
Rapport intermédiaire - ThermoPIC

EPHEC - Groupe 1

Melvin Campos Casares, Maxime de Cock, Guillaume
Vanden Herrewegen, Hubert Van De Walle

11 mars 2019

Rapport intermédiaire ThermoPIC - Groupe 1

Objectifs du projet	3
Introduction	3
Matériel nécessaire	3
Fonctionnement	4
Performance	4
Etat d'avancement	4
Général	4
Simulation	5
Programmation	5
Schéma	5
Schéma du circuit sur Eagle	5
Plaque de tirage PCB	6
Répartition du travail	6
Planning et échéances	7
Estimation du travail à faire	9
Objectifs personnels et attente	9
Melvin Campos Casares	9
Maxime de Cock	9
Guillaume Vanden Herrewegen	10
Hubert Van De Walle	10
Conclusion	10

Objectifs du projet

Introduction

ThermoPIC est un projet consistant en l'application d'une sonde de température envoyant au PIC une mesure analogique de la température ambiante d'une pièce.

Une intégration digitale et un signal d'alerte sont tous deux requis afin de prévenir l'utilisateur de l'éventuel dépassement d'un seuil critique défini par l'utilisateur. L'intégration digitale est l'affichage de la température sur un afficheur intégré au circuit, tout comme d'un indicateur sur le programme lancé sur l'ordinateur relié au PIC. Le signal d'alerte n'est autre que l'enclenchement d'une LED de couleur rouge et un message d'alerte sur l'ordinateur relié.

Ce projet est à construire de A à Z, dans le sens où il nous faut schématiser, simuler, mais également programmer l'intégralité du circuit électronique choisi.

Contraintes

Certaines contraintes nous ont été imposées. Tout d'abord, la communication entre le circuit et l'ordinateur doit se faire par le biais d'une application écrite en Java. De plus, certains composants du circuit sont prédéfinis comme le PIC, la sonde de température, les afficheurs 7 segments et des LED (une rouge et une verte). La sonde température doit fonctionner dans la gamme de 0°C à 100°C **minimum**, les afficheurs 7 segments doivent servir à l'affichage de la température et les LED aux alertes (LED rouge clignotante si alerte, LED verte continue si aucune alerte en cours).

La programmation du PIC est possible via l'interface RS232 ou via l'utilisation d'un module FTDI permettant la conversion du port série en port mini-USB.

Matériel nécessaire

1. PIC 18F458
2. Sonde température LM35
3. Amplificateur LM324
4. Crystal XTALS
5. Port série et convertisseur série MAX232
6. Deux afficheurs 7 segments à cathodes communes HD-1103
7. Alimentation 5V
8. Pin Header PINHD-1X3
9. Résistances, bouton poussoir 10-XX, LED rouge et verte LED3MM et/ou LED5MM

Fonctionnement

Le fonctionnement de ce circuit repose sur un programme tournant en boucle, qui va à chaque instant donné, récupérer la valeur de la sonde de température et comparer cette valeur à la température maximale entrée par l'utilisateur via une application Java. Ce programme va aussi vérifier si une donnée est reçue via son RX envoyée depuis l'application Java vers l'entrée Serial du PIC.

Lors du fonctionnement du circuit, si la température est inférieure à la valeur maximale désirée, une LED verte sera allumée pour signaler que tout va bien. Si cette température dépasse la valeur maximale, une LED rouge clignotera pour signaler l'utilisateur que la température maximale a été dépassée. L'interface de l'application Java signalera aussi cette alerte.

Performance

Nous avons pensé en groupe qu'il serait intéressant de réduire la consommation électrique et le besoin en composants de notre prototype afin que celui-ci respecte au mieux les désires de chacun. Pour respecter cet engagement, nous avons utilisé un affichage simultané sur deux afficheurs 7 segments et en n'utilisant qu'un seul décodeur à 7 segments. Grâce à deux transistors, nous parvenons à choisir sur quel afficheur afficher un chiffre. Si on transit rapidement entre les deux afficheurs, l'impact visuel sera inexistant. Cette manipulation a permis de réduire de 50% la consommation électrique de l'affichage et l'utilisation d'un seul décodeur 7 segments.

Une autre solution possible, mais plus difficile à mettre en place, serait d'allumer les segments de l'afficheur un par un. Seulement, cette option n'utiliserait pas un décodeur et donc nous devrions utiliser 7 sorties du PIC. Cette option n'était donc pas envisageable.

Etat d'avancement

Général

Ce projet étant non seulement intéressant, mais aussi amusant, nous voulions nous y mettre le plus vite possible pour ne pas être débordé par les autres projets en parallèle. La prise en main des programmes requis n'a pas été facile malgré l'intuitivité assez avancée de ces programmes. Nous avons, par souci de disponibilité des programmes, commencé par le schéma du circuit sur Eagle. Bien évidemment, nous avons par la suite fait la simulation sur Proteus lorsque nous avons accès à ce programme.

Simulation

La simulation est encore en cours de conception, en même temps que la programmation C. Le fichier Proteus ainsi que la programmation en C est d'ores et déjà disponible sur le GitHub même si pas encore finalisé. Nous les terminerons sous peu afin de pouvoir continuer la programmation Java.

Programmation

Pour la partie programmation, nous avons décidé de partir d'un circuit vide. Nous avons ajouté les différentes parties du circuit au fur et à mesure que le code avançait. La première partie était l'affichage de la température sur les afficheurs 7 segments. Une fois cette partie fonctionnelle, nous avons travaillé sur l'entré analogique de la sonde de température.

Nous avons eu quelques difficultés concernant cette partie du code car nous n'arrivions pas à visualiser quelle type de valeur le convertisseur analogique numérique sortait. Une fois cette valeur « maitrisée », nous l'avons convertie pour qu'elle puisse représenter la température relevée. Nous avons ensuite pu afficher aisément la température sur les afficheurs 7 segments.

Schéma

Schéma du circuit sur Eagle

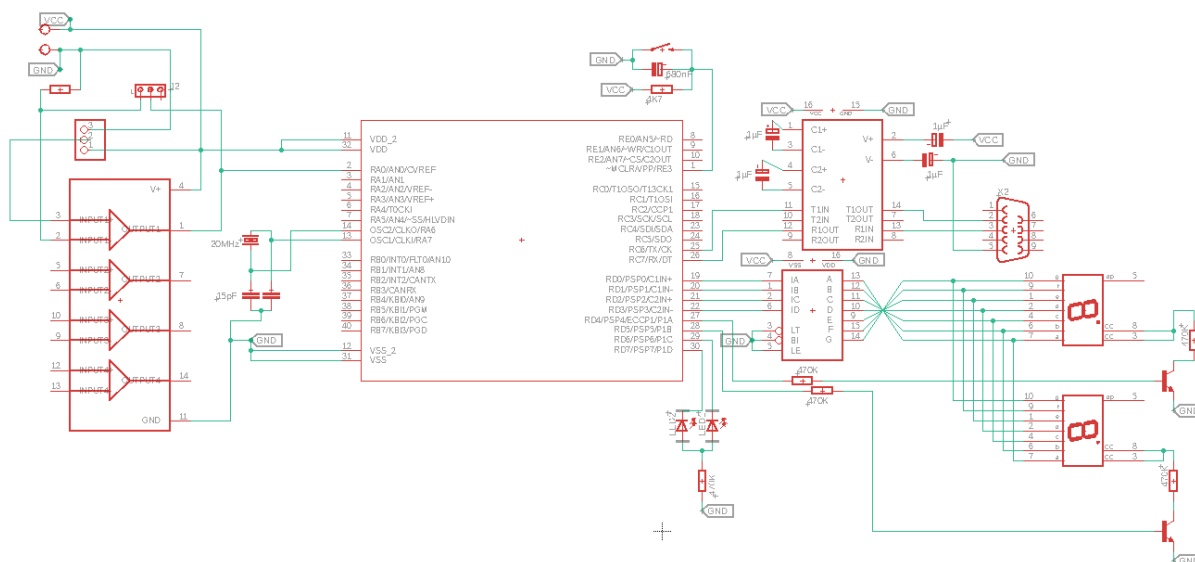


Figure 1: Schéma du circuit réalisé sur Eagle

Plaque de tirage PCB

Schéma de la plaque de tirage PCB réalisé sur Eagle sans l'isolation par souci de clarté dans le rapport :

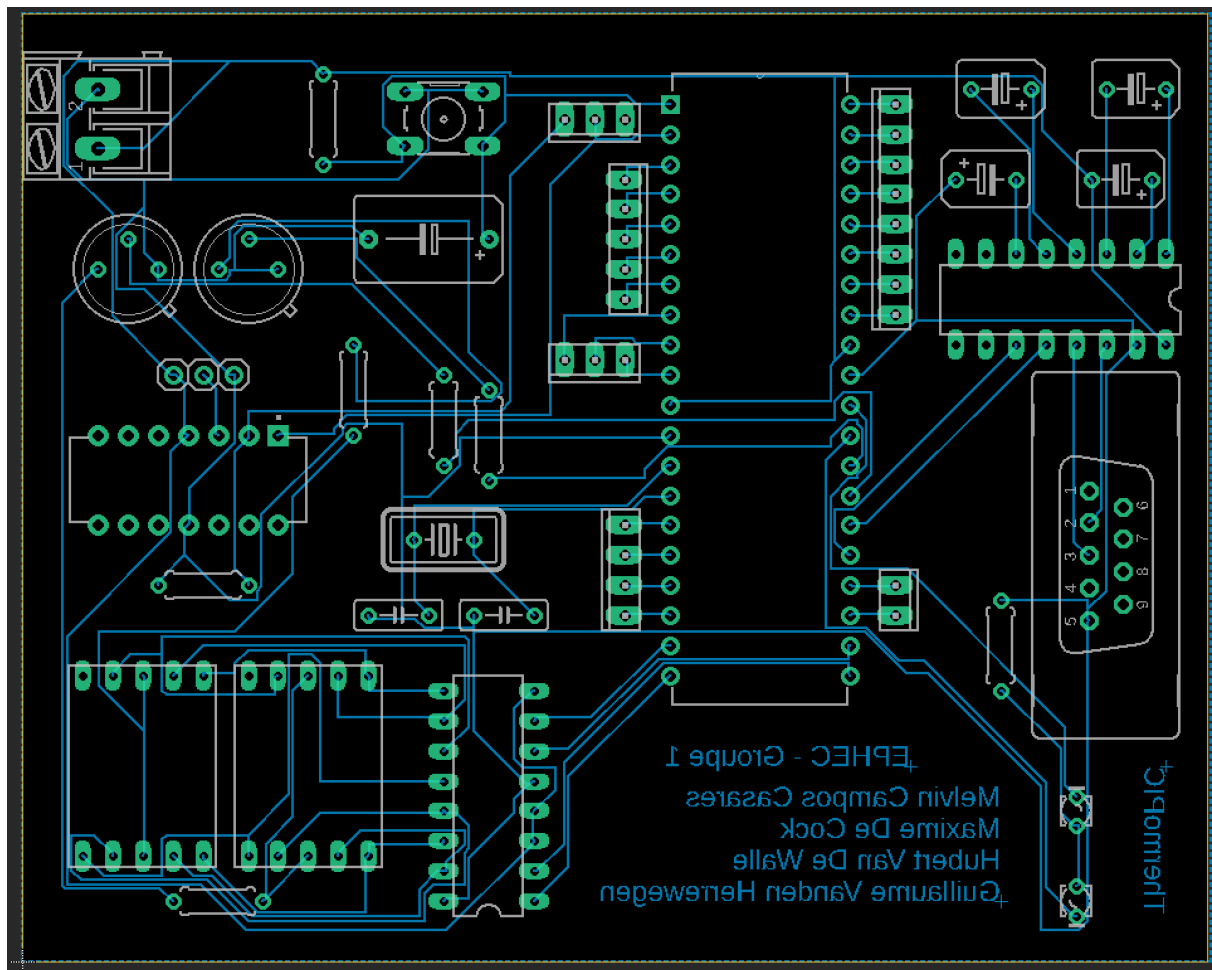


Figure 2: Plaque de tirage PCB réalisé sur Eagle

Répartition du travail

Nous nous sommes réparti le travail par le biais de la plateforme Trello. Il s'agit d'un outil collaboratif de gestion de projets qui nous a été conseillé lors de la prise de connaissance du cahier des charges de ce projet.

Nous avons déjà eu l'occasion d'utiliser cet outil lors de précédents projets.

Planning et échéances

Création d'un groupe de discussion sur Signal (Melvin Campos Casares & Hubert Van de Walle)

Créé le 4 février à 17h34

Afin de faciliter la communication entre les membres du groupe, nous avons décidé de créer un groupe de discussion sur l'application Signal, développée par Open Whisper Systems. Il nous permet également de passer des appels groupés tant en vocal qu'en visioconférence.

L'avantage de Signal est son cryptage accru de bout en bout, sa gratuité (sans publicité et pas de collecte d'informations, le développement est maintenu sur base de dons et subventions de la communauté) et le fait qu'il soit Open Source.

Création du repository GitHub (Melvin Campos Casares)

Créé le 4 février à 18h05

Par souci de facilité avec la synchronisation des fichiers, nous utilisons un repository GitHub où nous sommes tous contributeurs.

Le repository est public et accessible à l'adresse suivante : <https://github.com/melvinmajor/thermopic>

Création du tableau de planification Trello (Guillaume Vanden Herrewegen)

Créé le 5 février à 19h51

Afin de mieux coordonner notre planification et de suivre au plus près nos deadlines, nous utilisons la plateforme Trello qui nous a été conseillée lors de la présentation du projet et du cahier des charges.

Le Trello "*ThermoPic project*" étant privé, il n'y a pas de lien disponible afin d'y accéder à moins d'être invité à le rejoindre.

API de gestion de communication sur le port série (Hubert Van de Walle)

Finalisé le 6 février

Lors de la présentation du projet et du cahier des charges, il nous a été proposé d'utiliser l'API **RxTx** pour gérer la communication sur le port série. En analysant un peu cette API, nous avons pu remarquer qu'elle n'a plus été mise à jour depuis 6 ans et avons donc préféré trouver une autre API, plus récente, et pouvant convenir au projet.

Hubert a trouvé l'API `jSerialComm`, disponible sur GitHub, et permettant d'utiliser des événements.

Schéma technique sur Proteus et Eagle (Maxime de Cock & Guillaume Vanden Herrewegen)***Echéance le 8 février à 19h - Finalisé le 21 février***

Le délai ne fut pas respecté, entre autres car nous ne savions pas encore clairement quels composants utiliser plus précisément dans notre circuit, mais également par la difficulté de trouver une version propre de Proteus sans virus et disposant des librairies de base. Le schéma technique sur *Eagle* fut achevé le 16 février en courant de journée par Maxime. Pour ce qui concerne le schéma sur *Proteus*, nous attendions la fin du schéma sur Eagle pour pouvoir l'achever. Quelques dernières retouches ont été réalisées le 21 février sur Eagle tout comme la finalisation de Proteus, par Maxime.

Quelques modifications mineures ont été apportées par la suite afin d'optimiser au mieux notre plaque de tirage et améliorer quelque peu notre schéma technique.

Sélection des composants nécessaires au projet (groupe entier)***Finalisé le 17 février***

Bien qu'un point très important, nous avons fini de choisir les composants nécessaires au projet lorsque nous avons terminé la partie principale du schéma technique sur Eagle. Lors des cours, nous avons pu remarquer l'intérêt de certains composants et reprendre ces derniers dans notre schéma afin de finaliser correctement la partie technique de l'électronique du projet.

Fichier EAGLE (.brd) pour tirage PCB (Melvin Campos Casares)***Echéance le 11 mars à 17h - Finalisé le 28 février***

Avec le schéma achevé le 16 février, il n'y avait plus qu'à réaliser le fichier de carte de circuit imprimé (.brd). Pour ce faire, nous sommes partis du schéma Eagle et nous nous sommes rendu compte de certaines modifications à apporter au niveau du schéma afin que notre plaque d'impressions soit correcte. De plus, lors du commencement de la programmation du code C, nous avons pu remarquer également certaines modifications à apporter au niveau des résistances choisies, n'entraînant heureusement pas de nouveaux changements au niveau du fichier servant au tirage PCB.

Quelques modifications mineures ont été apportées par la suite afin d'optimiser au mieux notre plaque de tirage et améliorer quelque peu notre schéma technique.

Programmation du code C (groupe entier)***Commencé le 6 mars***

La programmation du code C de ce projet est principalement géré par Guillaume Vanden Herrewegen. Cela n'empêche que nous nous échangions des informations au fur et à mesure de l'évolution afin de pouvoir avancer lorsque nous rencontrons un problème.

Maxime De Cock à soulevé des points pertinents concernant la partie électronique à devoir tenir compte pour la programmation en C alors que Hubert Van De Walle et Melvin Campos Casares ont plus aidé concernant les choix et directives à prendre afin de contrecarrer certains problèmes rencontrés. A l'heure actuelle, Guillaume Vanden Herrewegen est la personne principale ayant écrit le code C et ayant réalisé les tests préalables permettant de s'assurer du bon fonctionnement.

Rapport intermédiaire (groupe entier)

Echéance le 11 mars à 17h

Le rapport intermédiaire fut commencé le 8 février et complété au fur et à mesure de l'avancement du projet. Il a été tenu à jour par tout le groupe et principalement écrit par Melvin sur base des retours des autres membres du groupe.

Estimation du travail à faire

Objectifs personnels et attente

Melvin Campos Casares

Pour ce projet, j'espérais grandement pouvoir enfin comprendre l'utilisation de certains logiciels d'électronique permettant la création d'un schéma ainsi que sa manipulation via simulation. De plus, avoir un bon travail d'équipe et s'assurer de tout terminer à temps et à heure pour l'échéance finale est l'un de mes objectifs prioritaires.

N'ayant pas de facilité à la programmation en C, pouvoir discuter de certains points me paraissant plus obscurs avec les autres membres du groupe afin de mieux comprendre et mieux visualiser ce langage est un de mes objectifs *bonus*.

Mêler de la programmation avec de l'électronique et obtenir quelque chose de concret et fonctionnel au quotidien est mon attente principale pour ce projet.

Maxime de Cock

Dès l'annonce du projet, j'ai été très enthousiaste à l'idée de faire ce montage, j'ai donc demandé à mes coéquipiers de me laisser la partie schéma électronique. J'aime vraiment ça, c'est pourquoi j'attends beaucoup de ce projet, entre l'apprentissage des différents langages/programmes et la compréhension des datasheet, ce sont mes premiers vrais objectifs.

Guillaume Vanden Herrewegen

Ayant déjà réalisé des projets en électronique de type Arduino, j'étais vraiment intéressé par ce projet. De plus, nous allons apprendre à manipuler des programmes qui permettent de simuler un circuit électronique.

Grâce à ce projet, j'espère pouvoir m'améliorer dans la gestion de groupe, mais aussi dans l'amélioration de mes compétences en électronique ainsi que pouvoir, suite à l'analyse d'une demande, concevoir et assembler un prototype d'un projet en électronique.

Hubert Van De Walle

L'existence d'un projet tel que celui-ci en deuxième année des études TI me semble indispensable. En effet, il m'a permis d'avoir une meilleure idée de l'interaction entre l'électronique et la programmation. J'ai également découvert les différentes étapes rentrant dans la réalisation d'un circuit imprimé. De plus, ne pas se limiter à la théorie et réaliser un projet concret m'a permis, personnellement, de renforcer mon intérêt envers l'électronique.

Conclusion

Ce projet nous permet de pouvoir utiliser les connaissances apprises en cours de manière concrète et différente que dans des cas isolés levés dans le cadre des laboratoires mis en place lors des travaux pratiques.

Nous avons hâte d'avancer dans ce projet et rendre ThermoPIC totalement fonctionnel.

En conclusion, la réalisation du projet s'est globalement bien déroulée. Nous avons atteint la plupart des objectifs fixés au départ lors de cette première partie et certains de ceux-ci ont été adaptés afin que, malgré le retard sur nos propres deadlines fixé, nous ayons corrigé certains problèmes rencontrés et amélioré certains points. Nous avons d'ores et déjà quelques pistes d'améliorations que nous tenterons de réaliser lors de la suite de ce projet.