentissage. Il existe une multitude de raccourcis clavier.

Une fois `gschem` ouvert (commande « `gschem` »), vous verrez une interface simple d'utilisation. Seulement quelques icônes sont dans la barre des tâches. Possibilité d'ajouter des composants, de créer des fils de connexions voir même des bus de données. Pour tourner les composants ou toutes autres tâches utiles, il est bien plus rapide de se servir des raccourcis clavier.

Le seul inconvénient notable de `gschem` est la fonction « `undo` ». Lorsque l'on annule les dernières opérations, le zoom est pris en compte. Cette fonction n'est pas vraiment pratique ...

Ensuite vient la réalisation du Printed Circuit Board, ou PCB.

Pcb

`Pcb` est un outil très performant mais nécessitant un temps d'adaptation non négligeable. Une fois toutes les règles de routages réglées, la réalisation de PCB est rendue extrêmement rapide.



Cependant nous pouvons déplorer le manque de fonctions dans la barre d'outils. Mais le développement actuel privilégie la performance. Encore une fois, beaucoup de raccourcis clavier sont disponibles.

Avec `Pcb`, d'un clic nous pouvons exporter la carte en fichier Gerber, png, ps, eps ...

Conclusion sur gEDA

Une des plus grandes richesses de `gEDA` est le traitement des données entre les différents programmes. Tous les fichiers sont en ASCII ce qui les rend très facile à manipuler. Il est donc possible de créer des scripts pour automatiser les tâches rébarbatives. Habituellement le Perl est utilisé pour cela.

Une fois en main, `gEDA` est à la fois simple et rapide à utiliser. De plus beaucoup de bibliothèques sont disponibles sur <http://www.gedasymbols.org/>. Il est également possible de demander sur la liste des `geda-user` si ce composant est disponible. Pour



cela voir <http://geda.seul.org/maillinglist/index.html>. À noter qu'il existe deux sortes de librairies : les **Oldlib** (librairies m4) et les **newlib**. L'une adaptée à la création textuelle, l'autre graphique. Se reporter à l'article connexe (page 20) pour plus de détails.

gEDA est donc une suite très complète. Sa principale difficulté de prise en main repose sur le fait que les tutoriaux ne sont pas à jour. Il faut préférer l'aide de Pcb.

Néanmoins cela reste une suite pour les concepteurs ne souhaitant pas se préoccuper d'une license payante et souhaitant un travail de professionnel. Pour obtenir ce résultat il faut bien entendu y passer un peu de temps, mais qui sera largement récompensé par la suite en efficacité. De plus, il y a réellement une grande bibliothèque de composants rapidement utilisables. Cela peut surprendre au début d'enregistrer des empreintes avec l'éditeur de texte, mais rien n'est plus facile que de copier, voire modifier du texte.

Pour de plus amples explications sur l'utilisation et la prise en main de gEDA, lire l'article connexe joint à ce rapport (page 20 des annexes).

Pour continuer dans la même lignée, nous allons maintenant explorer le domaine de la programmation des microcontrôleurs sous Linux.

2.2 Les Microcontrôleurs

Pour réaliser une carte modulable, j'ai décidé de la baser sur un microcontrôleur. Il doit gérer les entrées analogiques pour la sonde et une interface série pour le module sans-fils. Ne connaissant que les PIC de chez MICROCHIP, j'ai choisi le PIC18F2550, qui permet également la liaison Universal Serial Bus (USB).

2.2.1 Les PIC sous Linux

Il existe beaucoup de possibilités. Sur le site suivant <http://forum.microchip.com/tm.aspx?m=123533> plusieurs solutions sont repertoriées.

Actuellement, la meilleure solution est d'utiliser l'éditeur Piklab, le fruit d'une collaboration entre les programmeurs de Pikdev et d'autres logiciels libres..

2.2.2 Piklab

Il s'agit d'un environnement de développement adapté au PIC et dsPIC. Il remplace MPLAB, l'éditeur de Microchip (pour Windows). On lui spécifie le compilateur. Les principaux sont la suite gputils pour l'assembleur, SDCC et C18 pour le C. C18 est le compilateur propriétaire de Microchip. Il tourne sous Wine (outil



permettant d'exécuter des programmes Windows sous Linux). Il y a également Pic C (Lite) pour la programmation en C. Le simulateur disponible est GPSim. Simple mais puissant. Pour ce qui est des programmeurs supportés, c'est en constante évolution. Divers par liaison série, parallèles, USB (ICD2). Le Pickit2 etc. Pour les debugger, l'ICD2 (pour In Circuit Debugger 2) est supporté. Avec plusieurs chargeurs d'amorçage, ou " Boot loader" (Tiny, Pickit2, Picdem). Toute la documentation se trouve sur <http://piklab.sourceforge.net/>.

ICD2 Microchip

J'ai choisi l'In Circuit Debugger 2 de Microchip pour programmer. C'est une solution rapide à mettre en oeuvre, et puissante. La communication par USB ou port série est supportée sous linux. Il faut simplement installer le package pour la reconnaissance USB. **Package disponible ici : <http://kevin.raymond.free.fr/Stage/icd2prog-0.3.0.tar.gz>**.

Il est également possible d'installer MPLAB avec Wine. Il fonctionne à merveille et les messages d'erreurs sont plus explicites que sous windows.

Pour ce qui est du compilateur, j'ai choisi SDCC.

SDCC

Small Device C Compiler est un compilateur C sous linux adapté aux PIC. Il est porté sous Windows et il s'agit comme de Piklab d'un logiciel libre.

Les sources écrites avec SDCC sont donc portables. Beaucoup de PIC sont supportés, voir sur le site officiel la liste des PIC non supportés . <http://sdcc.sourceforge.net/>. L'optimisation du code fonctionne correctement.

Piklab associé à SDCC et l'ICD2 rend la programmation des PIC beaucoup plus facile et conviviale que sous Windows. Malgré que Pilab soit encore en version beta, cet environnement est plus stable que MPLAB.

De plus tout est conçu pour développer rapidement. La console est intégrée, tout les rapports sont à portée de clic et les messages de réponses au debugger sont très clairs. Alors que sous Windows nous sommes simplement informés que la carte ou le debugger sont défectueux, sous cet environnement nous savons que le PIC détecté n'est pas le bon, mais qu'il est programmable (sous réserve d'éventuelles erreurs).

Pour des exemples de programmes réalisés avec SDCC, voir :
http://www.luga.de/Angebote/Vortraege/Linux_USB/Linux_USB.pdf en allemand ;
<http://www.nutsvolts.com/%7Edownloads/USB.zip> un exemple très complet ;
http://swenmac.homeip.net:8080/lab3_sdcc.tgz un autre exemple.

3 Mesure d'une température

3.1 Choix d'un capteur



Depuis l'époque de Galilée et le premier thermomètre (basé sur la dilatation de certains liquides) en 1593, il y a eu d'énormes progrès dans ce domaine. Les capteurs analogiques sont apparus pour remédier à un problème : « Comment interfacer un thermomètre à mercure sur un système automatisé ? »

Cette mesure étant toujours complexe à mettre en oeuvre, il existe plusieurs solutions. La démarche habituelle est d'utiliser un capteur passif basé sur la variation d'une grandeur électrique en fonction du mesurande, la température dans notre cas. La précision et la justesse des mesures sont directement liées au circuit conditionneur associé à de tels capteurs.

Pour une plus grande simplicité de mise en oeuvre il est possible de s'orienter sur les capteurs actifs, permettant de mesurer une quantité électronique.

3.1.1 Capteurs actifs

Souvent sous la forme de circuits intégrés, ils sont appréciés pour leur simplicité de mise en oeuvre et leur faible coût. Ils peuvent dialoguer par liaison série, sur le BUS I2C et bien d'autres protocoles encore.

Parmi les plus utilisés il y a le DS1620 de chez MAXIM qui transmet par liaison série la température sur 9 bits. Sa résolution est de 0,5 °C. Les plages de mesures sont de l'ordre de -10 °C à +80 °C. Il existe beaucoup d'équivalents, par exemple le DS1820.

Il y a également le SMT160-30 de chez SMARTEC qui module le rapport cyclique du signal en sortie selon la température. Il s'agit de PWM (Modulation à Largeur d'Impulsion). Il a une précision absolue de 0,7 °C dans l'intervalle -30 °C à +100 °C. La sortie du capteur, de technologie CMOS, peut être déportée par un câble jusqu'à 20 mètres.

Pour interfacer sur le BUS I2C il y a notamment la sonde DS75 ou DS1621 de chez DALLAS. Le DS75 sur 12 bits permet une résolution de 0,0625 °C pour une précision de $\pm 2,0$ °C. Le DS1621 a une résolution de 0,5 °C. Il en existe également pour une liaison SPI (Serial Peripheral Interface) comme le DS1722 de chez MAXIM, il a les mêmes caractéristiques que le DS75.



Il existe aussi des capteurs intégrés du type LM35 ou LM335 de chez NATIONAL SEMICONDUCTOR. Ce type de capteur délivre un signal qualibré en millivolts par degré. Avec un LM35 on ne peut espérer atteindre une précision supérieure à $0.25\text{ }^{\circ}\text{C}$. Sa sortie est linéaire dans la plage $-40\text{ }^{\circ}\text{C}$ à $+110\text{ }^{\circ}\text{C}$ et vaut $10.0\text{ mV}/^{\circ}\text{C}$. Pour la fidélité, la moyenne des mesures donne $0.5\text{ }^{\circ}\text{C}$ « garanti ».

Les thermocouples font également parti des capteurs actifs. En effet, le principe de mesure est basé sur l'effet Seebeck. Deux fils métalliques de natures différentes sont reliés par deux jonctions. L'effet Seebeck crée une Force Électromotrice liée à la différence de température entre ces deux jonctions.

Les thermocouples sont des sondes pouvant mesurer des températures allant de $-270\text{ }^{\circ}\text{C}$ à $+2320\text{ }^{\circ}\text{C}$.

Leur coût est assez faible sauf pour les températures extrêmes.

Il existe plusieurs types de thermocouples. Ceux à base de métaux usuels, de métaux nobles ou de métaux réfractaires. Il faut les choisir en fonction des gammes de températures ainsi que de leur coefficient de Seebeck. Ce coefficient est directement lié à la sensibilité de la sonde.

De nos jours il n'en existe pas de plus juste. Que ce soit pour la précision, la fidélité ou même de plus grandes plages de mesures, rien n'égale les sondes analogiques.

3.1.2 Capteurs passifs

Il existe deux principaux types de sondes passives. Les thermistances et les résistances métalliques.

Lorsque le faible coût est privilégié par rapport à la précision, il est préférable d'utiliser les thermistances.

thermistances

Les thermistances sont constituées à partir d'oxydes métalliques semi-conducteurs polycristallins (MgO , MgAl_2O_4 , Mn_2O_3 , Fe_3O_4 , NiO etc.) Ce sont des résistances dont la valeur varie en fonction de la température.

Il en existe deux types : les thermistances à Coefficient de Température Négatif (CTN), ou celles moins courantes, à Coefficient de Température Positif (CTP).

Les thermistances ont une très grande sensibilité thermique, ce qui rend leur utilisation très aisée. Un simple étage d'amplification permet une mesure à $0,5\text{ }^{\circ}\text{C}$ près, celle-ci pouvant être améliorée avec un amplificateur d'instrumentation.

Un autre intérêt est que leur valeur ohmique est élevée (supérieur à $1000\ \Omega$) ce qui rend négligeable les fils de liaisons de la sonde. Les points faibles des thermistances sont la dérive de leurs caractéristiques au cours du temps et d'une interchangeabilité médiocre.



Lorsqu'il est nécessaire d'effectuer des mesures sur une grande plage de températures, ou lorsqu'il faut une plus grande précision, il est préférable d'utiliser les résistances métalliques (ou RTD).

RTDs et pRTD

Les résistances métalliques (ou Resistance Temperature Detector pour RTD) sont composées d'un fil métallique cylindrique dont la valeur ohmique est relation de la température. Il s'agit de sondes thermorésistives.

Pour une plus grande sensibilité il faut choisir un métal à Coefficient de Température (CTR) élevé. Pour une meilleure fidélité, le métal doit être exempt de tout défaut structural et doit présenter une grande résistance à la corrosion et à la pollution. La linéarité est également un point important mais en conflit avec la fidélité. En effet, pour obtenir un métal linéaire il est possible d'associer plusieurs métaux à CTR différents. Le métal doit être très ductile pour permettre un encombrement réduit.

Le platine est la matière la plus adaptée à ce type de sonde. Il peut s'obtenir très pur (99,999 %). De plus, il s'agit d'un matériau très ductile. Les RTDs de platine se nomment Platine Resistance Temperature Detectors (pRTDs).

Ces sondes bénéficient d'une grande plage de mesure. Il s'en distingue plusieurs types. Les Pt100, Pt500, Pt1000 etc.

Leur nom est directement lié à leur valeur ohmique à 0 °C. La Pt100 a une valeur de 100,00 Ω à 0,00 °C et de 138,51 Ω à 100,00 °C ; la Pt1000 1000,00 Ω à 0,00 °C, ...

A noter que, citation de Wikipedia (<http://fr.wikipedia.org/>), « la sonde à résistance de platine définie par l'EIT-90 est la plus élaborée et permet des mesures de température de très hautes exactitudes (voir "supplementary informations for the realization of the ITS-90", édité par le BIPM, Pavillon de Sèvres en 1990) [...] Un tel instrument permet, en montage 4 fils et relié à un pont de résistances de type Guidline ou ASL F900, de détecter des variations de températures de l'ordre de 0,1 milli-Kelvin et de réduire les incertitudes en le mettant en œuvre dans des cellules points fixes de qualité métrologique à un ordre de grandeur infime ». Pour ce faire il faut utiliser une Pt25.

On remarque que les meilleures précisions sont atteintes avec les PRTDs. La sonde Pt100 est la plus utilisée puisqu'elle a la courbe la plus linéaire, de plus elle est normalisée. Il s'agit de la norme DIN 43760 (IEC 751) (conférer à l'annexe A page 1 des annexes). Ce qui rend cette sonde facilement interchangeable. Contrairement aux thermistances, elle n'a pas besoin d'être qualifiée.

Avec les Pt100, il est possible d'atteindre une précision de 0,1 °C (voir supérieure par spéculation et une bonne linéarisation).

Il existe plusieurs classes de Pt100. La classe A est la plus précise, alors que la



classe B est la moins coûteuse. Son temps de réponse est 0,1 seconde. Sa sensibilité statique vaut $\frac{\Delta R}{\Delta^\circ\text{C}}$, soit $S = \frac{390,48-18,52}{1050} = 0,354 \Omega/^\circ\text{C}$.

Elles sont disponibles en trois versions différentes. La version deux fils est la plus répandue, mais il est nécessaire de passer à la version trois (voir quatre) fils pour annuler l'effet résistif des fils de connexion.

Pour exploiter cette sonde il faut utiliser un circuit conditionneur. Il peut être réalisé par un circuit intégré, ou plusieurs. Tout dépend du signal voulu en sortie, de l'alimentation disponible, ou de la précision désirée.

3.2 Pt100 : Circuit conditionneur

Pour exploiter une sonde Pt100 il existe de nombreuses possibilités. **Dans tout les cas, ne pas la faire parcourir par un courant supérieur à 3 mA pour éviter l'auto-échauffement produit par l'effet Joule.** Le constructeur préconise un courant de 1 mA.

3.2.1 Les circuits intégrés

Avant tout vérifier la disponibilité des produits qui vous paraissent idéaux. Je suis parti sur un **ADT70**, il semblait idéal pour mon utilisation : mais après la réalisation du schéma il a fallu passer la commande. Et là, grosse déception. Il était indisponible. Une vérification rapide sur <http://www.analog.com/> m'a éclairci : **This product is currently being phased out of production. Not recommended for new designs.** Ce qui en français correspond à « Production arrêtée ».

Ils nous précisent que pour trouver un produit équivalent il faut regarder dans un certain tableau. Personnellement, je n'en ai pas trouvé.

Néanmoins, en regardant la structure interne de l'ADT70 on peut se faire une idée d'un premier circuit conditionneur (voir ci-après figure 3.1 page 17). Il est composé d'une source de courant 1 mA basée sur une référence de tension 2,5 V. D'un amplificateur d'instrumentation et d'un autre amplificateur pouvant être branché en comparateur pour détecter un certain seuil de température par exemple.

Le XTR103 convertit la variation de résistance en variation de courant. Sa sortie est analogique et comprise entre 4 mA et 20 mA. Son alimentation est unipolaire. Avec l'alimentation positive comprise en 9 V et 40 V.

Ce circuit peut être suivi du RCV420, pour convertir la variation de courant (4 – 20 mA) en variation de tension (0 – 5 V).

L'Interface Transducteur Universelle (ITU) ou « Universal Transducer Interface » (UTI) est un convertisseur de signal du capteur en signal horaire, basé sur un oscillateur période-modulée. La sortie est sur 13 bits. il s'agit de Modulation

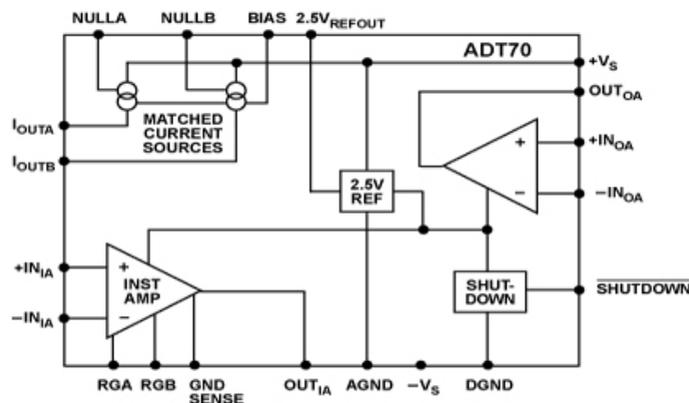


FIG. 3.1 – Structure interne de l'ADT70

de Largeur d'Impulsion (MLI, ou PWM en anglais pour " Pulse Width Modulation "). Site internet : <http://www.smartec.fr/fr/uti-FR.php>. Il a une alimentation unipolaire, mais sa précision est légèrement plus faible.

Le LTC2433 est un circuits dialoguant par bus SPI (pour Serial Peripheral Interface). Il a une résolution de 16 bits et une alimentation unipolaire.

Tous ces circuits ont également l'avantage d'être compensés. Le constructeur nous donne la fiabilité des mesures et leurs précisions. De plus, hormis la sonde (2, 3, ou 4 fils), ils n'ont en général besoin que d'une résistance pour le gain de l'amplificateur.

3.2.2 Un circuit adapté

Lorsque l'on a des contraintes extérieures (alimentation, températures, ...) il est plus difficile de trouver un circuit intégré nous convenant. Puisque je suis limité en consommation et en alimentation, j'ai trouvé préférable de produire mon circuit conditionneur.

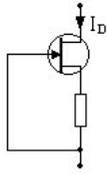
Pour éviter l'auto-échauffement de la sonde et simplifier les calculs par la suite, j'ai décidé de contrôler ma sonde en courant. L'utilisation d'un pont de Wheatstone s'est avéré nécessaire pour obtenir une meilleure précision de mesure. Pour amplifier le signal généré par le pont de Wheatstone (quelques mV/°C) il faut utiliser un amplificateur d'instrumentation.

Le principe général est de recréer l'ADT70 (conférer figure 3.1 page 17).

Pour la source de courant, plusieurs possibilités sont à étudier.

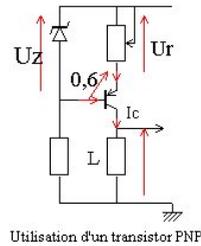


Source de courant stable



Pour réaliser une source de courant il y a la solution du transistor à effet de champ (JFET par exemple). Il faut relier la grille à la source. Pour le courant débité, il s'agit de I_{DSS} (en général de 0,2 à 5 mA). On en trouve sous forme de diodes. Ce sont tout simplement des diodes à courant constant.

Il est aussi possible d'en réaliser avec une diode zener associée à un transistor, mais cela introduit de nombreux composants. Tous susceptibles d'être affectés par la température. Sur ce schéma on trouve $I_C = \frac{U_z - 0,6}{R}$.



Utilisation d'un transistor PNP

Je me suis rabattu sur la solution des amplificateurs opérationnels pour maîtriser les composantes de défaut (offset, dérive, thermique, etc.) et l'alimentation. Voir figure 3.2 page 18.

Le schéma de principe est simple, il faut partir d'une source de tension stable (appelé référence de tension, d'où V_{ref}). Pour mon application la source de tension correspondant le mieux à mon utilisation est le MCP1525 de Microchip. Il s'agit d'une référence de tension 2,5 V. Faible bruit, faible dérive thermique (± 50 ppm/°C). Le MCP1525 est idéal pour les température supérieures à 25 °C : sa sortie est de 2,500 V à 25,00 °C et plus, pour 2,498 à 0,00 °C. Sa qualité se dégrade avec les températures négatives.

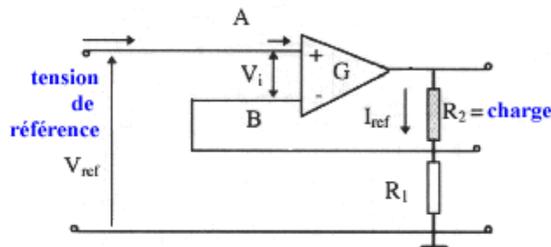


FIG. 3.2 – Source de courant stable

Soit I_{ref} le courant dans R_2 , on obtient $I_{ref} = \frac{V_{ref}}{R_1}$. Puisque ma charge est composée d'un pont de Wheatstone (conférer figure page 3.3 page 19), en R_2 , il faut $I_{ref} = 2$ mA. De cette manière la sonde sera parcourue par un courant de 1 mA.

Il faut bien entendu choisir R_1 et R_2 avec un très faible coefficient de température. C'est pourquoi j'ai choisi la résistance en premier pour ensuite calculer le courant I_{ref} .

Pour choisir un amplificateur opérationnel convenable il faut faire attention à l'offset d'entrée et également à la dérive thermique. L'alimentation n'est pas non plus un point négligeable.



Il m'a paru bon de prendre l'OPA335. Il a un offset de $5 \mu V$ (max), une dérive de $0,05 \mu V/^\circ C$. Une faible consommation et une alimentation unipolaire.

La charge R_2 est donc composée d'un pont de Wheatstone pour plus de précision.

Pont de Wheatstone

Ce circuit permet de déterminer la résistance électrique d'un élément situé dans une branche du pont. Voir figure 3.3 pour la représentation schématique du pont.

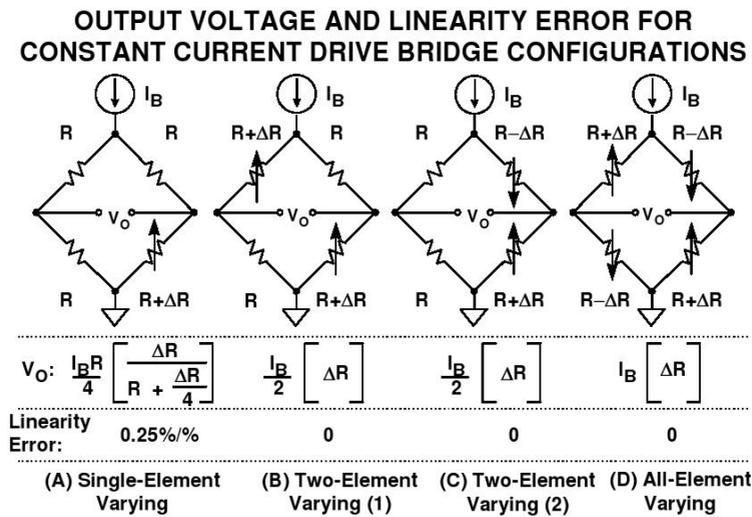


FIG. 3.3 – Pont de Wheatstone

V_0 est la tension à mesurer. Elle se nomme tension de déséquilibre. En effet, lorsque le pont est à l'équilibre V_0 est nulle. Pour une mesure commençant à $0^\circ C$ il est judicieux de placer le pont à l'équilibre pour $T = 0^\circ C$. Il faut donc choisir les résistances composant le pont à 100Ω pour la Pt100. Il faut bien entendu choisir des résistances de précision à faible coefficient de température.

Pour cela j'ai choisi des résistances de précision à couche métallique. Le constructeur les certifie à 0,05 % et le vendeur à 0,1 %. Elles sont à bruit et coefficient de température négligeables (d'après le constructeur).

De par sa nature, le pont de Wheatstone introduit une erreur de linéarité dépendant du nombre d'éléments variants. Se référer à la figure 3.3 pour comparer. Pour éviter cette erreur en plus, il faut utiliser deux éléments variants. Lorsque le pont est contrôlé en tension, l'erreur de linéarité est de 0,5 % pour un élément variant. Encore une raison de préférer la source de courant. De plus les sondes Pt100 étant normalisées, elles varieraient de paire.

Par calcul on retrouve rapidement la valeur de V_0 : En partant du schéma à deux éléments variant, on a $R_{Pt100} = R + \delta R$ avec R la valeur de la Pt100 à $0,00^\circ C$,



soit $R = 100,00 \Omega$.

Les deux branches ayant la même résistance équivalente, le courant traversant les deux sondes sera le même. Soit I le courant fourni par la source de courant et I' le courant traversant une sonde.

On a $V_0 = I'(R + \delta R) - I' \times R$, or $I' = \frac{I}{2}$. Finalement on trouve, $V_0 = \frac{I}{2} \times \delta R$.

L'équilibre du pont ne variant que de $0,4 \text{ mV}/^\circ\text{C}$ environ, il est nécessaire d'amplifier le signal.

Amplificateur

Pour éviter tout défaut supplémentaire il faut se servir d'un amplificateur d'instrumentation. Il est possible d'en réaliser avec trois amplificateurs opérationnels mais cela ne convient pas pour une faible consommation et cela ajoute des perturbations.

La solution est donc l'amplificateur d'instrumentation intégré, avec une alimentation unipolaire. Ils ont en général un très faible offset d'entrée et un faible coefficient de température. Il en existe beaucoup, ceux que j'ai retenus sont le INA122 et le INA118. Le INA118 est le plus approprié. Avec un offset d'entrée de $50 \mu\text{V}$ pour une variation de $0,5 \mu\text{V}/^\circ\text{C}$.

Son gain étant réglable par une résistance R_G il faut l'optimiser pour l'entrée analogique 10 bits du PIC18F. La sortie doit donc être comprise entre 0 et 5 V.

Soit G le gain, le constructeur nous donne la formule suivante : $G = 1 + \frac{50 \cdot 10^3}{R_G}$

Soit V_s la tension de sortie de l'amplificateur, 5 V pour un maximum de 30°C (soit $\delta R = 11,67 \Omega$ d'après l'annexe A on obtient un gain $G = 334,33$ (conférer Annexe B. Toujours en annexe, $R_g = 150 \Omega$).

On obtient donc en entrée du PIC $V_s = 334,33 \times \frac{I}{2} \times \delta R$, soit environ $0,166 \text{ V}/^\circ\text{C}$. Pour calculer la précision des mesures, se référer à l'annexe B (page 4 des annexes).

Interface CAN : exploitation de la sonde

Pour exploiter la tension analogique en sortie de l'amplificateur d'instrumentation, j'ai choisi d'utiliser le Convertisseur Analogique/Numérique (ou ADC) du PIC. Sa résolution est de 10 bits.

Des calculs effectués en annexe B, on retiendra les relations (B.18) et (B.19), soit à entrer dans le calculateur :



$$T = \left(T_0 + \frac{\sqrt{A^2 + 4B \cdot r} - A}{2B} \pm 0,07 \right) ^\circ\text{C} \quad (3.1)$$

$$r = \frac{2V_s}{G \cdot I \cdot R} \pm 0,0124 \quad (3.2)$$

$$A = 3,9083 \cdot 10^{-3} \text{ K}^{-1} \quad (3.3)$$

$$B = -5,775 \cdot 10^{-7} \text{ K}^{-2} \quad (3.4)$$

$$T_0 = 0,00 \text{ } ^\circ\text{C} \quad (3.5)$$

$$G = 425,53 \pm 0,4255 \quad (3.6)$$

$$G = 425,53 \pm 0,4255 \quad (3.7)$$

$$I = (2,499 \pm 0,004) \text{ mA} \quad (3.8)$$

$$R = (100 \pm 0,1) \Omega \quad (3.9)$$

La variable étant V_s .

On obtient une mesure fidèle à 0,1 °C (théoriquement), ce qui correspond au cahier des charges. Il ne manque plus qu'à pouvoir comparer avec un matériel plus performant.

Le code suivant permet de récupérer la valeur analogique V_s sur AN0 du PIC et de la retourner en valeur numérique.

```

1 unsigned char getTemp()
2 {
3     // Début de conversion
4     ADCON0bits.GO = 1;
5
6     // On attend la fin de la conversion
7     while (ADCON0bits.GO);
8
9     //On retourne les bits de données
10    return ADRESH;
11 }
```

Plus grande plage de température

Le défaut de linéarité de la sonde se fait sentir pour les températures négatives. Il faut compenser l'ajustement polynomial quadratique standard.

Pour cela se référer au document *Mesure de température par thermistance Pt100 / Pt1000* de Thomas Meader.

Ce dossier est disponible à cette adresse : <http://lpm.epfl.ch/webdav/site/lpm/shared/enseignement/divers/060914%20Mesure%20de%20température%20Pt100-Pt1000.pdf>

4 Transmission sans fil

4.1 Choix du protocole

Pour les satellites (carte de mesure de température) il est important de ne pas devoir les connecter par fil. En effet, cela permet de les positionner momentanément à divers endroits.

Pour ce projet la meilleure solution reste les ondes électro-magnétiques. De cette manière il est possible de s'affranchir de certains obstacles.

Il existe principalement trois technologies différentes, toutes exploitant la bande des micro-ondes, soit 2,4 Ghz :

1. **Le WIFI**, la référence lorsqu'il faut un débit important. On le rencontre surtout pour les réseaux ethernet,
2. **Le Bluetooth**, utilisé principalement pour connecter entre eux plusieurs périphériques (imprimante, mobile ...),
3. **Le ZigBee**, faible consommation et portée moyenne, ce protocole devrait être la référence en domotique.

| Protocole | ZigBee | Bluetooth | WIFI |
|----------------------|----------|-----------|----------------|
| Norme IEEE | 802.15.4 | 802.15.1 | 802.11a/b/g |
| Besoins mémoire | 4-32 Kb | 250 Kb + | 1 Mb + |
| Durée de vie | Années | Jours | Heures |
| Nombre de nœuds | 65 000+ | 7 | 32 |
| Vitesse de transfert | 250 Kb/s | 1 Mb/s | 11-54-108 Mb/s |
| Portée | 100 m | 10-100 m | 300 m |

Le protocole correspondant le mieux est donc le ZigBee : bonne portée, faible consommation et beaucoup de nœuds possibles (nombres de connexions par réseau).

Le Xbee de MAXSTREAM est très simple à mettre en oeuvre tout en très performant.

4.2 Le module Xbee

Disponible en deux versions (Xbee Pro ou Xbee) il est possible de choisir l'antenne. Filaire, externe ou CMS. Pour l'utilisation du Xbee Pro en Europe, on se



doit de limiter sa puissance d'émission à 10 dBm ce qui est rendu possible par programmation.

J'ai choisi le module Xbee puisque je ne voyais pas l'utilité du module Xbee Pro. Néanmoins, ces deux modules étant interchangeables, cela ne pose pas de problème pour une utilisation future.

La consommation du module Xbee inférieure à 10 μ A en repos et de 50 mA en émission/réception. Idéal pour une alimentation par pile. Ses 16 canaux permettent d'éviter les interférences.

Il dispose de plusieurs entrées-sorties analogiques. Le principal dialogue avec le microcontrôleur est effectué par liaison série. La configuration du module se fait par commande « AT ». Il s'agit d'un dialogue en chaîne de caractères.

Il peut être configuré pour recopier l'état de ses entrées sur un autre module si l'on connaît son adresse, ou même réémettre ce qu'il reçoit pour les modules les plus éloignés. Cette possibilité est très avantageuse pour éloigner les différents satellites.



Très facile à exploiter, il est également simple à connecter. **Attention cependant, son alimentation est de 3,3 V maximum.**

Sur vingt broches, seulement quatre sont nécessaires. Il s'agit de DIN et DOUT pour la liaison série, de Vcc et Gnd pour l'alimentation. Mais avec RTS et CTS il est possible de contrôler l'état des tampons d'émission et réception.

DTR, RTS et CTS permettent la reprogrammation du firmware. Pour cela il faut le kit de MAXSTREAM.

Pour mettre en veille le module, il faut contrôler la broche SLEEP. SLEEP_RQ permet de vérifier l'état du module.

Ce module comprend en tout neuf entrées analogiques/numériques, avec la possibilité de positionner des résistances de pull-up en interne.

Toutes les documentations sont disponibles ici : <http://www.maxstream.net/products/xbee/xbee-oem-rf-module-zigbee.php>.

5 L'USB

De nos jours, l'Universal Serial Bus est un bus de donnée largement utilisé pour sa simplicité d'utilisation et l'alimentation disponible avec un ordinateur.

Il existe plusieurs possibilités en utilisant des PIC :

- Le protocole HID, vitesse limitée à 64ko/s, aucun pilote nécessaire,
- Émulation port COM (CDC) : voir AN956 , sans pilote,
- USB low-speed, vitesse de 1.5Mb/s, nécessite un pilote,
- USB full-speed, vitesse de 12Mb/s, oscillateur 48Mhz minimum, nécessite un pilote

L'émulation du port COM (port série, protocole RS232) est assez facile à mettre en oeuvre : elle permet la compatibilité avec d'anciennes applications PC. Surtout utilisée dans le domaine industriel pour ne pas devoir développer d'autre applications devant gérer l'USB, quand les périphériques migrent du RS232 à l'USB.

La solution USB pour de plus grande vitesse de transfert nécessite la programmation d'un pilote : un programme lourd et complexe à mettre en oeuvre.

La solution d'une interface HID a donc été retenue pour sa relative simplicité, la vitesse de transmission étant amplement suffisante pour l'application désirée.

5.1 Human Interface Device

Le protocole HID (Human Interface Device) est de plus en plus utilisé puisque c'est la manière la plus simple d'utiliser le port USB. Sa seule limitation est la vitesse de transmission. Cette solution est utilisée pour les claviers, les souris, les manettes de jeux vidéos, etc.

L'USB peut nous fournir une alimentation de 5 V limitée à 500 mA. Amplement suffisant pour notre application.

Contrairement à la liaison RS232 et des interfaces sérielles similaires où le format des données envoyées n'est pas défini par défaut, l'USB est composé de plusieurs couches de protocoles imposées. Il s'agit d'un réseau en jeton.

Chaque transaction USB consiste d'un paquet Jeton (Token) pour savoir si il faut attendre une autre réponse ou poursuivre, d'un paquet DATA optionnel (contenant la "charge utile" (payload)) et d'un paquet d'Etat (utilisé pour valider les transactions et pour fournir des moyens de corrections d'erreurs).

Sur le bus USB, c'est l'hôte qui gère la transmission. Il s'agit de l'ordinateur. Il envoie le premier paquet avec le jeton au nouveau périphérique détecté pour qu'il



puisse s'identifier et ainsi créer la connexion. Pour les communications entre périphérique (sans passer par un ordinateur) il n'y a pas d'hôte. Il faut donc travailler en O-T-G (On-The-Go).

L'USB est un bus série. Il utilise quatre fils isolés dont deux sont l'alimentation (+5V et GND). Les deux restants forment une paire torsadée qui véhicule les signaux en mode différentiel. Il utilise un schéma d'encodage NRZI (Non Retour à Zéro inversé) pour envoyer des données.

5.2 Exploitation

Avant de dialoguer avec l'hôte, le périphérique doit s'identifier. Tout périphérique USB a un code VID et PID unique pouvant l'identifier. Le VID est le Vendor IDentification (pour microchip VID = 0x04D8) et PID est le Product IDentification. A la connexion, le périphérique s'identifie et spécifie le protocole (volumes des paquets transmis, le débit, le courant maximum à débiter, etc.).

La vitesse de la communication ne dépend que de deux paramètres, la taille du paquet envoyé (maximum 64 octets) et la période séparant chaque demande du pic (« pooling » minimum : une milliseconde). On aura donc 64 ko/s dans les meilleures conditions.

La connexion d'un nouveau périphérique est détectée par un changement de niveau sur les lignes de données. En effet, des résistances de pull-up y sont positionnées pour spécifier la vitesse de transmission (interne au PIC).

Pour le processus d'énumération (le périphérique s'identifie et l'hôte lui assigne une adresse), voir en fonction du système d'exploitation.

Pour linux, voir <http://www.linux-usb.org/> et <http://www.usbman.com/linuxusb.htm>.

Désormais le périphérique est en fonctionnement, l'hôte demande un pilote pour le nouveau périphérique détecté. Pour un périphérique HID, le pilote est reconnu automatiquement.

6 Impressions personnelles

Malgré un sujet de prime abord assez simple, la prise en main de gEDA à pris beaucoup de temps. De plus, l'installation de gEDA à nécessité la réinstallation du système.

En effet, Scientifique Linux Cern n'ayant pas Python 2.5, il à été plus rapide de passer sous Fedora Core 6.

Plus précisément, j'ai commencé à découvrir gEDA plus de trois semaines après mon arrivée. Avant j'ai essayé des installations et testé KiCaD. J'ai également fait beaucoup de recherches pour la mesure d'une température. L'étude m'a pris du temps, avant que j'organise mes recherches d'une autre manière.

Ne m'attendant pas à ce que gEDA soit si long à prendre en main, j'y ai aussi passé beaucoup de temps. Là encore j'ai changé ma méthode de recherche pour ne plus suivre les tutoriaux sur la suite mais bien sur les divers programmes qui eux-même évoluent très vite.

Cependant, en suivant les bons conseils et sans partir sur de mauvaises pistes, cette suite n'est pas si difficile à prendre en main. Pour ce faire, j'ai rédigé l'article connexe à ce rapport.

Certain composants commandés mi-avril étant encore en attentes, arrivé mi-juin la véritable programmation n'a pas encore débuté. De ce fait, le projet n'a pu arriver à son terme.

Néanmoins j'ai beaucoup appris durant ce stage, tant au niveau professionnel que personnel.

D'un point de vue personnel, je vais définitivement me tourner vers Linux pour la conception électronique. Ayant passé presque trois mois sous Linux, j'ai énormément progressé dans l'administration d'un système UNIX.

Professionnellement, j'ai découvert le domaine de l'étude et de la recherche. Étant un univers passionnant, je me dois de continuer mes études pour espérer élaborer et optimiser par la suite.

De part ce que j'ai appris et réalisé durant ce stage, c'est sans aucun regret que je recommencerais un tel projet. Merci l'équipe AB-CO-HT !

Conclusion

Au cours de cette étude, nous avons abordé plusieurs thèmes.

Dorénavant nous savons de quelle manière accéder à la Conception électronique Assisté par Ordinateur et la programmation des microcontrôleurs sous linux. Tous les programmes utilisés étant gratuits, cela augmente les chances que ce projet soit repris pour une utilisation future.

Pour ce qui est du projet en lui-même, les cartes étant réalisées il s'agit surtout maintenant de développer correctement les différents programmes pour permettre la communication sans fil et la liaison USB. Toutes les références citées au cours de ce projet l'ont été dans cette optique. En suivant ces quelques liens, il est possible de trouver de bons exemples que ce soit sur l'USB ou sur le Xbee.

La précision des mesures obtenue étant suffisante théoriquement, il suffirait de vérifier la justesse des mesures avec un élément plus précis. Cependant, la sonde devrait être assez fidèle. Pour mieux contrôler l'incertitude des mesures, je vous propose de reprendre mes calculs effectués en annexe **B** pour remplacer les éléments imprécis.

Dans la continuité du projet, il pourrait être avisé de monitorer les températures, avec par exemple *cacti* (basé sur RRD-tools).

Comme chez CMS : http://cmsdoc.cern.ch/cmscluster/newcacti/graph_image.php?action=view&local_graph_id=18&rra_id=1.

En ce qui concerne la consommation de la carte de mesure, les tests restent encore à effectuer (il manque encore la réception de certains composants), néanmoins elle ne devrait pas consommer plus de 2 mA lors des mesures. Et une dizaine de micro-Ampères au repos, c'est à dire 80 % du temps.

Glossaire

Ductile Se dit d'un matériau capable de se déformer plastiquement sans se rompre.

Justesse Qualité du capteur pour donner des indications égales au mesurande.

Kynaptic Gestionnaire de paquet pour KDE.

Linéarité Sensibilité du capteur, indépendante du mesurande.

Mesurande Grandeur physique à mesurer (température, pressions, ...).

Netlist Liste des fils de connexions entre composants.

Fidélité L'aptitude d'un capteur à donner des indications identiques pour une même valeur de mesurande.

Finesse Aptitude du capteur à fournir une grandeur mesurée sans la modifier par sa présence.

Précision Pour un capteur, aptitude à donner une indication proche de la vraie valeur.

Rapidité Temps de réponse, liée à la fréquence propre du capteur.

Résolution La plus petite valeur de mesurande que le capteur peut détecter.

Synaptic Gestionnaire de paquet sous Linux (Debian), voir aussi Kynaptic.

Les acronymes

CAN Convertisseur Analogique Numérique

CAO Conception Assisté par Ordinateur

GPS Global Positioning System

HID Human Interface Device

ICD(2) In Circuit Debugger (2)

JFET Junction Field Effect Transistor

PIC Nom des Microcontrôleurs conçus par Microchip

PCB Printed Circuit Board

SDCC Small Device C Compilateur

USB Universal Serial Bus

Bibliographie

<http://press.web.cern.ch/>, toutes les nouvelles du CERN, toutes les données et dates importantes.

<http://fr.wikipedia.org/>, diverses définitions.

<http://www.iut-lannion.fr/LEMEN/MPDOC/CMEST/mtherres.htm>, mesure de température.

<http://aviatechno.free.fr/thermo/rtd03.php>, mesure de température.

http://perso.orange.fr/michel.hubin/physique/elec/chap_aop2.htm, divers schémas électroniques.

<http://forums.futura-sciences.com/forum40.html>, un forum avec des sujets intéressants sur la Pt100.

<http://www.planete-sciences.org/forums/>, forum très intéressant sur l'utilisation du module Xbee. Voir section Robotique.

<http://www.jeffreyleblanc.org/energy/xbee.htm>, sources pour les commandes AT du module Xbee.

<http://www.todopic.com.ar/foros/index.php?topic=16393.0;topicseen>, un bon exemple sur l'utilisation du module Xbee.

<http://www.totofweb.net/>, un beau projet avec le module Xbee.

http://itp.nyu.edu/~raf275/meshnetworking/XBee/XBee_firmware_upgrade.html, des explications sur la mise à jour du module Xbee ("upgrading firmware").

<http://pic18fusb.online.fr/>, les PIC pour l'USB en HID.

http://www.analog.com/UploadedFiles/Associated_Docs/49470200sscsect2.PDF, un dossier très intéressant en rapport avec le pont de Wheatstone et la Pt100. Dommage que je ne l'ai trouvé qu'à la fin, il m'a permis de vérifier mon schéma, puisqu'il est reproduit à l'identique (page 11 figure 2.12).

<http://lpm.epfl.ch/webdav/site/lpm/shared/enseignement/divers/060914%20Mesure%20de%20température%20Pt100-Pt1000.pdf>, un dossier permettant d'ajuster l'erreur de l'ajustement quadratique standard pour les températures négatives de la Pt100 (et Pt1000).

<http://fr.wikipedia.org/wiki/Pt100> pour une sonde encore plus précise.

<http://aviatechno.free.fr/thermo/rtd03.ph> des explications sur la linéarité de la Pt100.

Annexes

A Tables Pt100

Table de résistivité de la Pt100 (valeur à multiplier par 10 pour avoir la table de la Pt1000)

Selon ITS-90/DIN EN 60751 Unités : T en °C et R en Ω

| °C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | °C |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|
| -200.00 | 18.52 | | | | | | | | | | -200.00 |
| -190.00 | 22.83 | 22.40 | 21.97 | 21.54 | 21.11 | 20.68 | 20.25 | 19.82 | 19.38 | 18.95 | -190.00 |
| -180.00 | 27.10 | 26.67 | 26.24 | 25.82 | 25.39 | 24.97 | 24.54 | 24.11 | 23.68 | 23.25 | -180.00 |
| -170.00 | 31.34 | 30.91 | 30.49 | 30.07 | 29.64 | 29.22 | 28.80 | 28.37 | 27.95 | 27.52 | -170.00 |
| -160.00 | 35.54 | 35.12 | 34.70 | 34.28 | 33.86 | 33.44 | 33.02 | 32.60 | 32.18 | 31.76 | -160.00 |
| -150.00 | 39.72 | 39.31 | 38.89 | 38.47 | 38.05 | 37.64 | 37.22 | 36.80 | 36.38 | 35.96 | -150.00 |
| -140.00 | 43.88 | 43.46 | 43.05 | 42.63 | 42.22 | 41.80 | 41.39 | 40.97 | 40.56 | 40.14 | -140.00 |
| -130.00 | 48.00 | 47.59 | 47.18 | 46.77 | 46.36 | 45.94 | 45.53 | 45.12 | 44.70 | 44.29 | -130.00 |
| -120.00 | 52.11 | 51.70 | 51.29 | 50.88 | 50.47 | 50.06 | 49.65 | 49.24 | 48.83 | 48.42 | -120.00 |
| -110.00 | 56.19 | 55.79 | 55.38 | 54.97 | 54.56 | 54.15 | 53.75 | 53.34 | 52.93 | 52.52 | -110.00 |
| -100.00 | 60.26 | 59.85 | 59.44 | 59.04 | 58.63 | 58.23 | 57.82 | 57.41 | 57.01 | 56.60 | -100.00 |
| -90.00 | 64.30 | 63.90 | 63.49 | 63.09 | 62.68 | 62.28 | 61.88 | 61.47 | 61.07 | 60.66 | -90.00 |
| -80.00 | 68.33 | 67.92 | 67.52 | 67.12 | 66.72 | 66.31 | 65.91 | 65.51 | 65.11 | 64.70 | -80.00 |
| -70.00 | 72.33 | 71.93 | 71.53 | 71.13 | 70.73 | 70.33 | 69.93 | 69.53 | 69.13 | 68.73 | -70.00 |
| -60.00 | 76.33 | 75.93 | 75.53 | 75.13 | 74.73 | 74.33 | 73.93 | 73.53 | 73.13 | 72.73 | -60.00 |
| -50.00 | 80.31 | 79.91 | 79.51 | 79.11 | 78.72 | 78.32 | 77.92 | 77.52 | 77.12 | 76.73 | -50.00 |
| -40.00 | 84.27 | 83.87 | 83.48 | 83.08 | 82.69 | 82.29 | 81.89 | 81.50 | 81.10 | 80.70 | -40.00 |
| -30.00 | 88.22 | 87.83 | 87.43 | 87.04 | 86.64 | 86.25 | 85.85 | 85.46 | 85.06 | 84.67 | -30.00 |
| -20.00 | 92.16 | 91.77 | 91.37 | 90.98 | 90.59 | 90.19 | 89.80 | 89.40 | 89.01 | 88.62 | -20.00 |
| -10.00 | 96.09 | 95.69 | 95.30 | 94.91 | 94.52 | 94.12 | 93.73 | 93.34 | 92.95 | 92.55 | -10.00 |
| 0.00 | 100.00 | 99.61 | 99.22 | 98.83 | 98.44 | 98.04 | 97.65 | 97.26 | 96.87 | 96.48 | 0.00 |
| °C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | °C |
| 0.00 | 100.00 | 100.39 | 100.78 | 101.17 | 101.56 | 101.95 | 102.34 | 102.73 | 103.12 | 103.51 | 0.00 |
| 10.00 | 103.90 | 104.29 | 104.68 | 105.07 | 105.46 | 105.85 | 106.24 | 106.63 | 107.02 | 107.40 | 10.00 |
| 20.00 | 107.79 | 108.18 | 108.57 | 108.96 | 109.35 | 109.73 | 110.12 | 110.51 | 110.90 | 111.29 | 20.00 |
| 30.00 | 111.67 | 112.06 | 112.45 | 112.83 | 113.22 | 113.61 | 114.00 | 114.38 | 114.77 | 115.15 | 30.00 |
| 40.00 | 115.54 | 115.93 | 116.31 | 116.70 | 117.08 | 117.47 | 117.86 | 118.24 | 118.63 | 119.01 | 40.00 |
| 50.00 | 119.40 | 119.78 | 120.17 | 120.55 | 120.94 | 121.32 | 121.71 | 122.09 | 122.47 | 122.86 | 50.00 |
| 60.00 | 123.24 | 123.63 | 124.01 | 124.39 | 124.78 | 125.16 | 125.54 | 125.93 | 126.31 | 126.69 | 60.00 |
| 70.00 | 127.08 | 127.46 | 127.84 | 128.22 | 128.61 | 128.99 | 129.37 | 129.75 | 130.13 | 130.52 | 70.00 |
| 80.00 | 130.90 | 131.28 | 131.66 | 132.04 | 132.42 | 132.80 | 133.18 | 133.57 | 133.95 | 134.33 | 80.00 |
| 90.00 | 134.71 | 135.09 | 135.47 | 135.85 | 136.23 | 136.61 | 136.99 | 137.37 | 137.75 | 138.13 | 90.00 |
| 100.00 | 138.51 | 138.88 | 139.26 | 139.64 | 140.02 | 140.40 | 140.78 | 141.16 | 141.54 | 141.91 | 100.00 |
| 110.00 | 142.29 | 142.67 | 143.05 | 143.43 | 143.80 | 144.18 | 144.56 | 144.94 | 145.31 | 145.69 | 110.00 |
| 120.00 | 146.07 | 146.44 | 146.82 | 147.20 | 147.57 | 147.95 | 148.33 | 148.70 | 149.08 | 149.46 | 120.00 |
| 130.00 | 149.83 | 150.21 | 150.58 | 150.96 | 151.33 | 151.71 | 152.08 | 152.46 | 152.83 | 153.21 | 130.00 |

| | | | | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 140.00 | 153.58 | 153.96 | 154.33 | 154.71 | 155.08 | 155.46 | 155.83 | 156.20 | 156.58 | 156.95 | 140.00 |
| 150.00 | 157.33 | 157.70 | 158.07 | 158.45 | 158.82 | 159.19 | 159.56 | 159.94 | 160.31 | 160.68 | 150.00 |
| 160.00 | 161.05 | 161.43 | 161.80 | 162.17 | 162.54 | 162.91 | 163.29 | 163.66 | 164.03 | 164.40 | 160.00 |
| 170.00 | 164.77 | 165.14 | 165.51 | 165.89 | 166.26 | 166.63 | 167.00 | 167.37 | 167.74 | 168.11 | 170.00 |
| 180.00 | 168.48 | 168.85 | 169.22 | 169.59 | 169.96 | 170.33 | 170.70 | 171.07 | 171.43 | 171.80 | 180.00 |
| 190.00 | 172.17 | 172.54 | 172.91 | 173.28 | 173.65 | 174.02 | 174.38 | 174.75 | 175.12 | 175.49 | 190.00 |
| °C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | °C |
| 200.00 | 175.86 | 176.22 | 176.59 | 176.96 | 177.33 | 177.69 | 178.06 | 178.43 | 178.79 | 179.16 | 200.00 |
| 210.00 | 179.53 | 179.89 | 180.26 | 180.63 | 180.99 | 181.36 | 181.72 | 182.09 | 182.46 | 182.82 | 210.00 |
| 220.00 | 183.19 | 183.55 | 183.92 | 184.28 | 184.65 | 185.01 | 185.38 | 185.74 | 186.11 | 186.47 | 220.00 |
| 230.00 | 186.84 | 187.20 | 187.56 | 187.93 | 188.29 | 188.66 | 189.02 | 189.38 | 189.75 | 190.11 | 230.00 |
| 240.00 | 190.47 | 190.84 | 191.20 | 191.56 | 191.92 | 192.29 | 192.65 | 193.01 | 193.37 | 193.74 | 240.00 |
| 250.00 | 194.10 | 194.46 | 194.82 | 195.18 | 195.55 | 195.91 | 196.27 | 196.63 | 196.99 | 197.35 | 250.00 |
| 260.00 | 197.71 | 198.07 | 198.43 | 198.79 | 199.15 | 199.51 | 199.87 | 200.23 | 200.59 | 200.95 | 260.00 |
| 270.00 | 201.31 | 201.67 | 202.03 | 202.39 | 202.75 | 203.11 | 203.47 | 203.83 | 204.19 | 204.55 | 270.00 |
| 280.00 | 204.90 | 205.26 | 205.62 | 205.98 | 206.34 | 206.70 | 207.05 | 207.41 | 207.77 | 208.13 | 280.00 |
| 290.00 | 208.48 | 208.84 | 209.20 | 209.56 | 209.91 | 210.27 | 210.63 | 210.98 | 211.34 | 211.70 | 290.00 |
| 300.00 | 212.05 | 212.41 | 212.76 | 213.12 | 213.48 | 213.83 | 214.19 | 214.54 | 214.90 | 215.25 | 300.00 |
| 310.00 | 215.61 | 215.96 | 216.32 | 216.67 | 217.03 | 217.38 | 217.74 | 218.09 | 218.44 | 218.80 | 310.00 |
| 320.00 | 219.15 | 219.51 | 219.86 | 220.21 | 220.57 | 220.92 | 221.27 | 221.63 | 221.98 | 222.33 | 320.00 |
| 330.00 | 222.68 | 223.04 | 223.39 | 223.74 | 224.09 | 224.45 | 224.80 | 225.15 | 225.50 | 225.85 | 330.00 |
| 340.00 | 226.21 | 226.56 | 226.91 | 227.26 | 227.61 | 227.96 | 228.31 | 228.66 | 229.02 | 229.37 | 340.00 |
| 350.00 | 229.72 | 230.07 | 230.42 | 230.77 | 231.12 | 231.47 | 231.82 | 232.17 | 232.52 | 232.87 | 350.00 |
| 360.00 | 233.21 | 233.56 | 233.91 | 234.26 | 234.61 | 234.96 | 235.31 | 235.66 | 236.00 | 236.35 | 360.00 |
| 370.00 | 236.70 | 237.05 | 237.40 | 237.74 | 238.09 | 238.44 | 238.79 | 239.13 | 239.48 | 239.83 | 370.00 |
| 380.00 | 240.18 | 240.52 | 240.87 | 241.22 | 241.56 | 241.91 | 242.26 | 242.60 | 242.95 | 243.29 | 380.00 |
| 390.00 | 243.64 | 243.99 | 244.33 | 244.68 | 245.02 | 245.37 | 245.71 | 246.06 | 246.40 | 246.75 | 390.00 |
| °C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | °C |
| 400.00 | 247.09 | 247.44 | 247.78 | 248.13 | 248.47 | 248.81 | 249.16 | 249.50 | 249.85 | 250.19 | 400.00 |
| 410.00 | 250.53 | 250.88 | 251.22 | 251.56 | 251.91 | 252.25 | 252.59 | 252.93 | 253.28 | 253.62 | 410.00 |
| 420.00 | 253.96 | 254.30 | 254.65 | 254.99 | 255.33 | 255.67 | 256.01 | 256.35 | 256.70 | 257.04 | 420.00 |
| 430.00 | 257.38 | 257.72 | 258.06 | 258.40 | 258.74 | 259.08 | 259.42 | 259.76 | 260.10 | 260.44 | 430.00 |
| 440.00 | 260.78 | 261.12 | 261.46 | 261.80 | 262.14 | 262.48 | 262.82 | 263.16 | 263.50 | 263.84 | 440.00 |
| 450.00 | 264.18 | 264.52 | 264.86 | 265.20 | 265.53 | 265.87 | 266.21 | 266.55 | 266.89 | 267.22 | 450.00 |
| 460.00 | 267.56 | 267.90 | 268.24 | 268.57 | 268.91 | 269.25 | 269.59 | 269.92 | 270.26 | 270.60 | 460.00 |
| 470.00 | 270.93 | 271.27 | 271.61 | 271.94 | 272.28 | 272.61 | 272.95 | 273.29 | 273.62 | 273.96 | 470.00 |
| 480.00 | 274.29 | 274.63 | 274.96 | 275.30 | 275.63 | 275.97 | 276.30 | 276.64 | 276.97 | 277.31 | 480.00 |
| 490.00 | 277.64 | 277.98 | 278.31 | 278.64 | 278.98 | 279.31 | 279.64 | 279.98 | 280.31 | 280.64 | 490.00 |
| 500.00 | 280.98 | 281.31 | 281.64 | 281.98 | 282.31 | 282.64 | 282.97 | 283.31 | 283.64 | 283.97 | 500.00 |
| 510.00 | 284.30 | 284.63 | 284.97 | 285.30 | 285.63 | 285.96 | 286.29 | 286.62 | 286.95 | 287.29 | 510.00 |
| 520.00 | 287.62 | 287.95 | 288.28 | 288.61 | 288.94 | 289.27 | 289.60 | 289.93 | 290.26 | 290.59 | 520.00 |
| 530.00 | 290.92 | 291.25 | 291.58 | 291.91 | 292.24 | 292.56 | 292.89 | 293.22 | 293.55 | 293.88 | 530.00 |
| 540.00 | 294.21 | 294.54 | 294.86 | 295.19 | 295.52 | 295.85 | 296.18 | 296.50 | 296.83 | 297.16 | 540.00 |
| 550.00 | 297.49 | 297.81 | 298.14 | 298.47 | 298.80 | 299.12 | 299.45 | 299.78 | 300.10 | 300.43 | 550.00 |
| 560.00 | 300.75 | 301.08 | 301.41 | 301.73 | 302.06 | 302.38 | 302.71 | 303.03 | 303.36 | 303.69 | 560.00 |
| 570.00 | 304.01 | 304.34 | 304.66 | 304.98 | 305.31 | 305.63 | 305.96 | 306.28 | 306.61 | 306.93 | 570.00 |
| 580.00 | 307.25 | 307.58 | 307.90 | 308.23 | 308.55 | 308.87 | 309.20 | 309.52 | 309.84 | 310.16 | 580.00 |
| 590.00 | 310.49 | 310.81 | 311.13 | 311.45 | 311.78 | 312.10 | 312.42 | 312.74 | 313.06 | 313.39 | 590.00 |

| °C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | °C |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 600.00 | 313.71 | 314.03 | 314.35 | 314.67 | 314.99 | 315.31 | 315.64 | 315.96 | 316.28 | 316.60 | 600.00 |
| 610.00 | 316.92 | 317.24 | 317.56 | 317.88 | 318.20 | 318.52 | 318.84 | 319.16 | 319.48 | 319.80 | 610.00 |
| 620.00 | 320.12 | 320.43 | 320.75 | 321.07 | 321.39 | 321.71 | 322.03 | 322.35 | 322.67 | 322.98 | 620.00 |
| 630.00 | 323.30 | 323.62 | 323.94 | 324.26 | 324.57 | 324.89 | 325.21 | 325.53 | 325.84 | 326.16 | 630.00 |
| 640.00 | 326.48 | 326.79 | 327.11 | 327.43 | 327.74 | 328.06 | 328.38 | 328.69 | 329.01 | 329.32 | 640.00 |
| 650.00 | 329.64 | 329.96 | 330.27 | 330.59 | 330.90 | 331.22 | 331.53 | 331.85 | 332.16 | 332.48 | 650.00 |
| 660.00 | 332.79 | 333.11 | 333.42 | 333.74 | 334.05 | 334.36 | 334.68 | 334.99 | 335.31 | 335.62 | 660.00 |
| 670.00 | 335.93 | 336.25 | 336.56 | 336.87 | 337.18 | 337.50 | 337.81 | 338.12 | 338.44 | 338.75 | 670.00 |
| 680.00 | 339.06 | 339.37 | 339.69 | 340.00 | 340.31 | 340.62 | 340.93 | 341.24 | 341.56 | 341.87 | 680.00 |
| 690.00 | 342.18 | 342.49 | 342.80 | 343.11 | 343.42 | 343.73 | 344.04 | 344.35 | 344.66 | 344.97 | 690.00 |
| 700.00 | 345.28 | 345.59 | 345.90 | 346.21 | 346.52 | 346.83 | 347.14 | 347.45 | 347.76 | 348.07 | 700.00 |
| 710.00 | 348.38 | 348.69 | 348.99 | 349.30 | 349.61 | 349.92 | 350.23 | 350.54 | 350.84 | 351.15 | 710.00 |
| 720.00 | 351.46 | 351.77 | 352.08 | 352.38 | 352.69 | 353.00 | 353.30 | 353.61 | 353.92 | 354.22 | 720.00 |
| 730.00 | 354.53 | 354.84 | 355.14 | 355.45 | 355.76 | 356.06 | 356.37 | 356.67 | 356.98 | 357.28 | 730.00 |
| 740.00 | 357.59 | 357.90 | 358.20 | 358.51 | 358.81 | 359.12 | 359.42 | 359.72 | 360.03 | 360.33 | 740.00 |
| 750.00 | 360.64 | 360.94 | 361.25 | 361.55 | 361.85 | 362.16 | 362.46 | 362.76 | 363.07 | 363.37 | 750.00 |
| 760.00 | 363.67 | 363.98 | 364.28 | 364.58 | 364.89 | 365.19 | 365.49 | 365.79 | 366.10 | 366.40 | 760.00 |
| 770.00 | 366.70 | 367.00 | 367.30 | 367.60 | 367.91 | 368.21 | 368.51 | 368.81 | 369.11 | 369.41 | 770.00 |
| 780.00 | 369.71 | 370.01 | 370.31 | 370.61 | 370.91 | 371.21 | 371.51 | 371.81 | 372.11 | 372.41 | 780.00 |
| 790.00 | 372.71 | 373.01 | 373.31 | 373.61 | 373.91 | 374.21 | 374.51 | 374.81 | 375.11 | 375.41 | 790.00 |
| 800.00 | 375.70 | 376.00 | 376.30 | 376.60 | 376.90 | 377.19 | 377.49 | 377.79 | 378.09 | 378.39 | 800.00 |
| 810.00 | 378.68 | 378.98 | 379.28 | 379.57 | 379.87 | 380.17 | 380.46 | 380.76 | 381.06 | 381.35 | 810.00 |
| 820.00 | 381.65 | 381.95 | 382.24 | 382.54 | 382.83 | 383.13 | 383.42 | 383.72 | 384.01 | 384.31 | 820.00 |
| 830.00 | 384.60 | 384.90 | 385.19 | 385.49 | 385.78 | 386.08 | 386.37 | 386.67 | 386.96 | 387.25 | 830.00 |
| 840.00 | 387.55 | 387.84 | 388.14 | 388.43 | 388.72 | 389.02 | 389.31 | 389.60 | 389.90 | 390.19 | 840.00 |
| 850.00 | 390.48 | | | | | | | | | | 850.00 |
| °C | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | °C |

FIG. A.1 – Table Pt100

B Calcul d'incertitude

Ici nous allons effectuer en détail les calculs d'incertitude pour estimer la précision de la chaîne de mesure. Ceci dans le but de connaître l'incertitudes des mesures de températures envoyées par le PIC. Les calculs seraient allégés en utilisant les dérivées logarithmiques, mais j'ai préféré n'utiliser qu'une seule méthode.

B.1 Données

Les documentations constructeurs nous donnent :

$$R_s = 1 \text{ k}\Omega \text{ } 0,1\% ;$$

$$R = 100 \text{ }\Omega \text{ } 0,1\% ;$$

$R_{Pt100} = (R_0 + \delta R) \text{ }\Omega$ à 0,01%. Pour la suite on choisira $R_0 = R$ pour avoir R_0 à 0,1% et δR à 0,01% ;

Pour une température supérieure à $\theta = 0 \text{ }^\circ\text{C}$, on à $2,498 \text{ V} \leq V_{ref} \leq 2,500 \text{ V}$.

L'OPA335 a++++ un offset d'entrée de $5 \text{ }\mu\text{V}$ et l'amplificateur d'instrumentation (INA118) un offset de $50 \text{ }\mu\text{V}$;

R_g la résistance de gain, est à 0,1%.

Soit T la température et T_0 la température de référence (0,00 $^\circ\text{C}$ dans notre cas). Pour les calculs d'incertitudes pour le cas le plus défavorable, on choisit la température la plus éloignée, $\theta = 30 \text{ }^\circ\text{C}$.

B.2 Les étapes

Incertitude absolue de V_{ref}

$$V_{ref} = \overline{V_{ref}} \pm \Delta V_{ref}$$

$$\boxed{V_{ref} = (2,499 \pm 0,001) \text{ V}} \quad (\text{B.1})$$

Incertitude absolue sur R_s , R et δR

$\Delta R_s = (R_s \times 0,1\%) \text{ }\Omega$. Soit :

$$\boxed{R_s = (1000 \pm 1) \text{ }\Omega} \quad (\text{B.2})$$

De même on trouve

$$\boxed{R = (100 \pm 0,1) \text{ }\Omega} \quad (\text{B.3})$$

Pour calculer l'incertitude sur δR on choisira le cas le plus défavorable, c'est-à-dire la plus grande variation à mesurer. Pour $\delta R(\theta) = 11,67 \Omega$.

D'où $\Delta\delta R = (\delta R(\theta) \times 0,01\%) = 0,0012 \Omega$. Finalement,

$$\boxed{\delta R = (\delta R \pm 0,001) \Omega} \quad (\text{B.4})$$

Incertitude absolue sur I

On a $I = \frac{V_{ref}}{R_s}$

$$dI = \frac{\partial I}{\partial V_{ref}} dV_{ref} + \frac{\partial I}{\partial R_s} dR_s = \frac{1}{R_s} dV_{ref} - \frac{V_{ref}}{R_s^2} dR_s$$

On en déduit l'incertitude relative ΔI :

$$\Delta I = \left| \frac{1}{R_s} \right| \Delta V_{ref} + \left| \frac{-V_{ref}}{R_s^2} \right| \Delta R_s$$

Application numérique : $\Delta I = \left| \frac{1}{100} \right| \times 0,001 + \left| \frac{-2,499}{100^2} \right| \times 1$. D'où :

$$\boxed{I = (2,499 \pm 0,004) mA} \quad (\text{B.5})$$

Incertitude absolue sur V_0

Soient V_A et V_B tels que $V_0 = V_A - V_B$ et I' le courant traversant une sonde. Il faut avant tout déterminer $\Delta I'$.

Incertitude sur I'

$I' = I \frac{2R + \delta R}{4R + 2\delta R} = \frac{I}{2} = 1,2495$. D'où la différentielle suivante :

$$\begin{aligned} dI' &= \frac{\partial I'}{\partial I} dI \\ \Delta I' &= \left| \frac{1}{2} \right| \Delta I = 1,7 \mu A \end{aligned}$$

$$\boxed{I' = (1,250 \pm 0,002) mA} \quad (\text{B.6})$$

Incertitude sur V_A

On a $V_A = I' \times R$

$$\begin{aligned} dV_A &= \frac{\partial V_A}{\partial I'} dI' + \frac{\partial V_A}{\partial R} dR \\ &= R dI' + I' dR \\ \Delta V_A &= |R| \Delta I' + |I'| \Delta R \end{aligned}$$

$$\boxed{\Delta V_A = 0,325 \text{ mV}} \quad (\text{B.7})$$

Incertitude sur V_B

On a

$$V_B = I \frac{2R + \delta R}{4R + 2\delta R} (2R + \delta R) = I \frac{(2R + \delta R)^2}{4R + 2\delta R}$$

$$\begin{aligned} dV_B &= \frac{\partial V_B}{\partial I} dI + \frac{\partial V_B}{\partial R} dR + \frac{\partial V_B}{\partial \delta R} d\delta R \\ &= \frac{(2R + \delta R)^2}{4R + 2\delta R} dI + I \frac{(8R + 4\delta R)(4R + 2\delta R) - 4(2R + \delta R)^2}{(4R + 2\delta R)^2} dR + \\ &\quad I \frac{(4R + 2\delta R)(4R + 2\delta R) - 2(2R + \delta R)^2}{(4R + 2\delta R)^2} d\delta R \\ &= \frac{(2R + \delta R)^2}{4R + 2\delta R} dI + I \frac{16R^2 - 8\delta R^2}{(4R + 2\delta R)^2} dR + I \frac{1}{2} d\delta R \end{aligned}$$

Pour un cas général on obtient donc :

$$\boxed{\Delta V_B = \left| \frac{(2R + \delta R)^2}{4R + 2\delta R} \right| \Delta I + \left| I \frac{16R^2 - 8\delta R^2}{(4R + 2\delta R)^2} \right| \Delta R + \left| \frac{1}{2} \right| \Delta \delta R} \quad (\text{B.8})$$

Dans notre étude, nous allons prendre le cas le plus défavorable, c'est-à-dire $T = \theta$. Soit $\delta R = 11,67 \Omega$.

Application numérique :

$$\boxed{\Delta V_B(\delta R(\theta)) = 1,157 \text{ mV}} \quad (\text{B.9})$$

Incertitude sur V_0

En reprenant le calcul réalisé dans le rapport (page 20) on a :

$$V_0 = \frac{I}{2} \times \delta R$$

Et pour l'incertitude,

$$\Delta V_0 = \Delta V_A + \Delta V_B = 1,482 \text{ mV}$$

D'où

$$\boxed{V_0 = \left(\frac{I}{2} \times \delta R \pm 1,482 \times 10^{-3} \right) \text{ V}} \quad (\text{B.10})$$

Incertitude absolue sur V_s

Gain

Soit V_s le signal disponible en sortie de l'amplificateur d'instrumentation, et R_g sa résistance de gain à 0,1%. La détermination de R_g est issue de la formule disponible dans la documentation constructeur (voir ci-dessous) pour une pleine échelle (0 V–5 V).

$$\boxed{R_g = (150 \pm 0,15) \Omega} \quad (\text{B.11})$$

On a

$$\begin{aligned} G &= 1 + \frac{50 \times 10^3 [\Omega]}{R_g} = 334,33 \\ dG &= \frac{\partial G}{\partial R_g} dR_g = \frac{-200 \times 10^3}{R_g^2} dR_g \\ \Delta G &= \left| \frac{50 \times 10^3}{R_g^2} \right| \Delta R_g = 0,3333 \Omega \end{aligned}$$

Soit

$$\boxed{G = 334,33 \pm 0,3333} \quad (\text{B.12})$$

Calcul de l'incertitude sur V_s

$$\begin{aligned} V_s &= V_0 \times G \\ dV_s &= \frac{\partial V_s}{\partial V_0} dV_0 + \frac{\partial V_s}{\partial G} dG = G dV_0 + V_0 dG \\ \Delta V_s(\theta) &= G \cdot \Delta V_0 + V_0 \cdot \Delta G \end{aligned}$$

Application numérique pour $T = \theta$: $\Delta V_s = 0,5003 \text{ V}$. On en déduit donc que les $5 \mu\text{V}$ d'offset de l'INA118 sont négligeables.

$$\boxed{V_s = \left(G \cdot \frac{I}{2} \cdot \delta R \pm 0,5003 \right) \text{ V}} \quad (\text{B.13})$$

Calcul final

Des calculs précédents, on trouve (avec l'incertitude valable pour une plage 0–30 °C) :

$$V_s = \left(G \cdot \frac{I}{2} \cdot \delta R \pm 0,5003 \right) \text{ V} \quad (\text{B.14})$$

Les tables (annexe A) donnent la résistance standard R_{Pt100} de la Pt100 à la température T , pour une résistance nominale $R_0 = 100,00 \text{ Ohm}$ à la température de référence $T_0 = 0 \text{ °C}$. Dépendance pouvant être approximée par la relation :

$R_{Pt100} = R_0 (1 + A \cdot \Delta T + B \cdot \Delta T^2)$, où : $\Delta T = T - T_0$. Avec $A = 3,9083 \times 10^{-3} K^{-1}$ et $B = -5,775 \times 10^{-7} K^{-2}$. Ce qui donne pour déterminer la température : $T = T_0 + \frac{\sqrt{A^2 + 4B \cdot r} - A}{2B}$, où

$$r = \frac{R_{Pt100}}{R_0} - 1 \quad (\text{B.15})$$

Or on a $R_0 = R$ et $R_{Pt100} = R + \delta R$. Notre variable étant V_s , la tension disponible en entrée du PIC, il ne faut obtenir que V_s pour calculer la température. On l'obtient par la relation (B.14). Soit,

$$\delta R = \frac{2V_s}{G \cdot I}$$

d'où :

$$\begin{aligned} r &= \frac{R + \delta R}{R} - 1 = \frac{\delta R}{R} = \frac{2V_s}{G \cdot I \cdot R} \\ dr &= \frac{\partial r}{\partial V_s} dV_s + \frac{\partial r}{\partial G} dG + \frac{\partial r}{\partial I} dI + \frac{\partial r}{\partial R} dR \\ &= \frac{2}{G \cdot I \cdot R} dV_s - 2V_s \left(\frac{dG}{G^2 \cdot I \cdot R} + \frac{dI}{G \cdot I^2 \cdot R} + \frac{dR}{G \cdot I \cdot R^2} \right) \\ &= \frac{2}{G \cdot I \cdot R} dV_s - 2 \frac{G \cdot I \cdot \delta R}{2} \left(\frac{dG}{G^2 \cdot I \cdot R} + \frac{dI}{G \cdot I^2 \cdot R} + \frac{dR}{G \cdot I \cdot R^2} \right) \\ \Delta r &= \frac{2}{G \cdot I \cdot R} \Delta V_s + \delta R \left(\frac{\Delta G}{G \cdot R} + \frac{\Delta I}{I \cdot R} + \frac{\Delta R}{R^2} \right) \end{aligned}$$

Soit l'application numérique pour le cas le plus défavorable : $\Delta r = 0,0124$
Finalement on obtient :

$$\boxed{T = T_0 + \frac{\sqrt{A^2 + 4B \cdot r} - A}{2B}} \quad (\text{B.16})$$

Avec

$$\boxed{r = \frac{2V_s}{G \cdot I \cdot R} \pm 0,0124} \quad (\text{B.17})$$

Incertitude finale

On reprend $r = \frac{\delta R(\theta)}{R}$ pour calculer l'incertitude sur T :

$$\begin{aligned} T &= T_0 + \frac{\sqrt{A^2 + 4B \cdot r} - A}{2B} \\ dT &= \frac{\partial T}{\partial r} dr = \frac{1}{2B} \cdot \frac{4B}{2\sqrt{A^2 + 4B \cdot r}} dr = \frac{1}{\sqrt{A^2 + 4B \cdot r}} dr \\ \Delta T &= \left| \frac{1}{\sqrt{A^2 + 4B \cdot r}} \right| \Delta r \end{aligned}$$

Application numérique : $\Delta T = 0,066 \text{ } ^\circ\text{C}$

Résultat

Théoriquement, on obtient une mesure fidèle à 0,1 °C (avec une marge de 0,03 °C). Avec pour relations :

$$T = \left(T_0 + \frac{\sqrt{A^2 + 4B \cdot r} - A}{2B} \pm 0,07 \right) ^\circ\text{C} \quad (\text{B.18})$$

$$r = \frac{2V_s}{G \cdot I \cdot R} \pm 0,0124 \quad (\text{B.19})$$

$$A = 3,9083 \cdot 10^{-3} \text{ K}^{-1} \quad (\text{B.20})$$

$$B = -5,775 \cdot 10^{-7} \text{ K}^{-2} \quad (\text{B.21})$$

$$T_0 = 0,00 \text{ } ^\circ\text{C} \quad (\text{B.22})$$

Avec pour le circuit présenté :

$$G = 425,53 \pm 0,4255 \quad (\text{B.23})$$

$$I = (2,499 \pm 0,004) \text{ mA} \quad (\text{B.24})$$

$$R = (100 \pm 0,1) \text{ } \Omega \quad (\text{B.25})$$

C Cartes

C.1 Emetteur

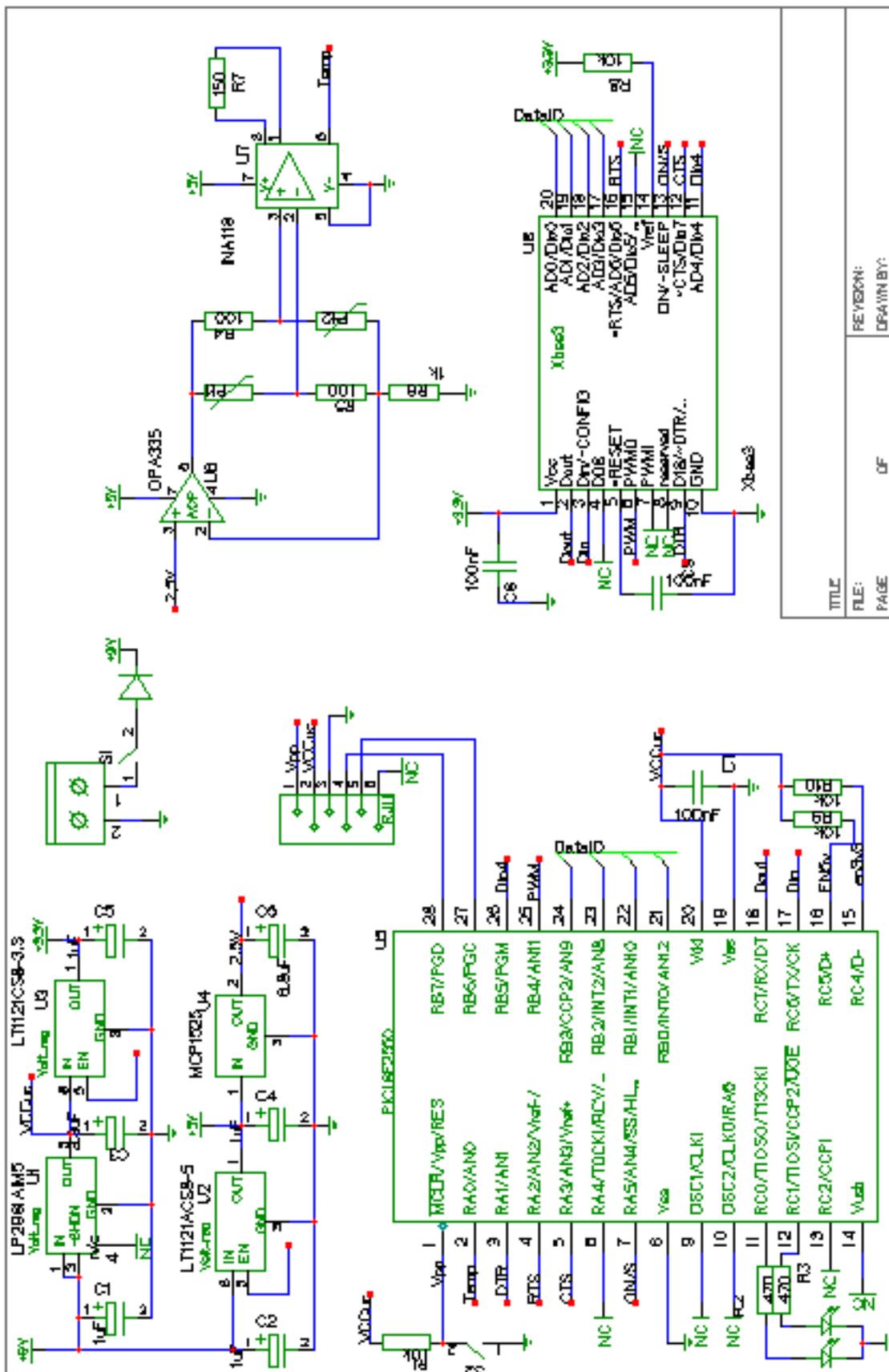
Vous trouverez page suivante le schéma de la carte émettrice. Celle qui mesure la température. Pour un soucis de consommation, plusieurs régulateurs avec "Enable" ont été utilisés pour pouvoir couper l'alimentation de tout ce qui est inutile à certain moment.

Puis les pages suivantes sont directement tirée du fichier PostScript exporté par Pcb. Il s'agit des typons de la carte émettrice.

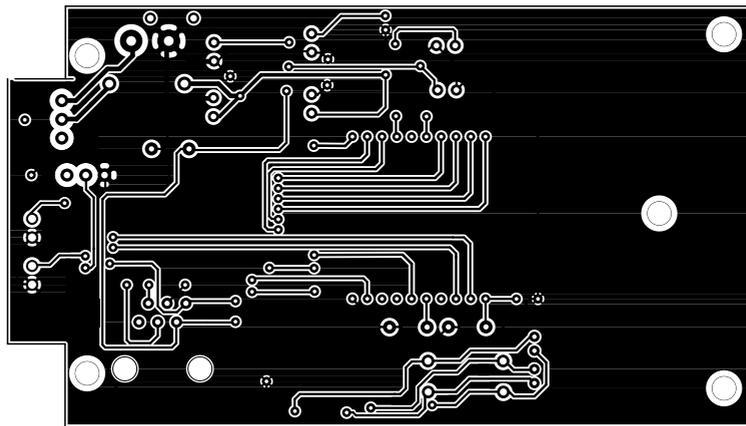
C.2 Récepteur

Ensuite vous trouverez le schéma de la carte récepteur. Bien remarquer le condensateur de 470 nF. C'est par lui qu'est fixé le potentiel 3,3 V pour fixer la vitesse de transmission sur la liaison USB.

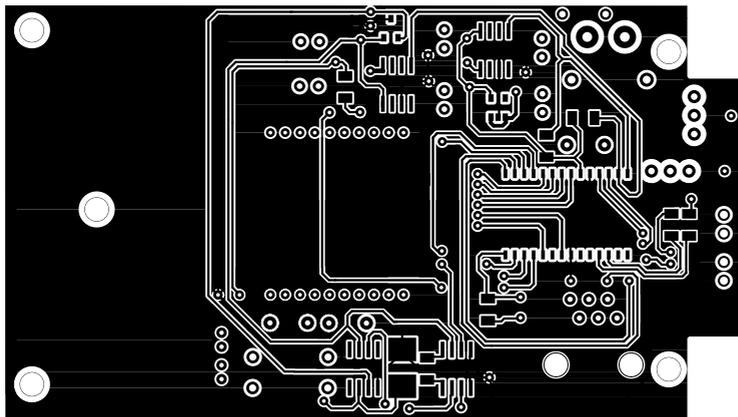
Puis les typons de la carte récepteur exportés par Pcb. 3 feuilles parmi 12 (masque, trou de perçage, ...)



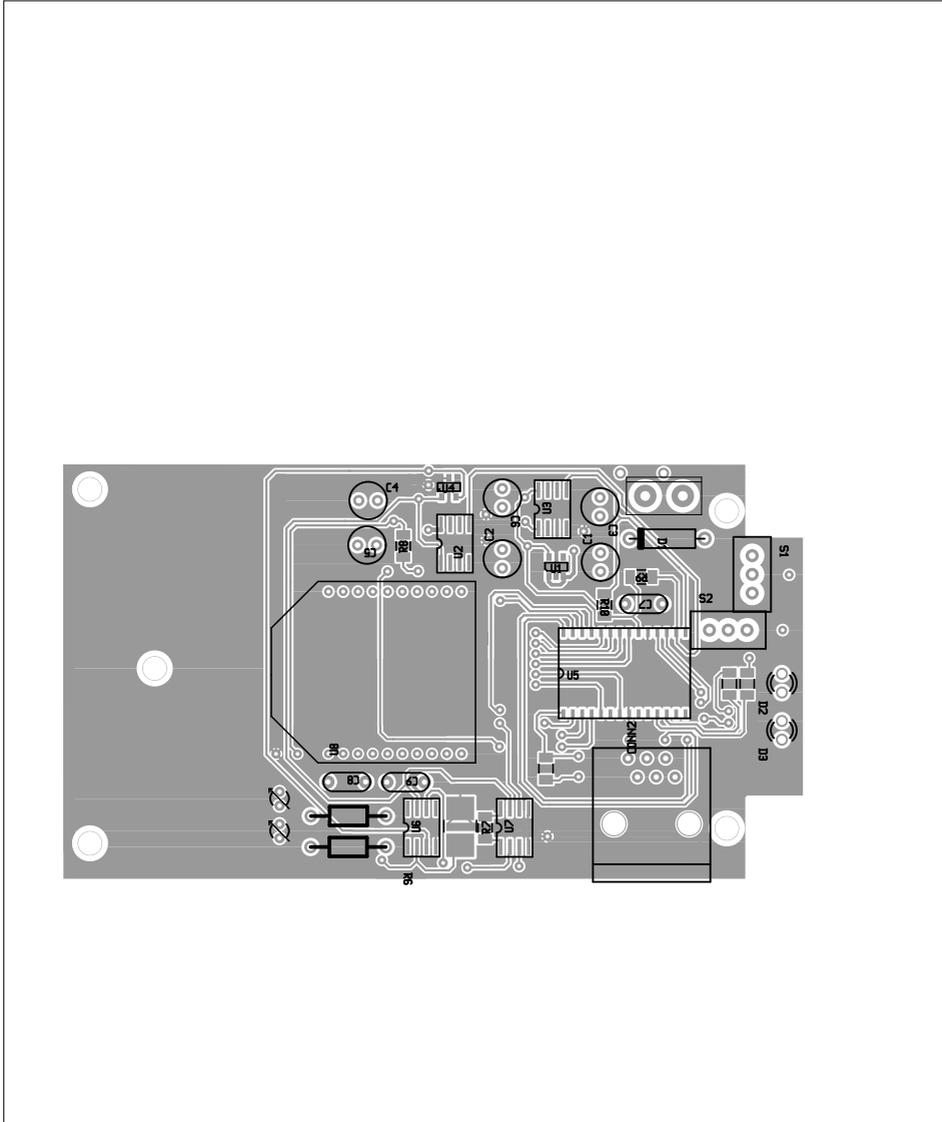
| | | |
|-------|---------|------------|
| TITLE | REV:001 | OF |
| FILE: | | DESIGN BY: |
| PAGE | | |



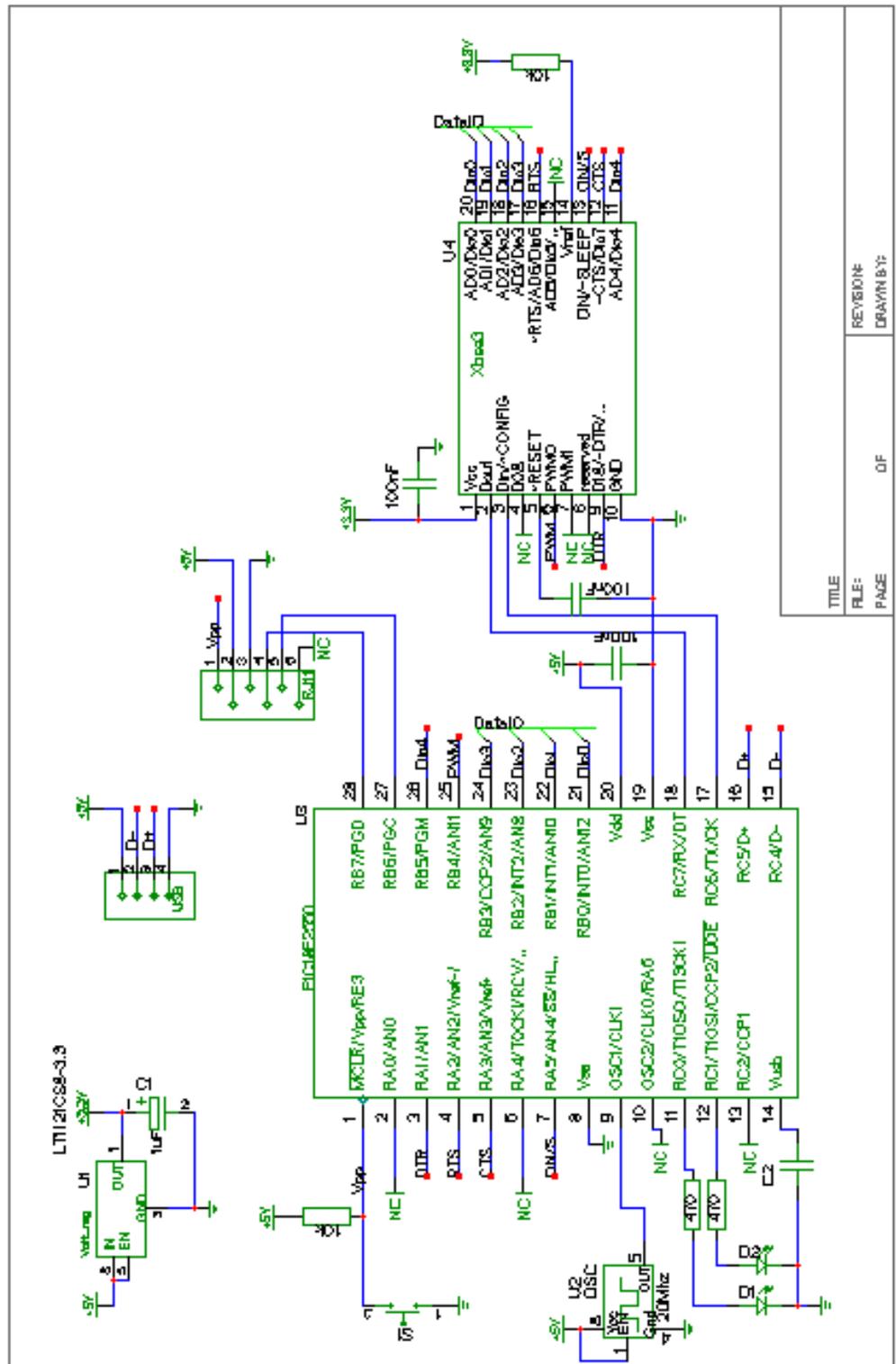
(null), back (mirrored), scale = 1:1.000
/user/kraymond/Emetteur.pcb



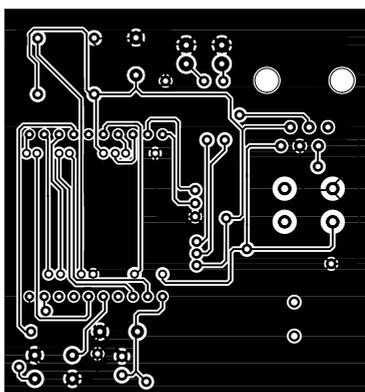
(null), front, scale = 1.000
/user/kraymond/Emetteur.pcb



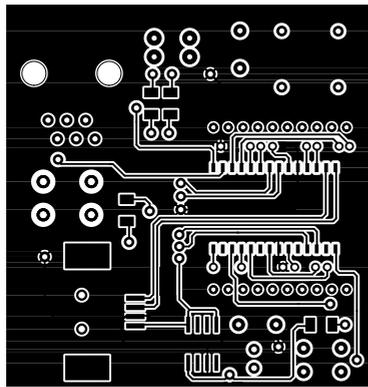
(null), frontassembly, scale = 1:1.000
/user/kraymond/Emetteur.pcb



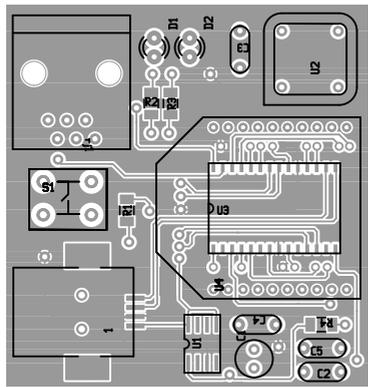
| | |
|-------|-----------|
| TITLE | REVISION: |
| FILE: | DRAWN BY: |
| PAGE | OF |



(null), back (mirrored), scale = 1:1.000
/user/kraymond/Recepteur.pcb



(null), front, scale = 1.000
/user/kraymond/Recepteur.pcb



(null), frontassembly scale = 1:1.000
/user/kraymond/Recepteur.pcb

Article Connexe



KÉVIN RAYMOND
Année 2007

Utilisation de gEDA, suite libre de CAO
Programmation des PIC sous Linux
Utilisation de la carte Mesure de température

Résumé :

Article destiné à aider à prendre en main la suite de CAO gEDA, Piklab pour programmer les PIC sous linux et quelques notes sur l'utilisation de la carte pour mesurer les températures.

Table des matières

| | |
|---|-----------|
| Introduction | 3 |
| 1 gEDA | 4 |
| 1.1 Installation | 4 |
| 1.2 gschem | 6 |
| 1.3 Pcb | 8 |
| 2 Programmation des PIC sous linux | 13 |
| 2.1 Installation | 13 |
| 2.2 Utilisation | 13 |
| 3 Acquisition d'une température | 14 |
| 3.1 Principe de fonctionnement des cartes | 14 |
| 3.2 Le module Xbee | 15 |
| 3.3 Le programme | 16 |
| A Cartes | 18 |
| A.1 Emetteur | 18 |
| A.2 Récepteur | 18 |

Introduction

Cet article est destiné à favoriser la prise en main de gEDA, suite libre de Conception électronique Assisté par Ordinateur. Ce n'est en aucun cas une documentation suffisante, il a pour but de compléter les articles rédigés par les concepteurs.

Nous verrons également de quelle manière programmer les PIC (micro-contrôleurs de chez Microchip) sous Linux en C.

Finalement nous étudierons les cartes électroniques permettant la mesure d'une température avec les sondes Pt100.

1 gEDA

Les logiciels de Conception électronique Assisté par Ordinateur (CAO) propriétaires sont une bonne solution pour débiter rapidement et obtenir simplement des cartes exploitables.

Mais ne serait-il pas plus souhaitable d'utiliser un outil complètement personnalisable avec lequel on n'obtient très rapidement des cartes au rendu professionnel ? Ceci est possible grâce aux logiciels libres. gEDA en est idéal pour la CAO. C'est un outil en plein développement mais d'ores et déjà évolué.

De nombreuses bibliothèques sont disponibles et il est très rapide d'en ajouter. Il en existe deux mécanismes. Le premier appelé « Oldlib » « pcblib » ou « bibliothèque M4 » dépend du langage macro M4. Les empreintes sont générées à la volée. À utiliser pour créer rapidement des familles d'empreintes. Le deuxième mécanisme est appelé « Newlib ». Ce sont les empreintes que l'on peut créer graphiquement avec Pcb, ou avec un éditeur de texte (code ASCII).

gEDA pour GPL Electronic Design Automation est une suite Open Source permettant la création de schémas, la simulation et la réalisation du Printed Circuit Board (PCB).

Cette suite d'outils est principalement composée de :

- **gschem**, l'éditeur de schémas et de symboles,
- **gnetlist**, un translateur vers d'autres utilitaires,
- **PCB**, un outil de dessins de circuits imprimés et de création d'empreintes,
- **ngspice**, un clone de spice avec des fonctions étendues,
- **gnucap**, un simulateur original avec compilateur de modèles,
- **geda**, le gestionnaire de projet (non actuel).

Attention à l'utilisation de geda le gestionnaire de projet. Il n'a pas suivi l'évolution des autres programmes, il est donc préférable de ne pas l'utiliser. Un terminal suffit pour relier les différents outils.

1.1 Installation

Avant tout il faut l'installer. La dernière version de gEDA nécessite python 2.5 ou supérieur. Il est possible que gEDA, ainsi que toutes ses dépendances, soit dans les dépôts de votre distribution (voir par exemple avec Synaptic). Mais ce ne sera peut-être pas la dernière version. Pour cette suite

en constante évolution il est important d'avoir la dernière version, pour éviter certains bogues.

Une autre possibilité est de télécharger l'image iso du CD. Elle inclut la dernière version de tous les programmes. Ainsi que les dépendances nécessaires. Voir dans la partie download du site officiel : <http://www.geda.seul.org/>. La solution la plus simple est de monter le cd sur un lecteur virtuel. Très simple sous linux.

```
$ sudo mount -o loop <image>.iso /mnt
```

Pour poursuivre l'installation l'aide nécessaire se trouve sur le CD. En ce qui concerne les tutoriaux, attention à bien lire les dernières versions.

Une fois l'installation terminée, il faut avoir le synoptique de la suite logicielle en tête pour commencer une carte.

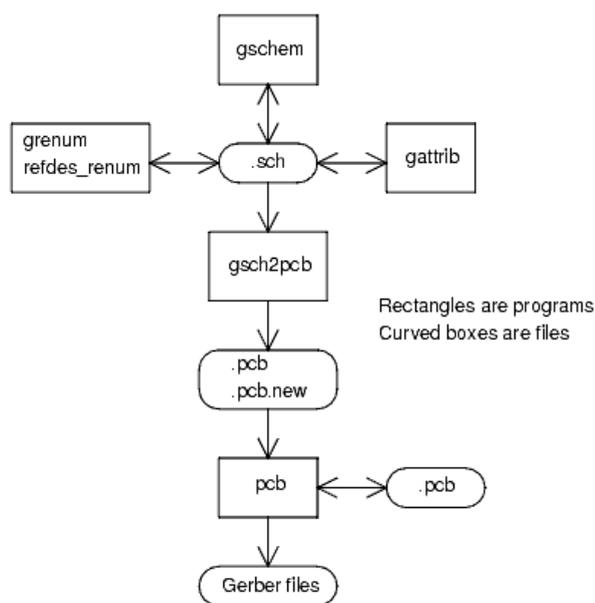


FIG. 1 – Synoptique de la suite gEDA

Tout commence avec gschem. Une fois le schéma réalisé et vérifié avec le contrôleur DRC il faut créer la netliste. Grenum sert à assigner une référence

à tous les composants. Ce qui peut être réalisé automatiquement avec *gschem*. De même *gattrib* permet d'assigner des empreintes aux composants. Mais il est plus évident au début de le faire sous *gschem*. *gsch2pcb* permet de créer la netlist et le fichier *.pcb. *Pcb* permet de placer les composants et de router les connexions générées par la netlist. Enfin, les fichiers Gerber sont exportés par *Pcb*.

1.2 *gschem*

Il est conseillé de créer un dossier de projet, car de nombreux fichiers seront créés. Une fois *gschem* ouvert (commande « *gschem* »), vous verrez une interface simple d'utilisation : seulement quelques icônes sont dans la barre d'outils. Possibilité d'ajouter des composants, de créer des fils de connexion voire des bus de données. Pour tourner les composants ou toute autre tâche utile, il est bien plus rapide de se servir des raccourcis.

En vue de simplifier la génération de la netlist, veuillez à bien renseigner au minimum trois champs importants pour *gschem*.

Double-cliquer sur un composant pour ouvrir ses propriétés. En général le champ **device** et **refdes** est rempli. Le champ de loin le plus important est **footprint**, il permet de spécifier quelle empreinte prendre pour ce composant. N'importe quel symbole (ou composant) peut appeler n'importe quelle empreinte. Cela permet de créer une seule empreinte pour plusieurs composants. Ils seront différenciés sous *Pcb* par leur identifiant *device* et *refdes*. Une fois les composants ajoutés, il est possible de renseigner automatiquement tous les champs *refdes* par « Attributs -> Annotation automatique ». Les champs *slot* et *numslot* permettent de spécifier le numéro de la porte lorsqu'il y en a plusieurs par boîtier.

Lors de la création de symboles il faut spécifier le champ **pintype** pour que le vérificateur d'erreurs (DRC) puisse vérifier les erreurs de connexions. Les différentes options pour ce champ sont « *io* » pour une entrée-sortie, « *pas* » pour une broche passive, « *in* » pour une entrée, « *out* » pour une sortie et « *pwr* » pour une alimentation.

Pour spécifier les broches non connectées il faut ajouter le symbole de non connexion. Il se trouve dans *share/gEDA/sym/misc/*.

Pour lancer le vérificateur d'erreurs (ou DRC), le code est le suivant :

```
$ gnetlist -g drc2 text.sch -o drc_output.txt
```

le résultat de DRC est visible dans le fichier `drc_output.txt`.

Personnellement, je trouve plus simple de lire le résultat directement dans la console :

```
$ gnetlist -g drc2 text -o -
```

Si vous rencontrez une erreur du type

```
Checking slots ...  
ERROR : Reference U5 : Slot out of range (1).  
slotnumber = 1
```

Vous avez sûrement une empreinte avec un nom finissant par une minuscule. Ce qui est interprété comme le numéro d'une porte logique à l'intérieur d'un composant. S'il ne s'agit pas d'une erreur il est possible de passer outre en spécifiant le champ **numslot** à 0.

La netlist est créée en même temps que le rapport du DRC, il faut donc maintenant invoquer `gsch2pcb` pour commencer le routage.

```
$ gsch2pcb -v text.sch
```

L'option verbose (`-v`) est facultative, mais permet de suivre les différentes erreurs dues principalement à des empreintes introuvables. La double option verbose (`$ gsch2pcb -v -v text.sch`) permet de voir tous les fichiers scannés pour trouver les différentes empreintes. Lorsqu'il n'y a plus d'erreurs, il faut passer sur Pcb. Le fichier `text.pcb` est créé par `gsch2pcb`. Il faut maintenant l'ouvrir avec Pcb.

Je précise que les principales démarches à suivre sont rappelées par `gsch2pcb` à la fin du rapport. Lors d'une première exécution on peut lire par exemple :

Next step :

1. Run `pcb` on your file `test.pcb`. You will find all your footprints in a bundle ready for you to place or disperse with « Select -> Disperse all elements » in PCB.
2. From within PCB, select « File -> Load netlist file » and select `test.net` to load the netlist.
3. From within PCB, enter `:ExecuteFile(test.cmd)`

Lors d'une mise à jour (le fichier `test.pcb` existe déjà) d'autres conseils apparaissent :

Next steps :

1. Run pcb on your file test.pcb.
2. From within PCB, select « File -> Load layout data to paste buffer » and select test.new.pcb to load the new footprints into your existing layout.
3. From within PCB, select « File -> Load netlist file » and select test.net to load the updated netlist.
4. From within PCB, enter :ExecuteFile(test.cmd)

Dans tous les cas la prochaine étape est Pcb.

1.3 Pcb

Je conseille d'ouvrir directement Pcb par un autre terminal puisque le programme retourne un certain nombre d'erreurs (notamment « unknown action » pour une commande inconnue) . Procéder par `$ pcb test.pcb` tout en étant dans le répertoire du projet.



Pcb ouvre automatiquement une fenêtre de log qui retourne les différentes actions de l'utilisateur prises en compte par le programme.

Une fois le fichier test.pcb ouvert il faut réorganiser les composants (Select -> Disperse all elements). Il ne reste plus qu'à importer la netlist (File -> Load netlist file). Ensuite lancer le fichier test.cmd par l'invite de commande (le raccourci pour cette action est [:]). `ExecuteFile(test.cmd)`. Puis remettre à jour. Action possible par la touche de raccourci [O]. Si les fils de connexions (ratsnet) ne sont pas visibles, activer l'option en cliquant sur le bouton « ratsnet » (à gauche).

Pour mettre à jour le PCB si un changement doit apparaître (modification sous gschem des connexions ou composants par exemple), lancer le DRC pour vérifier et régénérer la netlist, ce qui a pour effet de créer un fichier test.pcb.new. Ouvrir ce fichier par « File -> Load layout-data to past buffer ». Recharger la netlist, lancer la commande `ExecuteFile(test.cmd)` et réactualiser les fils de connexions ([O]).

Pour une prise en main assez rapide de Pcb :

Dans la barre d'outils verticale (sur la gauche) il est possible de sélectionner l'affichage des plans, via, pistes etc. Par défaut il existe plusieurs couches avec une couleur pour chaque signal (alimentation, données ...). Il est possible de modifier ces options par « File -> Preference ».

En dessous sont représentés les outils de dessins. Line pour les pistes, polygon pour les plans de masses par exemple, LOCK pour verrouiller un composant. Utile par exemple pour les plans ou les gros composants, cela évite de le sélectionner alors que l'ont visait une piste. L'outil THRM permet rapidement d'ajouter ou d'enlever des freins thermiques aux pastilles.

Pour déplacer un composant avec ses pistes (ou ratsnet), le clic souris suffit. Parfois il est préférable de déplacer un composant sans ses connexions. Pour cela il faut le sélectionner (clic souris) puis le déplacer par un autre clic souris.

Plusieurs actions sont disponibles par [shift]+Clic droit. Notamment supprimer l'ensemble de la sélection. La touche [Shift] lors d'un tracé de piste sert à modifier la position de l'angle. Très pratique!

Pour une mesure rapide [CTRL]+[M] ré-affecte l'origine (les mesures sont en haut à droite). Pour avoir la vue de la face de dessous il suffit de presser la touche [TAB]. Alors que [B] permet de passer le composant d'une face à l'autre.

[U] pour « Undo » (annuler) et [Shift]+[R] pour « Redo » (restaurer).

[Q] permet de modifier l'apparence des pastilles (rondes ou carrées). [D] permet d'afficher rapidement le numéro des broches et [shift]+[D] en donne l'aperçu.

[O] qui permet de réactualiser les connexions est souvent utilisée de même que [F] qui permet de mettre en surbrillance les connexions directes. [S] et [shift]+[S] servent à modifier la taille des pastilles, pistes, etc.

[V] permet d'avoir une vue globale de la carte. Pour le reste des raccourcis utilisés, voir dans « Window -> Key References ».

Attention à bien activer l'option « Settings -> New lines, arcs clear polygon » si vous devez faire un plan de masse ou autre par la suite. À défaut vous retrouverez toutes vos pistes court-circuitées par le plan. Attention cela pose problème pour les freins thermiques. Citation : « à consommer avec modération ».

Cependant il est toujours possible d'activer l'option à la fin, un travail assez rébarbatif. Dans les tutoriaux on peut trouver la commande « :changejoin(selected) » pour changer l'option sur toutes les pistes sélectionnées, mais cela n'avait aucun effet sur ma version. Découvert par le « unknown ac-

tion » retourné dans le terminal. La touche raccourci pour le même résultat est [shift]+[J]. Il est possible de modifier la clearance des pistes (espace vide entre la piste et le plan) par la touche [K] (et [shift]+[K]).

Lorsque vous avez le programme en main, il est très rapide de router ses cartes puisque le logiciel permet d'un clic de copier des éléments dans un tampon (éléments précédemment créés). On peut donc avoir à portée de clic des dessins souvent utiles (logo, empreinte de frein thermique, etc.)

Créer des empreintes (Footprint) :

Partir d'un élément existant est le plus rapide. Il est possible (et très utile, pour vérification notamment) de créer des empreintes directement sur notre PCB.

Pour ce faire, ouvrir le gestionnaire de librairie par « Window -> library ». Ajouter le composant le plus ressemblant. Le sélectionner pour ensuite le copier dans le buffer ([CTRL]+[X]). Maintenant il faut le convertir en « pièce » : « Buffer -> Convert buffer to pieces ». Coller cette pièce.

[N] permet de numéroter les broches. Une fois toutes les modifications effectuées, copier la pièce dans le buffer puis « Buffer -> Convert buffer to elements » pour enfin l'enregistrer « Buffer -> Save buffer to File ».

Noter que les pastilles « carrées » sont obtenues par des lignes et ensuite modifiées avec la touche [Q] (une fois converties en élément).

Attention, si le nom des empreintes contient une minuscule comme dernier caractère, elle sera considérée comme une porte à l'intérieure d'un autre composant (souvent nommée a, b, c ...) Elles seront ignorées. Il faut mettre un chiffre ou une majuscule.

Pour plus d'informations :

<http://ronja.twibright.com/guidelines/footprints.php>, pour une plus grande explication sur la création d'empreintes graphiques (par Pcb).

http://www.brorson.com/gEDA/land_patterns_20050129.pdf, un tutoriel pour dessiner ses empreintes à la main, avec un éditeur de texte.

Une fois le PCB terminé, d'un clic on peut exporter la carte en fichier Gerber, png, ps, eps ...

Une des plus grandes richesses de gEDA est le traitement des données entre les différents programmes. Tous les fichiers sont en ASCII ce qui les rend très facile à manipuler. Il est donc possible de se créer des scripts pour automatiser des tâches rébarbatives. Habituellement le Perl est utilisé pour

cela. Voici les scripts distribués par gEDA :

1. John Luciani possède un large éventail de scripts disponibles sur son site we (<http://www.luciani.org/>). Dans cette collection, des scripts sont inclus pour générer des empreintes, notamment Footgen.
2. David Rowe possède des scripts pour mettre à jour des éléments de même qu'ajouter/supprimer des fichiers PCB les uns des autres sur son site web (<http://www.rowetel.com/perl4pcb.html>).
3. Stuart Brorson a écrit un script simple qui génère des empreintes pour deux ponts thermiques passifs en SMD. Un tarball gzippé est disponible à <http://www.brorson.com/gEDA/Smtgen.pl.gz>.

Une fois en main, gEDA est à la fois simple et rapide à utiliser. De plus beaucoup de bibliothèques sont disponibles sur <http://www.gedasymbols.org/>. Il est également possible de demander sur la liste des gEDA-user si ce composant est disponible. Pour cela voir <http://geda.seul.org/maillinglist/index.html>.

Pour la recherche de bibliothèques, suivre les liens précédemment cités ou <http://www.gedasymbols.org/>. On en apprend également beaucoup sur <http://www.geda.seul.org/wiki/geda:usage.fr>.

Sur divers sites internet on trouve des tutoriaux pour gEDA, mais en général ils ne sont pas à jour. Tout le problème est là. Les plus intéressants sont les tutoriaux des logiciels et non de la suite. Eux sont à jour. Je pense par exemple au tutorial de Pcb disponible sur <http://pcb.sourceforge.net/manual.html>. Pour comprendre un peu mieux la gestion d'un projet sous gEDA il peut être utile de lire le tutorial en Français disponible là : <http://www.iznogood-factory.org/pub/gEDA/tutorialfr.html>. Il n'est pas basé sur la dernière version, car certaines configurations ont changé. Sur ce site (http://ofset.sourceforge.net/freeduc/book/book_29.html) est disponible un exemple rapide de prise en main de Pcb. En lisant plusieurs tutoriaux dont la version n'est plus à jour, il est plus simple de voir ce qui est toujours d'actualité. Pour une aide à la création d'empreinte, voir : http://www.iznogood-factory.org/pub/gEDA/land_patterns_fr.pdf.

Ce qui peut perturber au début c'est la profusion de raccourcis. Tant pour gSchem que pour Pcb. Attention à ne pas tester toutes les touches sans vérifier au risque de voir plus tard que la grille a été modifiée, que le pas n'est plus en inch mais en millimètres, etc. À noter également que les raccourcis sont différents suivant les programmes.

Le seul inconvénient notable de gSchem est la fonction « undo ». Lorsque

l'on annule les dernières opérations, le zoom est pris en compte.

gEDA est donc une suite très complète. Sa principale difficulté de prise en main repose sur le fait que les tutoriaux ne sont pas à jour. Il faut préférer l'aide de Pcb.

Néanmoins cela reste une suite pour les concepteurs ne souhaitant pas se préoccuper d'une license payante et souhaitant un travail de professionnel. Pour obtenir ce résultat il faut bien entendu y passer un peu de temps, mais qui sera largement récompensé par la suite en efficacité. De plus, il y a réellement une grande bibliothèque de composants rapidement utilisables. Cela peut surprendre au début d'enregistrer des empreintes avec l'éditeur de texte, mais rien n'est plus facile que de copier, voire modifier du texte.

2 Programmation des PIC sous linux

MPLAB, l'environnement distribué par Microchip, est disponible sous Windows. Pour linux, la possibilité est d'utiliser Wine, mais il est préférable d'utiliser les logiciels libres quand ceux-ci sont performants.

C'est le cas de Piklab, auquel on intègre le compilateur, le créateur de lien, le simulateur, le debugger, etc.

2.1 Installation

Également disponible dans la plupart des dépôts, la dernière version est à télécharger sur <http://piklab.sourceforge.net/download.php>.

Pour la programmation des PIC, le langage utilisé pour ce projet est le C. Small Device C compiler est un outil libre et performant. Également disponible sous Windows, il permet de partager plus facilement les sources et donc de trouver des exemples utilisables.

La dernière version de SDCC est disponible là : <http://sdcc.sourceforge.net/index.php#Download>.

Le programmeur utilisé est le In Circuit Debugger 2 de chez Microchip. Il permet en plus de la programmation par port série ou USB d'utiliser la fonction debugger.

Pour l'utiliser il faut installer un package supplémentaire (icd2prog). Une version est disponible ici : <http://kevin.raymond.free.fr/Stage/icd2prog-0.3.0.tar.gz>. Suivant les dépendances, il peut ne pas être reconnu. Une simple installation de MPLAB sous Wine règlera le problème.

2.2 Utilisation

Sous Piklab, lors de la création d'un nouveau projet, il faut spécifier le compilateur et le programmeur. Par défaut beaucoup sont proposés. Nous utiliserons SDCC avec l'ICD2.

Pour l'ICD2 il ne faut pas activer l'option "Low voltage programmation".

Si tout ce passe bien, une fois connecté un rapport avec les différentes tensions est affiché. Exemple :

Vpp du programmeur = 12.4613 V

Vdd de la cible = 5.00224 V

Vpp de la cible = 12.4613 V

Etant très intuitif, cet environnement est très rapide à prendre en main.

3 Acquisition d'une température

3.1 Principe de fonctionnement des cartes

Le système est basé sur deux cartes différentes. L'une mesurant la température à une fréquence variable et l'autre les recevant pour les transmettre à un ordinateur.

La liaison entre ces deux cartes est sans-fils (voir 3.2) et la gestion de l'alimentation est importante pour se module autonome.

La mesure d'une température est effectuée pour plus de précision avec une résistance métallique de platine, une Pt100.

Le schéma est le suivant :

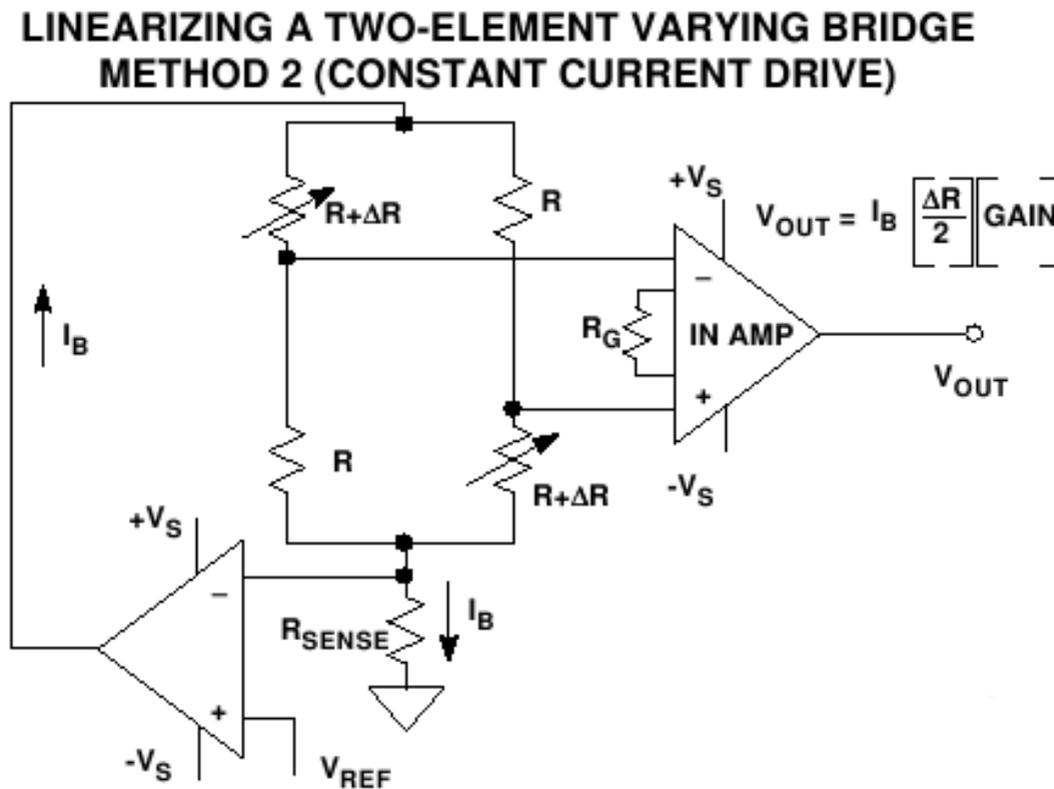


FIG. 2 – Circuit conditionneur

La source de courant est basée sur l'amplificateur OPA335, avec en entrée une référence de tension de 2,5 V (MCP1525). Tous les éléments apparaissant sur ce schéma doivent être au maximum indépendants de la température. Les résistances sont choisies avec une précision d'au moins 0,1 %.

Les résistances du pont de Wheatstone sont de 100,00 Ohm pour avoir un équilibre à 0,00 °C. Il est composé de deux éléments variants (des Pt100) pour avoir une erreur de linéarité nulle.

La carte émettrice est composée de quatre régulateurs. Le premier est un régulateur 5 V pour alimenter le PIC. Il est suivi d'un régulateur 3,3 V pour alimenter le module Xbee. Ce dernier régulateur dispose d'une broche de validation contrôlée par le PIC. De cette manière, le PIC décide ou non d'alimenter le module Xbee.

En parallèle est positionné un régulateur 5 V dédié à l'alimentation des amplificateurs unipolaires pour la mesure de température. Lui aussi dispose d'une entrée de validation connectée au PIC. Il est suivi de la référence de tension 2,5 V. En arrêtant ce régulateur 5 V, le PIC décide d'éteindre toute la chaîne de mesure.

De part cette alimentation, en quelques instants seulement il ne reste plus que le PIC d'alimenté. Celui-ci disposant d'une fonction « veille », lors d'une attente de mesure, la consommation est minime. De l'ordre de quelques dizaines de micro-volts.

Sur les différentes cartes, beaucoup d'entrées/sorties du module Xbee ont été connectées. Ceci simplement pour une utilisation future.

D'indispensable il n'y a pour l'instant que les broches suivantes : DIN et DOUT pour la liaison série et les commandes AT, contrôlant le module ; RTS et CTS pour contrôler les tampons d'émission et de réception ; SLEEP permettant de mettre en veille le module, associée à SLEEP_RQ pour vérifier l'état du module.

En ce qui concerne l'utilisation à proprement parler, l'interrupteur permet de couper l'alimentation et le bouton poussoir à effectuer un Reset du PIC. Les led pourront avoir une utilité suivant la programmation.

Le schéma ainsi que les typons sont disponibles en annexe.

3.2 Le module Xbee

Il s'agit d'un module Zigbee développé par MAXSTREAM. Il a l'avantage d'être contrôlé par commandes AT en liaison série. La reprogrammation du firmware s'effectue par le programme X-CTU et la liaison série du module. Prévoir les connexions de CTS, DTR et RTS.

Sa simplicité d'utilisation est basée sur le fait que le microcontrôleur dialogue par liaison série. Les modules se reconnaissent dès la mise en route,

ce qui permet de commencer très rapidement à étudier le fonctionnement en détail. Il est possible d'assigner des adresses aux différents modules, avec un dialogue bidirectionnel sur plusieurs canaux.

Il est possible d'améliorer la carte pour pouvoir reprogrammer le PIC par le module Xbee. De cette manière il serait possible de reprogrammer toutes les cartes en réseau très rapidement. Un domaine déjà exploré par des utilisateurs sur le forum :<http://www.planete-sciences.org/forums/> section robotique.

3.3 Le programme

Le signal V_s fonction de la température, issue de l'amplificateur, est connecté à la broche RA0 du PIC. Elle est configurée en entrée analogique (AN0).

Pour récupérer la valeur de V_s , il suffit d'effectuer ces quelques lignes :

```
1 unsigned char getTemp()
2 {
3     // Début de conversion
4     ADCON0bits.GO = 1;
5
6     // On attend la fin de la conversion
7     while (ADCON0bits.GO);
8
9     // On retourne les bits de données
10    return ADRESH;
11 }
```

Le calcul de la température s'effectue par les relations suivantes :

$$T = \left(T_0 + \frac{\sqrt{A^2 + 4B \cdot r} - A}{2B} \pm 0,07 \right) C \quad (1)$$

$$r = \frac{2V_s}{G \cdot I \cdot R} \pm 0,0124 \quad (2)$$

$$A = 3,9083 \cdot 10^{-3} K^{-1} \quad (3)$$

$$B = -5,775 \cdot 10^{-7} K^{-2} \quad (4)$$

$$T_0 = 0,00 C \quad (5)$$

$$G = 425,53 \pm 0,4255 \quad (6)$$

$$I = (2,499 \pm 0,004) mA \quad (7)$$

$$R = (100 \pm 0,1) \Omega \quad (8)$$

$$R = (100 \pm 0,1) \Omega \quad (9)$$

La suite concernant l'utilisation des modules Xbee et de la liaison USB reste à rédiger. Cependant, sur internet beaucoup d'exemples sont disponibles.

A Cartes

A.1 Emetteur

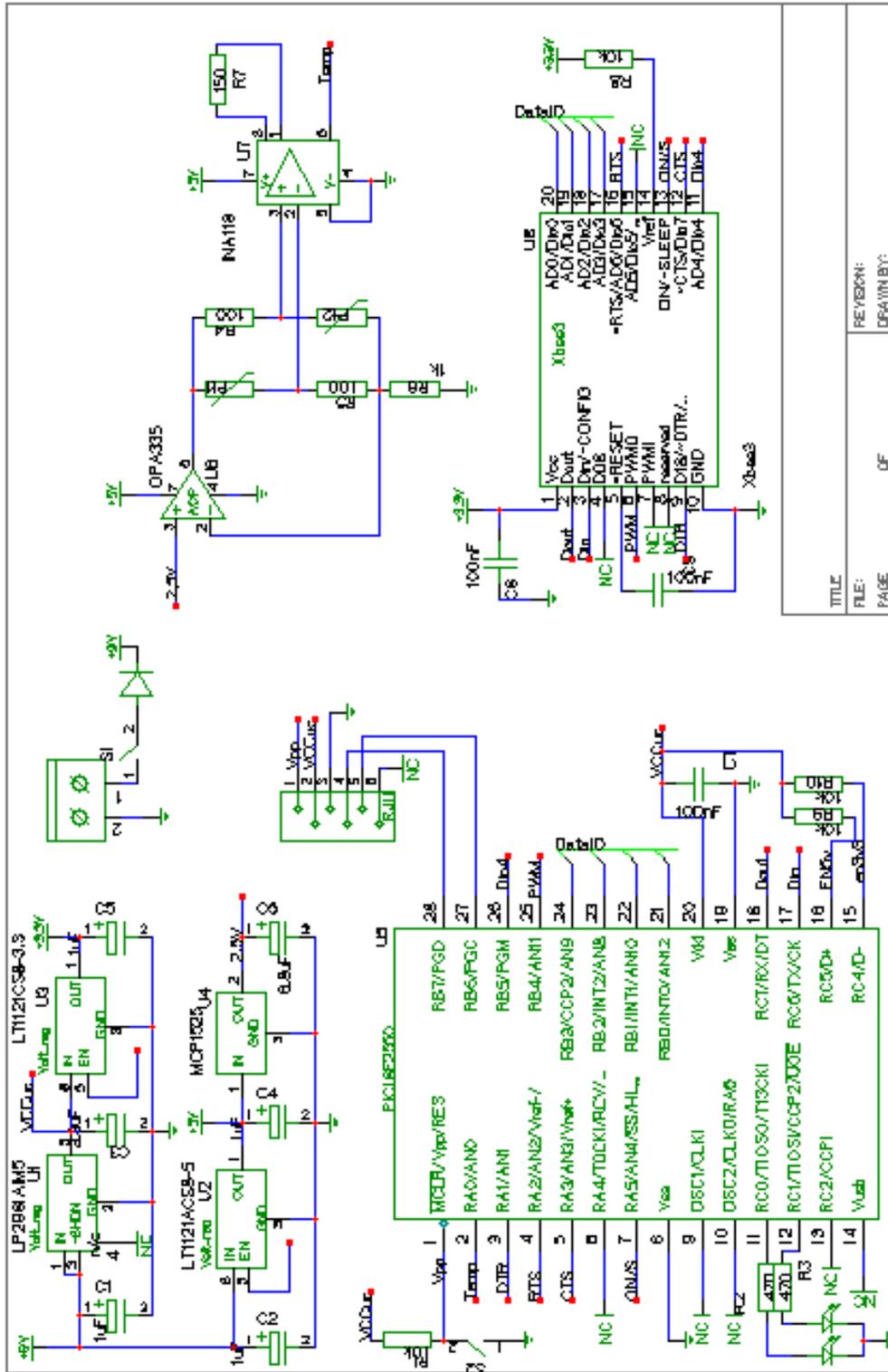
Vous trouverez page suivante le schéma de la carte émettrice. Celle qui mesure la température. Pour un soucis de consommation, plusieurs régulateurs avec "Enable" ont été utilisés pour pouvoir couper l'alimentation de tout ce qui est inutile à certain moment.

Puis les pages suivantes sont directement tirée du fichier PostScript exporté par Pcb. Il s'agit des typons de la carte émettrice.

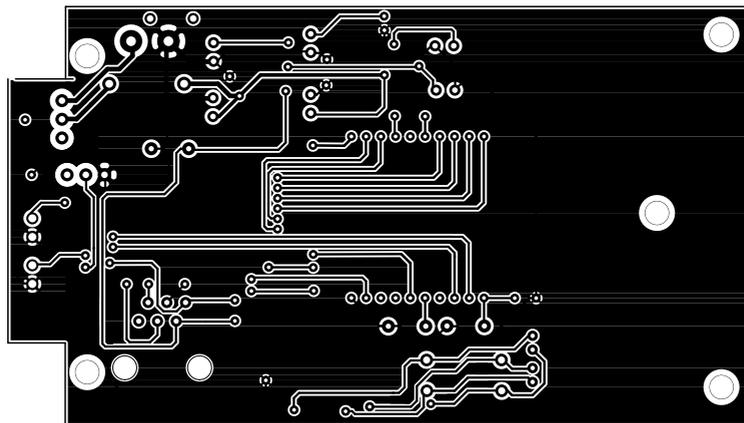
A.2 Récepteur

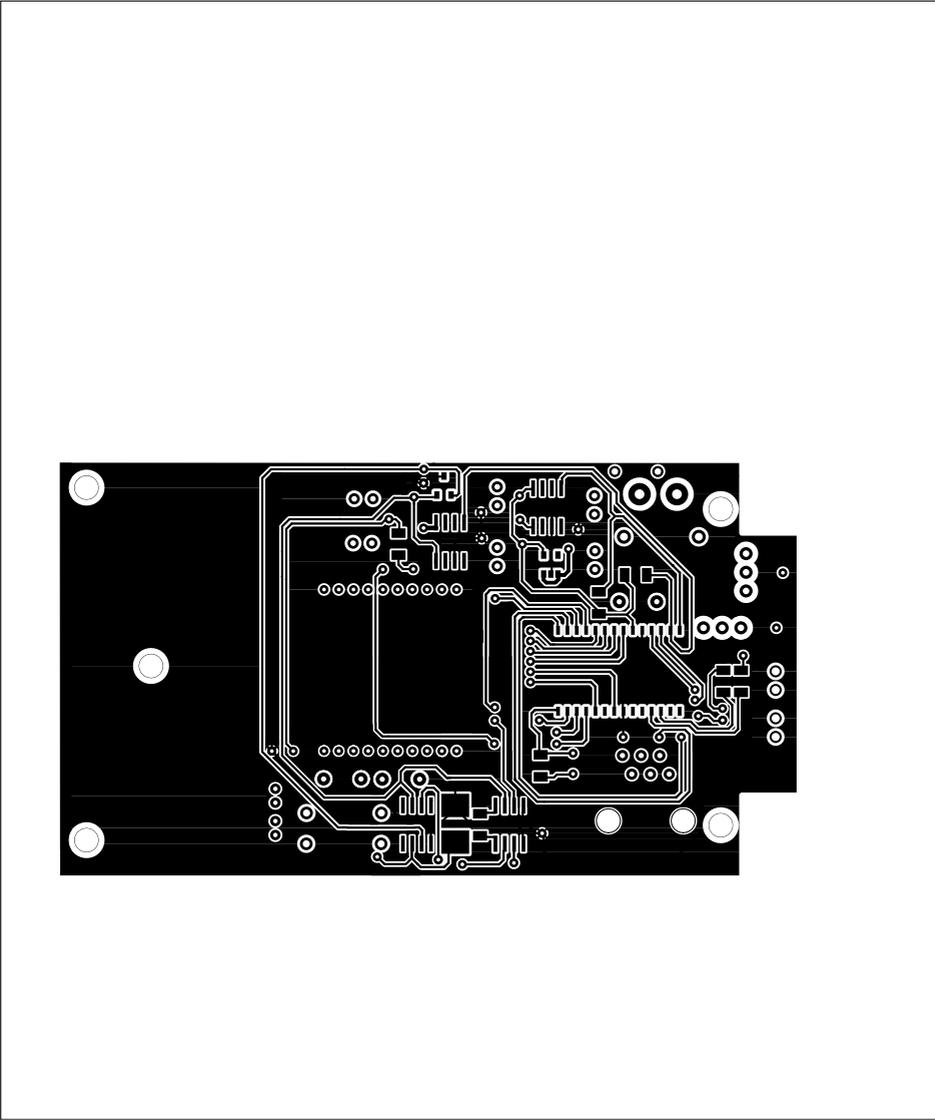
Ensuite vous trouverez le schéma de la carte récepteur. Bien remarquer le condensateur de 470 nF. C'est par lui qu'est fixé le potentiel 3,3 V pour fixer la vitesse de transmission sur la liaison USB.

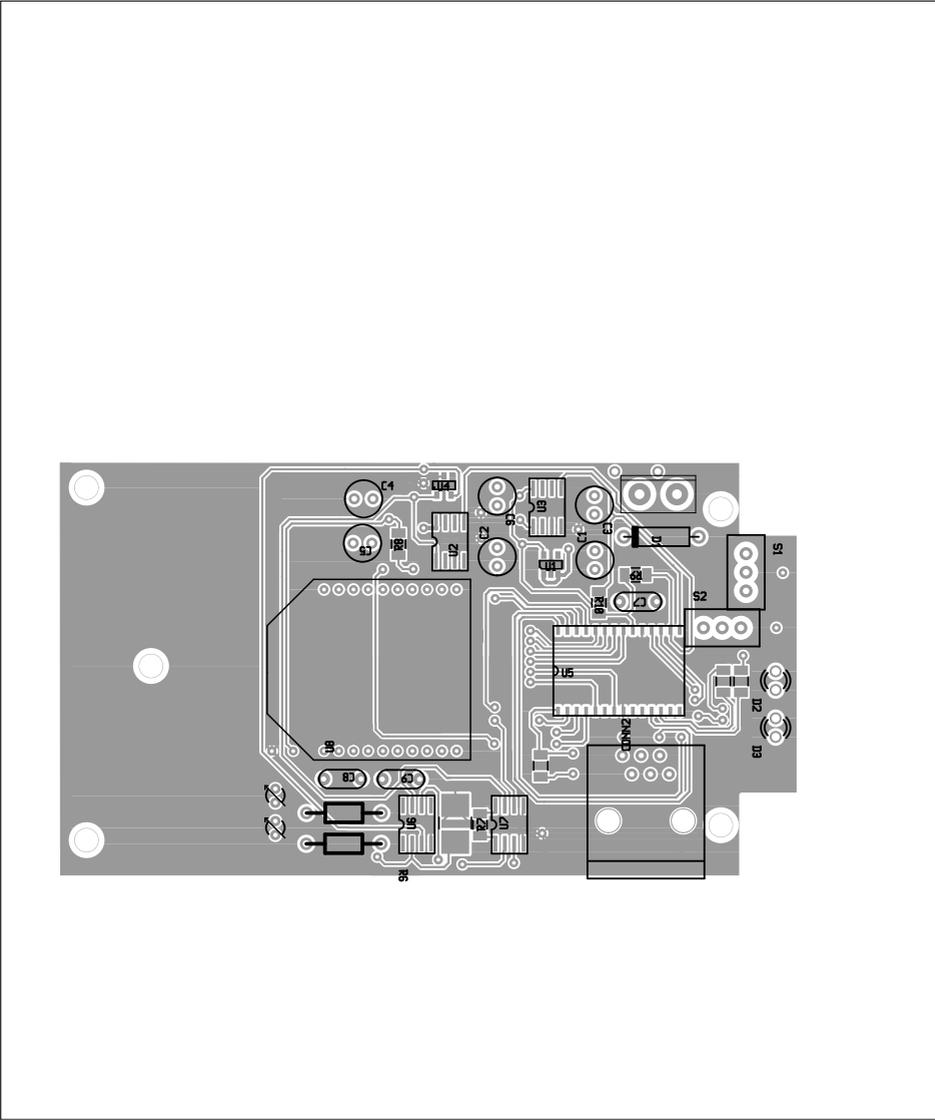
Puis les typons de la carte récepteur exportés par Pcb. 3 feuilles parmi 12 (masque, trou de perçage, ...)

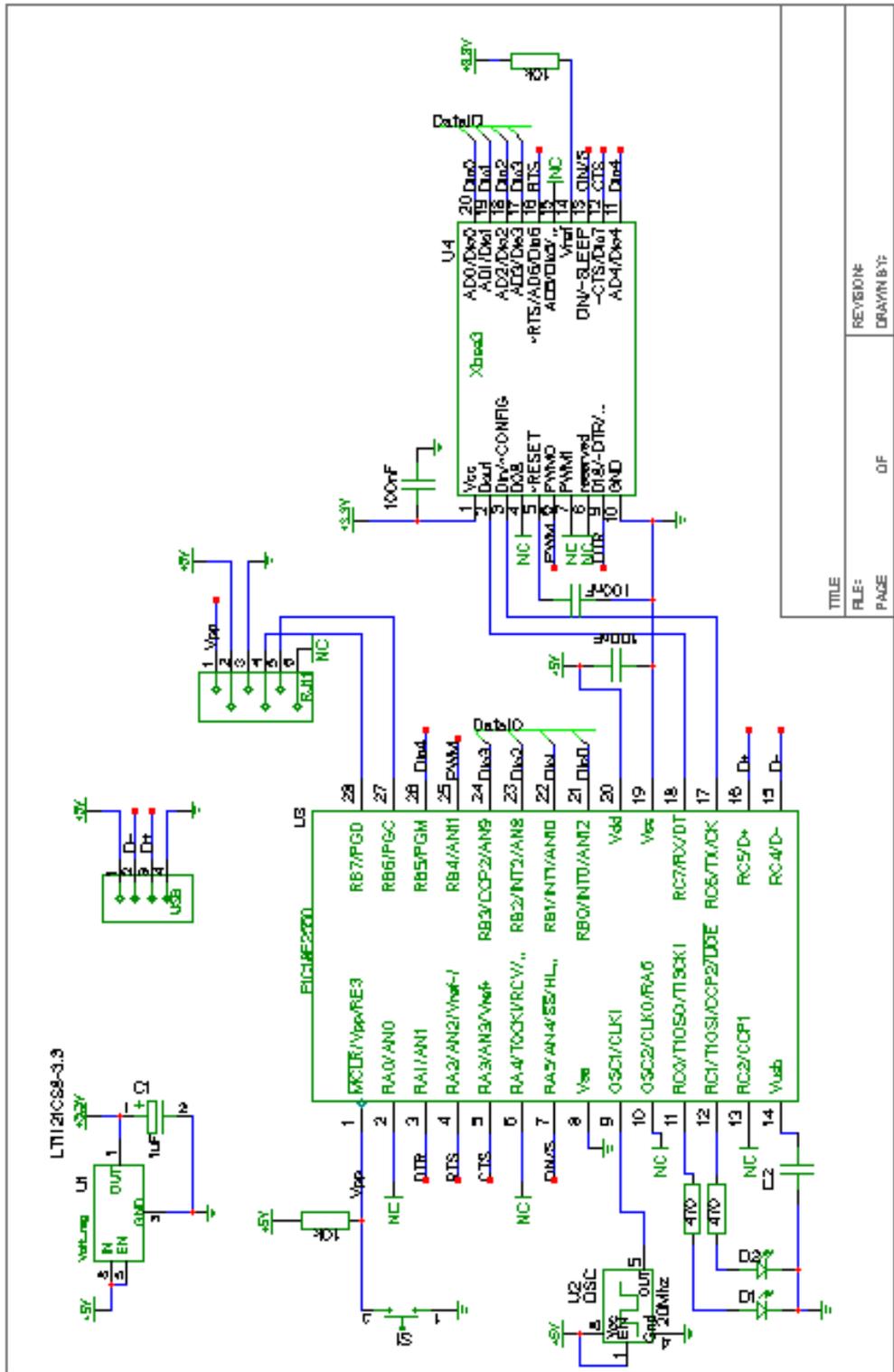


| | | |
|-------|----|-----------|
| TITLE | OF | REVISION: |
| FILE: | | DRAWN BY: |
| PAGE: | | |

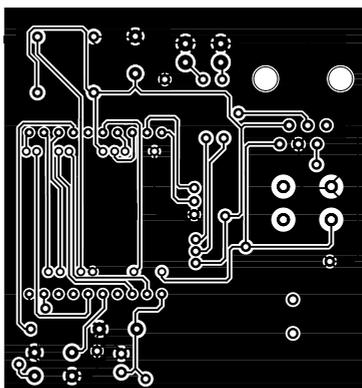


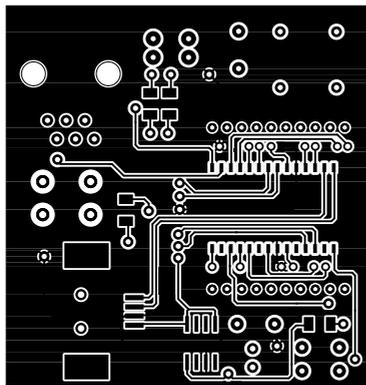


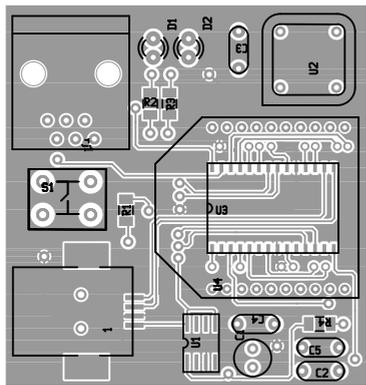




| | |
|-------|-------------|
| TITLE | REVISION: |
| FILE: | DRAWING BY: |
| PAGE | OF |







CERN
Organisation Européenne pour
la Recherche Nucléaire
F-01631 CERN Cedex
France
<http://www.cern.ch/>



Kévin Raymond
IUT Anecy
Département GEII
Année 2007



Résumé

Mesure d'une température avec précision et transmission par liaison sans fil

Dans cette étude, nous abordons la mesure d'une température ambiante avec précision. Après avoir décrit le fonctionnement globale d'une telle mesure et des différentes possibilités de mises en œuvre, nous étudierons le défaut de la chaîne de mesure et l'incertitude introduite.

Ceci passe par l'utilisation des logiciels libres de Conception électronique Assisté par Ordinateur et de programmation des microcontrôleurs, ainsi que par l'utilisation de divers protocoles comme le ZigBee ou l'USB.

Pour une plus grande portabilité, l'étude est réalisée dans un environnement GNU/Linux.

Mots-clés : Mesure de température, RTD, PRTD, Pt100, ZigBee, Xbee, USB, PIC, GNU, Linux.

Abstract

High precision measurement of temperature and wireless transmission of measurement results.

This report explores various possibilities for measuring high precision temperature, focusing on circuit conditioning. It discusses various tools

such as PIC programation, Electronic Design automation tool, as well as protocols such as USB and Zigbee solutions. The entire study is based on the GNU/Linux platform.

Keywords : Temperature sensor, RTD, PRTD, Pt100, ZigBee, Xbee, USB, PIC, GNU, Linux.

Rapport disponible : <http://kevin.raymond.free.fr/Stage/rapport/RapportKR.pdf>