



COLOMBAN Romain
IUT1 GEII1 AS1



Rapport de Stage
Spécification, Conception et Réalisation
d'une carte de gestion des sécurités pour
le robot bipède BIP

Maître de stage : Pascal Di GIACOMO

Tuteur IUT : Mr Bée

Stage du 22 avril au 2 juillet 2004

REMERCIEMENTS

Je remercie l'INRIA Rhone Alpes de m'avoir accueilli pour mon stage de fin d'année, m'offrant ainsi une expérience très enrichissante dans un cadre de travail peu commun, celui de la recherche.

J'adresse aussi mes remerciements à toute l'équipe du SED, tout particulièrement à Pascal Di GIACOMO et Roger PISSARD-GIBOLLET pour leur aide et leur soutien tout au long du stage.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Unité de Recherche INRIA Rhône-Alpes :

ZIRST - 655 avenue de l'Europe - Montbonnot - 38334 Saint Ismier Cedex - France

Téléphone : +33 (0)4 76 61 52 00 - Fax : +33 (0)4 76 61 52 52

www.inrialpes.fr

Sommaire

1-	Introduction	3
2-	Description de l'entreprise	4
2-1.	Présentation de l'INRIA	4
2-2.	Présentation de l'INRIA Rhône-Alpes	5
2-3.	Présentation du service SED	6
3-	Présentation du stage	7
3-1.	Le robot BIP 2000	7
3-2.	Problématique	8
3-3.	Sujet du stage	8
4-	Travail effectué	9
4-1.	Etude préliminaire	9
4-1.1.	L'architecture du robot	9
4-1.2.	Les sources d'arrêt d'urgence	11
4-1.3.	Arrêt du robot	12
4-2.	Spécifications de la carte	13
4-2.1.	Communications	13
4-2.2.	Synoptique	13
4-3.	Conception de la carte	14
4-3.1.	Détection des défauts	14
4-3.2.	Commande d'arrêt d'urgence du robot	15
4-3.3.	Communication avec la partie commande	16
4-3.4.	Programmation du microcontrôleur	17
4-4.	Réalisation de la carte	18
4-4.1.	Connectique	18
4-5.	Tests et intégration	19
5-	Conclusion	20
6-	Bibliographie	21
7-	Annexes	22

1- **Introduction**

Dans le cadre de ma formation en DUT Génie Electrique et Informatique Industrielle à l'IUT1 de Grenoble, j'ai effectué un stage – de dix semaines – concrétisant ma formation. Il s'est déroulé au sein de l'INRIA, un institut national de recherche en informatique et automatique (Cf. §2.1).

Le service SED (Support aux Expérimentations et Développement logiciel), en étroite collaboration avec les chercheurs, développe les plateformes expérimentales nécessaires aux travaux de recherche, assure le bon fonctionnement de celles-ci et participe aux expérimentations. Les plateformes concernent les domaines de la robotique, de la réalité virtuelle et les clusters de PC.

Mon travail au sein de ce service a été d'améliorer la sécurité d'un robot, en concevant et réalisant un module intégré dans le robot bipède BIP. Ce module, indépendant du système de commande, doit assurer avec le plus de fiabilité possible l'arrêt du robot lorsqu'un défaut de fonctionnement est constaté ou lorsqu'une demande d'arrêt (par logiciel ou interrupteur) est formulée. Il communique avec le système central de commande.

2- Description de l'entreprise

2-1. Présentation de l'INRIA

Créé en 1967 à Rocquencourt près de Paris, l'INRIA [1], Institut National de Recherche en Informatique et en Automatique, est un établissement public à caractère scientifique et technologique qui mène des recherches avancées dans le domaine des Sciences et Technologies de l'Information et de la Communication (STIC). Ce domaine inclut l'informatique et l'automatique, mais aussi les télécommunications et le multimédia, la robotique, le traitement du signal et le calcul scientifique.

L'INRIA est placé sous la double tutelle du Ministère de la Recherche et du Ministère de l'Economie, des Finances et de l'Industrie.

L'INRIA a l'ambition d'être au plan mondial un institut de recherche au cœur de la société de l'information.

Sa volonté est de mettre en réseau des compétences et des talents de l'ensemble du dispositif de recherche français dans le domaine des STIC. Ce réseau permet de mettre l'excellence scientifique au service des progrès technologiques, créateurs d'emplois, de richesse et de nouveaux usages répondant à des besoins sociaux économiques.

Son organisation décentralisée (5 unités de recherche), ses petites équipes autonomes et évaluées régulièrement permettent à l'INRIA d'amplifier ses partenariats, 47 projets de recherche sur 87 sont communs avec les universités, les grandes écoles et les organismes de recherche. Il renforce son implication dans les travaux de valorisation des résultats de recherche et le transfert technologique : 600 contrats R&D avec l'industrie et un peu moins d'une cinquantaine de sociétés sont issues de l'INRIA.



Implantation de l'INRIA en France

Ressources budgétaires

-budget total : 125 M Euros.

Ressources humaines

- titulaires INRIA : 900 (400 chercheurs, 500 ingénieurs et techniciens) ;
- post-doctorants, stagiaires, invités : 750 ;
- doctorants : 800 ;
- chercheurs et enseignants d'autres organismes : 450 ;
- "ingénieurs experts" (sur contrat de recherche) : 200.

2-2. Présentation de l'INRIA Rhône-Alpes

a) Introduction

Créée en décembre 1992, l'INRIA Rhône-Alpes [2] est la plus récente des cinq unités de recherche de l'INRIA.

Menées au sein d'une région en plein essor technologique, les activités de l'unité de recherche INRIA Rhône-Alpes mobilisent plus de 400 personnes, dont 220 chercheurs, géographiquement réparties sur deux sites : la ZIRST [3] de Montbonnot et le site technopolitain de Lyon [4].

Ces activités s'inscrivent dans le cadre des missions que doit accomplir l'INRIA en tant qu'établissement de recherche national, tout en se focalisant sur les objectifs stratégiques poursuivis par l'institut dans le domaine des sciences et technologies de l'information et de la communication.

L'INRIA Rhône-Alpes accueille plus de 130 doctorants, ingénieurs et stagiaires. Ses chercheurs participent à l'enseignement supérieur au sein des universités et grandes écoles de la région Rhône-Alpes (Institut National Polytechnique de Grenoble, université Joseph Fourier, université Pierre Mendès-France, université de Savoie, Ecole Nationale Supérieure de Lyon).

b) Pôles de recherche

L'INRIA Rhône-Alpes mène ses activités en étroite collaboration avec les laboratoires de recherche publics et privés, nationaux et internationaux, et elle entretient des liens privilégiés avec la fédération d'Informatique et Mathématiques Appliquées de Grenoble (IMAG) [5]. Ces activités sont organisées autour de quatre pôles de recherche :

- Maîtriser les systèmes et réseaux informatiques : Réseaux, parallélisme et systèmes répartis.
- Aider à la conception et à la création : Base de connaissances, documents multimédia, modèles cognitifs.
- Percevoir, simuler et agir : Synthèse d'images, réalité virtuelle, vision par ordinateur et robotique.
- Modéliser les phénomènes complexes : Automatique, simulation et calcul scientifique.

2-3. Présentation du service SED

Mon stage se déroule au sein du service SED [6] : Support Expérimentations et Développement logiciel. Je suis physiquement installé dans la halle robotique.

Composé d'une douzaine de personnes, son rôle principal est le développement et le fonctionnement des plateformes expérimentales dans tous les domaines : mécanique, électronique, logiciel.

Impliqué dans plus de 10 projets sur les thèmes de la robotique et de la vision, il a aussi une activité de recherche par la mise en place et la participation à des expérimentations.

Quelques plateformes présentes dans la halle robotique



Trois véhicules électriques CYCAB



Deux robots six roues KOALA



Un portique dédié à la vision

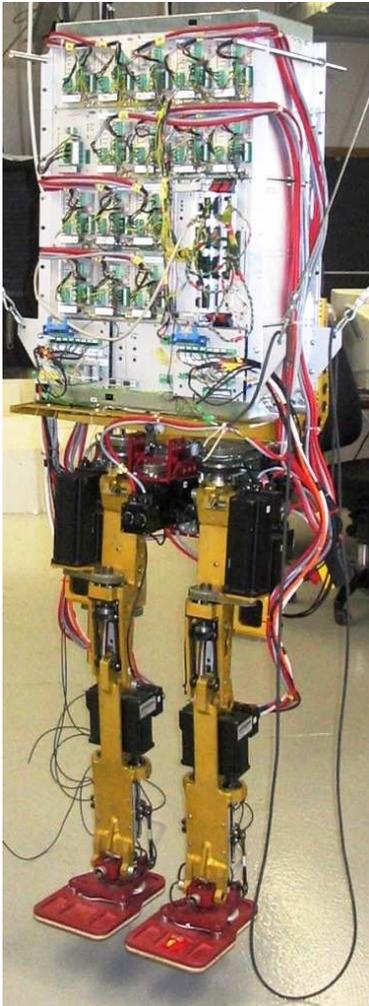


Un bras manipulateur 6 axes RX90

Un robot bipède, BIP, sur lequel j'interviens durant le stage (cf. Page suivante).

3- Présentation du stage

3-1. Le robot BIP 2000



Robot BIP 2000

Le projet BIP a débuté il y a 10 ans, il a pour objectif la conception de robots marcheurs de type bipède, et plus spécifiquement la mise au point de leur commande. L'intérêt de tels robots réside dans leur capacité « naturelle » à évoluer dans les environnements de notre vie quotidienne, essentiellement conçus pour la bipédie.

Le but de ce prototype est de disposer d'un système expérimental pour mettre en œuvre de nouvelles commandes de marche élaborées par le projet BIPOP [7].

Un axe important de recherche concerne les applications biomédicales, en particulier la réhabilitation des paraplégiques par électrostimulation (projet DEMAR [8]).

Son développement s'est fait par plusieurs prototypes de plus en plus avancés, pour en arriver au BIP2000, un robot ayant 15 degrés de libertés (DDL). Mesurant 180cm pour 100kg, sa morphologie et ses mouvements sont voulus proches du modèle humain.

Le système de commande [13] se situe dans le buste, il est composé de 15 variateurs commandant les moteurs et d'un ordinateur au format VME intégrant entre autres un microprocesseur Motorola sous linux et des modules d'Entrées/Sorties analogiques et numériques.

Chaque articulation est commandée par un ou plusieurs moteurs à courant continu sans balais (Brushless) avec resolver (et codeurs incrémentaux), associés à des systèmes vis-écrou-biellettes.

Des potentiomètres sur chaque axe fournissent une information grossière sur la position absolue de l'axe.



Détails des pieds

3-2. Problématique

D'une part le robot bipède est un prototype donc son fonctionnement n'est pas encore totalement éprouvé ; d'autre part les tâches (mouvements) qui lui sont demandées constituent des expérimentations dont on ne connaît pas à priori les résultats. Pour ces deux raisons, la probabilité d'un dysfonctionnement n'est pas faible.

Par exemple, le robot peut atteindre une butée mécanique sur une articulation, et entraîner des dommages. Il peut aussi arriver que la commande ne réponde plus correctement (problème matériel ou logiciel), le robot devenant alors incontrôlable : vitesse importante, trajectoires erronées ...

Il est alors nécessaire de détecter les différents défauts qui peuvent survenir, afin d'assurer la sécurité des personnes et du robot lui-même. Il est aussi utile de savoir quel défaut est survenu, pour en chercher la cause et tenter d'y remédier.

Jusqu'à présent le robot ne disposait que de quelques sécurités opérationnelles, comme des capteurs de fin de course sur chaque articulation (butées électriques précédant les butées mécaniques) et un interrupteur de type coup de poing pour l'arrêt d'urgence manuel, mais sans aucune remontée d'information vers le contrôleur logiciel du robot.

3-3. Sujet du stage

Mon travail au sein de l'INRIA consiste à réaliser de cette carte de sécurité, selon les étapes suivantes :

- Etude / Compréhension de l'architecture matérielle et logicielle du robot ;
- Spécification des fonctions de la carte ;
- Conception de la carte de sécurité et de son programme ;
- Réalisation d'un prototype wrappé ;
- Tests du prototype.

Cette carte est architecturée autour d'un microcontrôleur PIC, dont les principales fonctions sont les suivantes :

- Arrêt du robot lors d'un mauvais fonctionnement.
- Communication avec la partie commande du robot et l'utilisateur.

4- Travail effectué

4-1. Etude préliminaire

Une première étude du robot permet de mettre en évidence les problèmes que rencontre le robot, ainsi que les différents moyens de détection afin de provoquer un arrêt d'urgence de son fonctionnement.

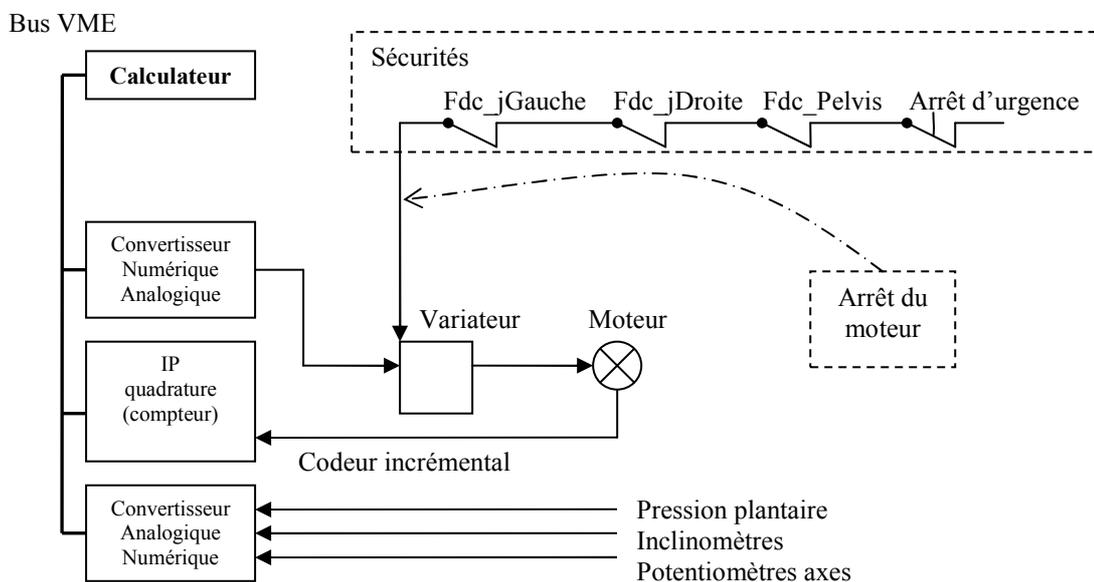
La logique de la sécurité d'arrêt d'urgence ne doit pas être traitée par le calculateur car elle doit être assurée même si le logiciel n'est plus fonctionnel : c'est pour cette raison qu'elle est sous-traitée à un module spécialisé (logique câblée, programmée ou microcontrôleur).

4-1.1. L'architecture du robot

a) Sans dispositif de gestion des sécurités (état actuel)

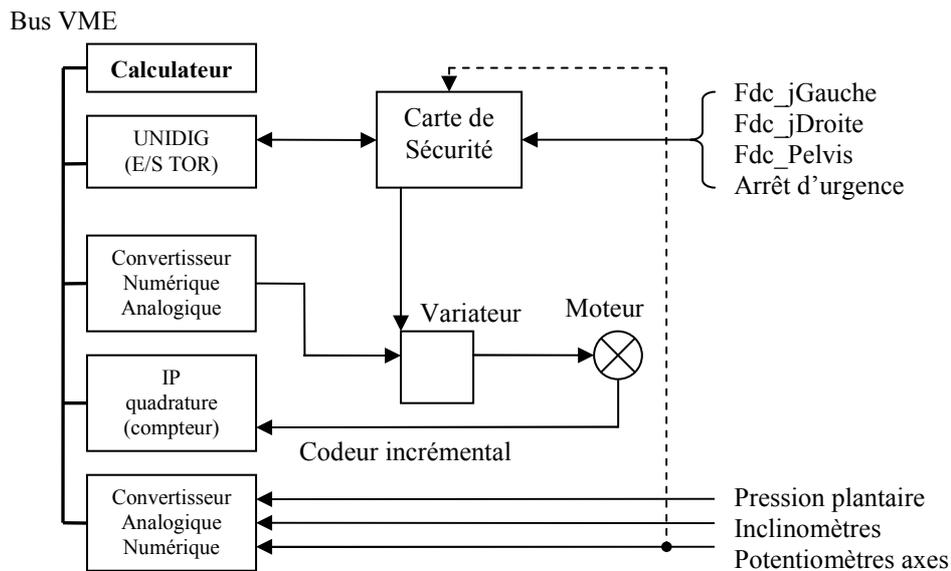
Le synoptique ci-dessous rend compte du fonctionnement global du robot avant mon stage, et en particulier des sécurités qui sont installées : quelques capteurs (fins de course) et un interrupteur d'arrêt d'urgence sont connectés entre eux et commandent directement une entrée de validation des variateurs.

Le calculateur sur lequel fonctionne le logiciel de contrôle n'est pas informé des problèmes de sécurité, il détecte indirectement un problème en surveillant le seuil de la norme de l'erreur de la commande.



b) Avec le module de gestion des défauts

Mon travail consiste à créer un module remplaçant et améliorant la gestion des sécurités existante, de façon indépendante de l'architecture de commande :



Quelques remarques sont à faire à propos de l'architecture du robot :

- Les cartes connectées au bus VME, les quinze variateurs et la carte de sécurité sont installés dans la baie de commande, formant le torse du robot ;
- Les quinze degrés de liberté (donc moteurs) sont asservis en couple via une consigne du CNA qui assure cette fonction ;
- Les codeurs incrémentaux ne sont pas physiquement présents sur le robot, ils sont émulés par les variateurs à partir des informations fournies par le resolver.

4-1.2. Les sources d'arrêt d'urgence

L'utilisation d'un microcontrôleur rapide et possédant beaucoup d'entrées-sorties (PIC 16F877, 33 E/S, entrées analogiques) permet de détecter un plus grand nombre de sources d'arrêts d'urgence par rapport à celles qui existaient jusqu'à présent.

Pour des raisons de sécurité, toutes les sources d'arrêts d'urgence sont ou ont été rendues actives à l'état haut.

a) Capteurs sur le robot

- Fins de course



Fin de course

Les capteurs de fins de course sont placés aux extrémités de la course de chaque articulation, soit trente microrupteurs pour les quinze degrés de libertés existants sur le robot.

Ils permettent donc d'éviter qu'une articulation atteigne sa butée mécanique.

Ces fins de course sont regroupés en fonction de la morphologie du robot, soit en trois groupes : Jambe gauche, Jambe droite et Pelvis.

De types normalement fermés, ils sont connectés en série et permettent ainsi une sécurité passive qui provoque un niveau haut (via une résistance de pull-up) lors de la rupture d'un fil, comme pour l'atteinte d'une butée.

- Potentiomètres des axes



Potentiomètres

Chacun des axes du robot entraîne un potentiomètre sans butée afin d'avoir une information grossière sur l'angle de l'axe.

En particulier, sur trois articulations (chevilles et tronc) commandées par des actionneurs parallèles (Cf. p7 : photo du pied), les butées mécaniques peuvent être atteintes sans qu'aucun microrupteur ne le détecte.

Il faut alors utiliser les potentiomètres placés sur chaque axe des articulations pour connaître son angle et le comparer à un angle limite qui provoquera le cas échéant un arrêt d'urgence.

Ces six potentiomètres peuvent être démontés pour des activités de maintenance, il faut alors pouvoir réinitialiser les angles limites facilement. C'est pour cela qu'ils seront connectés au Convertisseur Analogique Numérique du PIC pour permettre un calibrage rapide et sûr.

La carte que j'ai développée intègre cette évolution ultérieure. Il suffira de connecter les entrées concernées et de modifier le logiciel.

b) Coup de poing d'arrêt d'urgence



De type normalement fermé, cet interrupteur sera traité au niveau du microcontrôleur de la même manière qu'un fin de course, de plus il agira aussi directement sur l'inhibition des couples moteurs afin d'assurer une sécurité maximale.

Interrupteur Coup de poing

c) Défaut du variateur

Chaque variateur [9] possède une sortie nommée « Drive Enabled », elle informe de tout problème concernant le fonctionnement du variateur (surchauffe, surintensité, etc.).

Ces sorties, à collecteurs ouverts, sont câblées de manière à avoir une seule entrée sur le PIC concernant l'état des 15 variateurs.

d) Contrôleur logiciel

Le logiciel de commande [14] doit pouvoir provoquer l'arrêt du robot, que ce soit volontairement (via la programmation de l'utilisateur) ou lors d'un plantage de celui-ci, et ainsi stopper le fonctionnement avant l'apparition d'un autre défaut.

- **Arrêt logiciel**

Cette source d'arrêt d'urgence est sous le contrôle logiciel de l'expérimentateur et provient d'une sortie de la commande du bipède. En entrée du PIC elle sera traitée de la même manière qu'un fin de course.

- **Chien de garde**

Il provient d'une sortie de la partie commande, et réarme un monostable qui indique que les processus de la commande s'exécutent correctement. Cette entrée doit pouvoir être désactivée en fonction de l'utilisation faite du robot.

L'utilisation du chien de garde n'étant pas critique, il ne sera pas implémenté pour l'instant mais à l'instar des potentiomètres, elle est prévue.

4-1.3. Arrêt du robot

L'arrêt du robot se fait en stoppant le fonctionnement de tous les moteurs, directement par l'action sur une entrée prévue à cet effet de chaque variateur.

L'arrêt des moteurs se fait par l'ouverture du contact entre l'entrée « Torque Enable » et la masse de tous les variateurs. Cette entrée met à zéro le couple moteur ce qui a pour conséquence la chute du robot, c'est pour cela qu'il est en permanence relié par des câbles avec du mou à un dispositif de type pont roulant afin de limiter la chute de l'armoire et de la structure du robot.

4-2. Spécifications de la carte

En plus de gérer les entrées-sorties décrites précédemment, la carte de sécurité doit communiquer avec la partie commande du robot et l'expérimentateur.

4-2.1. Communications

Après l'arrêt du robot, quatre sorties logiques du microcontrôleur sont positionnées en fonction de la source ayant causé l'arrêt (Fin de course, Arrêt d'Urgence,...) et envoyées à la baie de commande via un module d'entrées/sorties logiques du robot, le module UNIDIG [10].

Cette information sur le défaut constaté sera aussi visualisée en face avant, pour informer l'utilisateur, à l'aide de six leds :

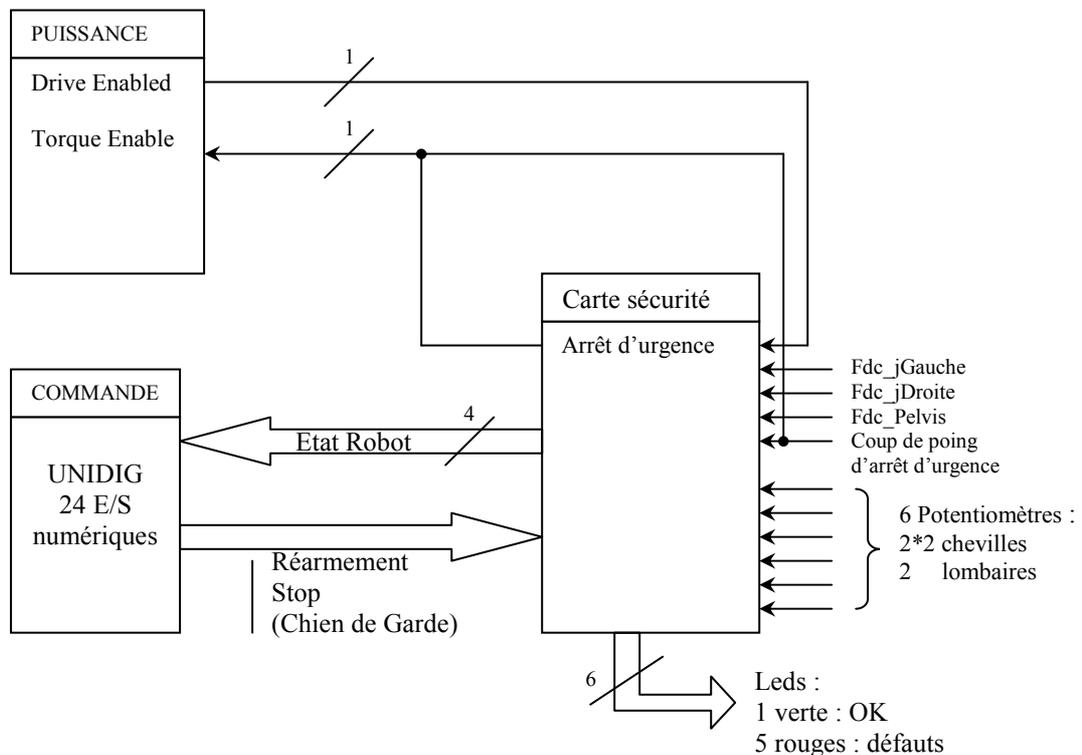
- 1 verte indiquant le bon fonctionnement de la carte de sécurité ;
- 5 rouges pour les différents défauts et l'arrêt d'urgence.

La remise en route des moteurs ne sera possible que par l'activation d'une entrée de réarmement (provenant de la commande) et l'absence de tout défaut ou l'inhibition, si nécessaire, du défaut.

4-2.2. Synoptique

Cette première étude permet la réalisation du synoptique ci-dessous, rendant compte ainsi de l'architecture externe de la carte.

Il est à noter que la carte, dans sa première version, ne tient pas compte des potentiomètres ni du chien de garde.



4-3. Conception de la carte

La conception matérielle de la carte s'est faite principalement en fonction des fonctionnalités du microcontrôleur PIC, pour avoir une programmation simple et efficace.

Cette section ne présente que les parties essentielles de la carte, les schémas complets sont disponibles en annexe 4.

4-3.1. Détection des défauts

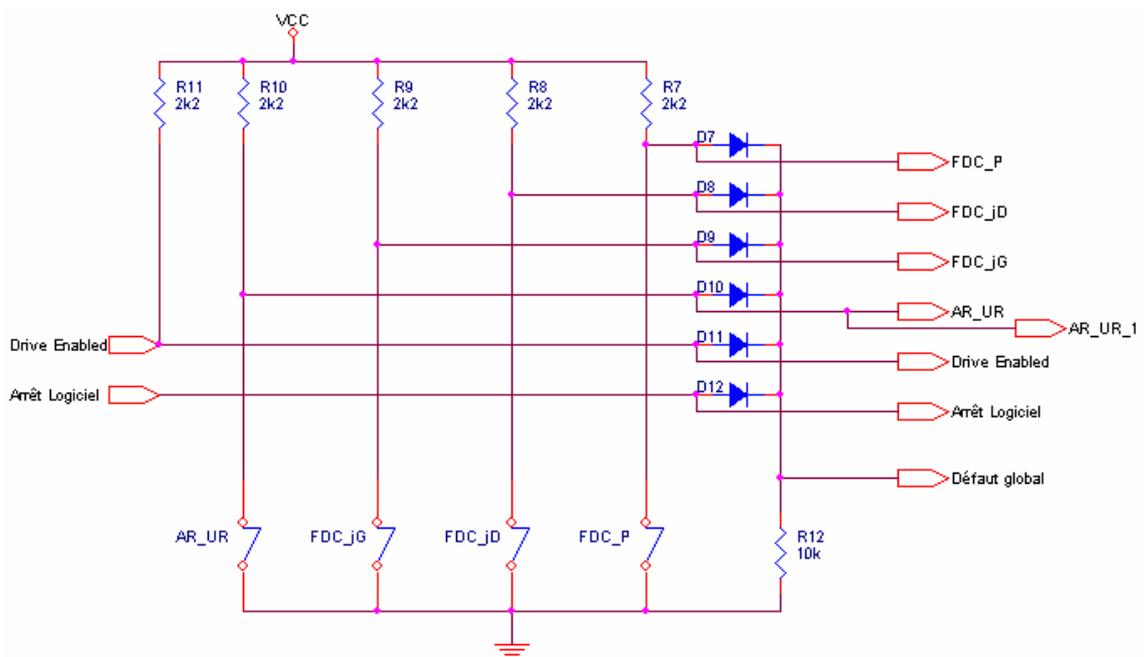
Cette fonction est la fonction principale de la carte, elle doit être traitée prioritairement à tous les instants.

Le choix s'est donc porté sur l'utilisation d'une interruption externe parmi les deux existantes :

- INT : interruption sur front (montant OU descendant) de l'entrée 0 du PORTB ;
- RBI : interruption sur changement des entrées 4 à 7 du PORTB (donc front montant ET descendant).

Etant donné que six entrées différentes sont implémentées, on ne peut pas utiliser la deuxième interruption, on utilisera donc INT en détection de front montant.

D'un point de vue matériel, il faut générer un signal de défaut global qui soit connecté à l'entrée PORTB-0. Puisque toutes les entrées sont actives à l'état haut, il faut simplement utiliser une fonction logique OU qui a été réalisée grâce aux diodes D7 à D12 et à la résistance R12, selon le schéma ci-dessous :



Détection des défauts

Pour savoir quelle entrée a causé l'arrêt, on connecte les différentes entrées de défauts sur le PORTB et dans le programme on teste quelle entrée est au niveau haut.

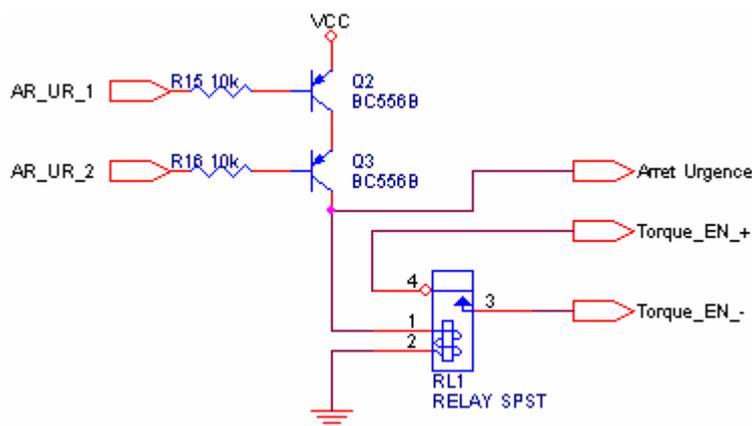
Grâce à cette architecture, on peut facilement ajouter de nouvelles sources d'arrêt d'urgence en ajoutant des diodes et en les connectant sur des entrées du PIC.

4-3.2. Commande d'arrêt d'urgence du robot

L'arrêt des moteurs se fait par l'ouverture du contact entre les entrées « torque enable » et la masse de tous les variateurs (Cf. annexe 6), l'utilisation d'un relais est par conséquent la manière la plus simple de procéder.

Comme vu lors de l'étude préliminaire, l'arrêt d'urgence pourra être déclenché par deux sources : une sortie du microcontrôleur et directement par l'interrupteur du coup de poing d'arrêt d'urgence, toutes deux actives à l'état haut.

Le simple montage à transistors ci-dessous permet d'effectuer cette fonction, il s'agit d'une fonction logique OU si l'on considère le relais fermé comme un niveau bas.



Circuit de sortie

Une LED sur la face avant indique l'état de cette sortie :

- Allumée : Fonctionnement normal ;
- Éteinte : Arrêt d'urgence actif.

4-3.3. Communication avec la partie commande

La communication avec la partie commande se fait via un module IP (Industry Pack) d'entrées-sorties tout ou rien de type UNIDIG-I-O-24I/O (Cf. annexe 5).

Les principales caractéristiques de ce module sont :

- 24 entrées-sorties configurables séparément
- Possibilité de déclencher des interruptions
 - Entièrement configurables (entrées, fronts)
- Entrées et sorties optocouplées
 - Nécessité d'une alimentation externe : 5V pour notre cas

La carte de sécurité dispose de cinq sorties vers ce module :

- Une sortie informant de la présence d'un arrêt d'urgence (I/O-4)
- Quatre sorties codées (codage modifiable) pour la source de l'arrêt d'urgence (I/O-5 à I/O-8)

Source	I/O-8	I/O-7	I/O-6	I/O-5
Arrêt Logiciel	0	0	0	1
Drive Enabled	0	0	1	0
Arrêt d'Urgence	0	1	0	0
Fdc Pelvis	1	0	0	1
Fdc jambe Droite	1	0	1	0
Fdc jambe Gauche	1	1	0	0
Autre	1	1	1	1

Codage en sortie

Ce codage est également utilisé pour visualiser le défaut en face avant de la carte, donc sur l'armoire de commande du robot.

Deux sorties de l'UNIDIG (entrées de la carte de sécurité) sont utilisées :

- Arrêt logiciel (I/O-1) ;
- Réarmement (I/O-2) : redémarrage du robot après un arrêt, il n'est possible que si aucune source d'arrêt d'urgence n'est active. Pour être prise en compte, l'entrée de réarmement doit être active (niveau haut) au moins 300ms.

4-3.4. Programmation du microcontrôleur

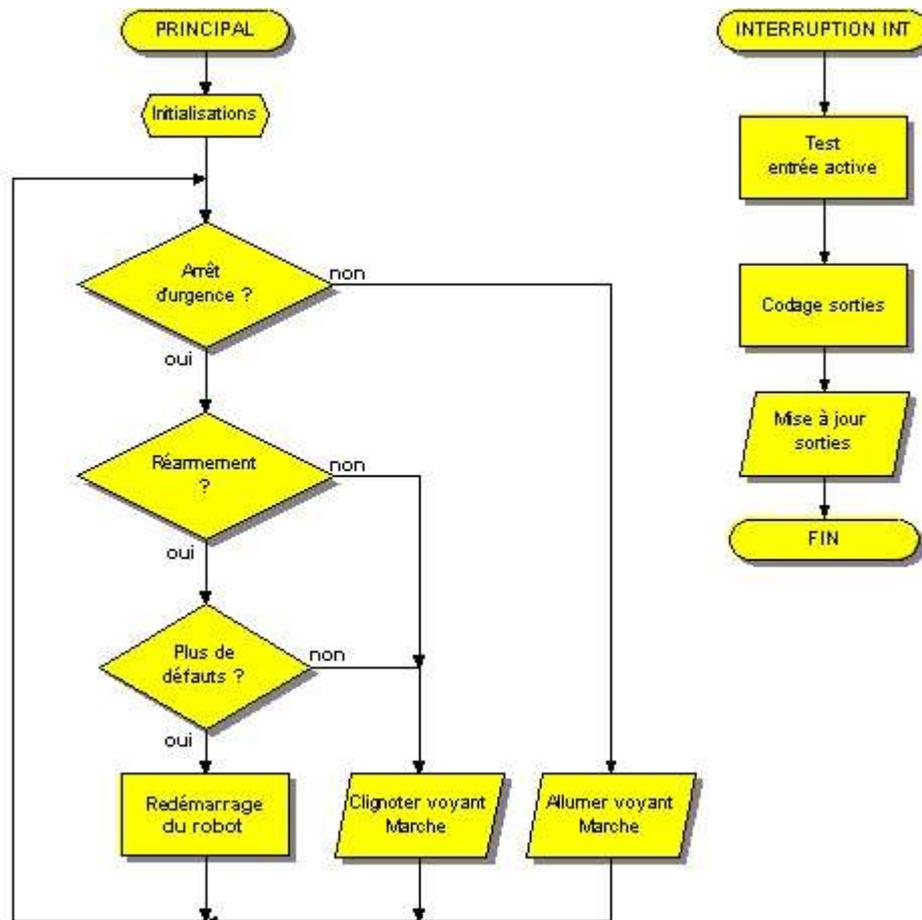
Le PIC a été programmé en langage C avec le compilateur CC5x [11] en s'aidant de cours de programmation [12].

Pour des facilités de développement, la carte de sécurité est pourvue d'un bootloader, c'est-à-dire un petit programme dans le PIC et une liaison série permettant la programmation rapide directement sur site à l'aide d'un logiciel spécifique (Cf. annexe 1).

Un cavalier présent sur la carte permet de choisir le démarrage du bootloader ou du programme normal, il faut veiller à retirer ce cavalier pour permettre le fonctionnement normal.

Pour des raisons de sécurité et de conservation du matériel, la liaison PC pour la programmation du PIC est optocouplée.

L'architecture globale du programme (source disponible en annexe 2) est la suivante :



Rappel : l'interruption INT intervient lors d'un front montant sur PORTB.0

4-4. Réalisation de la carte

Pour des contraintes de temps, une carte à connections enroulées (ou carte wrappée) a été réalisée, elle est placée dans la partie basse de l'armoire de commande.

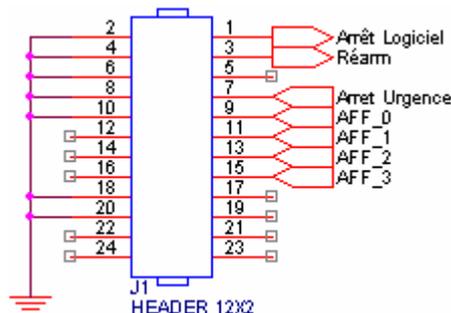
L'utilisation de la méthode du wrapping (Cf. photo page 19 et annexe 3) permet une modification aisée du circuit et donc une évolutivité accrue, de manière à pouvoir répondre à tous les besoins futurs comme la gestion des potentiomètres. C'est un des principaux avantages de ce mode de câblage pour la réalisation de prototypes.

La carte de gestion des sécurités communique avec l'armoire de commande (système VME) du bipède via 24 Entrées/Sorties logiques (module IP). Seules les E/S nécessaires sont câblées, les autres pourront l'être facilement pour les futures évolutions.

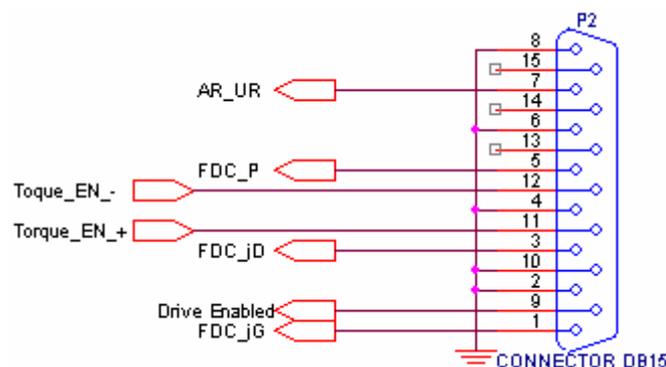
4-4.1. Connectique

Toutes les entrées-sorties de la carte passent par l'intermédiaire de connecteurs pour permettre un démontage facile. On comptabilise un total de 4 connecteurs, chacun d'entre eux est dédié à une fonction particulière :

- Bornier d'alimentation +5V en façade ;
- HE-14 24 broches : liaison vers l'UNIDIG (partie commande) ;

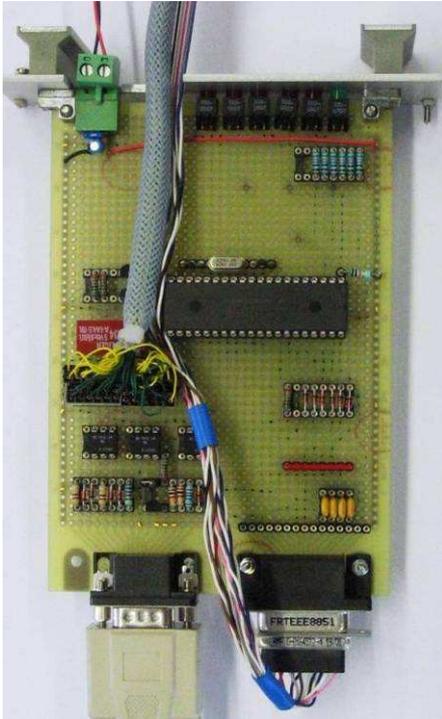


- SUB-D 15 broches mâle : liaison vers le robot (fins de course, variateurs,...) ;



- DUB-D 9 broches femelle : Liaison PC (RS232) pour Bootloader.

4-5. Tests et Intégration

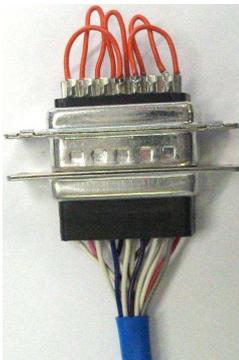


Face avant de la carte de sécurité

Un premier test de la carte seule, en connectant les entrées à des interrupteurs, montre le bon fonctionnement de la carte de sécurité.

Le seul problème observé est le non fonctionnement du bootloader, lié à l'utilisation d'un ordinateur portable qui ne fournit pas les tensions standard en sortie du port série et ne fonctionne donc pas avec l'interface utilisée.

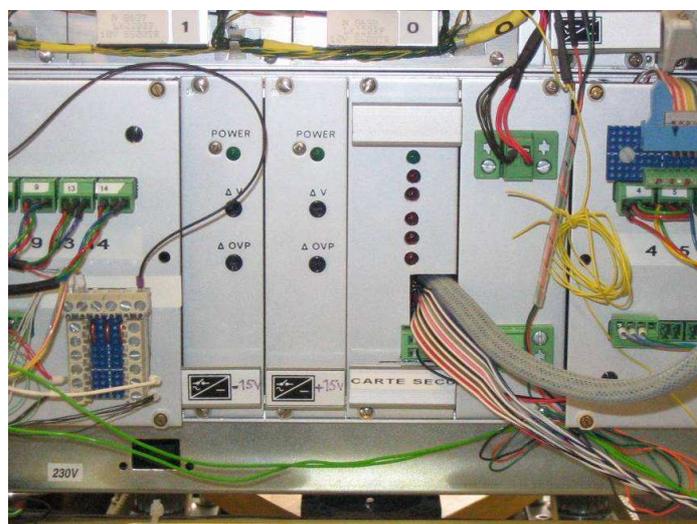
Ce problème ne perturbe pas le fonctionnement propre de la carte de sécurité, le bootloader pourra même être dé-wrappé si la mise en place de nouvelles fonctions nécessite plus de place sur la carte.



Bouchon

Durant la phase d'intégration de la carte, un « bouchon » a été placé sur le câble reliant le robot à la carte (connecteur DB-15). Il a pour fonction de restaurer l'ancienne gestion des sécurités (sécurités câblées) lorsque la carte de sécurité n'est pas opérationnelle.

De cette façon, les expérimentations sur le robot peuvent être effectuées, la gestion des sécurités est toujours garantie.



Intégration de la carte de sécurité dans l'armoire de commande

5- Conclusion

Ce stage, d'une durée de deux mois et demi, m'a permis de m'intégrer parfaitement dans un laboratoire de recherche, un milieu professionnel que je ne connaissais pas du tout.

En effet, la recherche en robotique est un domaine peu connu du grand public, hormis les résultats finis qu'ils produisent (exemple : robots humanoïdes asiatiques).

Relativement autonome dans mon travail, l'échange de connaissances et de techniques s'est effectué dans les deux sens. J'ai pu – entre autres – aussi bien apprendre de nouvelles techniques de réalisation qu'introduire à l'équipe l'utilisation du PIC pour de petites applications.

La réalisation de cette carte s'inscrit dans le développement continu du robot destiné à fournir un support expérimental fiable aux chercheurs l'utilisant. Dans cette optique, ce stage m'a permis de connaître le fonctionnement de ce type de robots.

Mon stage a abouti sur une première version de la carte de sécurité intégrée au robot et fonctionnant correctement. De plus, elle a été conçue de façon à ce que les ingénieurs du service ou les futurs stagiaires puissent étendre ses fonctionnalités.

Ce stage est donc une réussite, il est pour moi bien plus qu'une simple immersion dans le monde professionnel. Je le considère comme une véritable première expérience professionnelle dans un milieu que j'affectionne tout particulièrement.

6- Bibliographie et liens Web

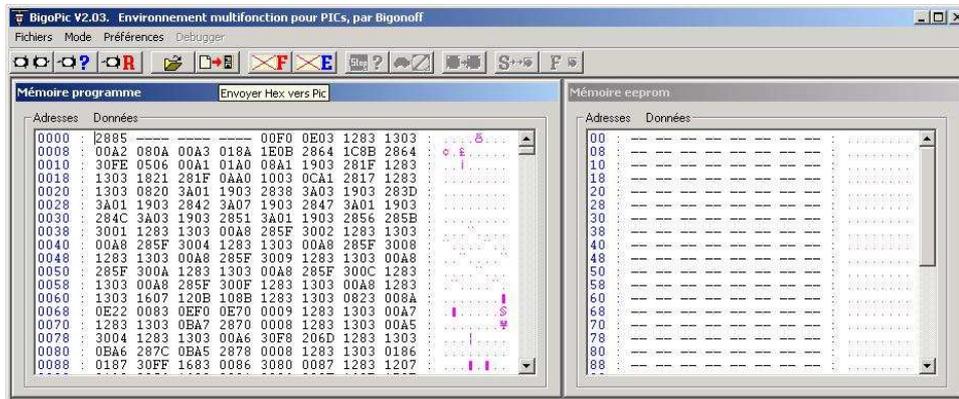
- [1] INRIA France www.inria.fr
- [2] INRIA Rhone Alpes www.inrialpes.fr
- [3] ZIRST www.zirst.com.
- [4] Lyon Technopôle <http://www.techgrandlyon.prd.fr>
- [5] IMAG www.imag.fr
- [6] Service SED www.inrialpes.fr/sed
- [7] Projet BIPOP www.inrialpes.fr/bipop
- [8] Projet DEMAR <http://www.lirmm.fr/~w3rob>
- [9] PARVEX, « SERVOAMPLIFICATEUR SBS 2^{ème} génération », 09/1997.
Documentation partielle disponible en annexe 6.
- [10] GreenSpring Computers, « IP-Unidig-I-O User Manual », SBS, 03/1996.
Documentation partielle disponible en annexe 5, <http://www.sbs.com>
- [11] B Knudsen Data, Compilateur CC5xfree, <http://www.bknd.com/cc5x>
- [12] Bigonoff, « La Programmation des PICs par Bigonoff » 1^{ère}, 2^{ème} et 3^{ème} partie,
<http://www.bigonoff.org>, 2004.
- [13] G.Baille – P.Di Giacomo – H.Mathieu – R.Pissard-Gibollet, « L'armoire de commande du robot bipède BIP2000 » N°0243, INRIA, 07/2000.
- [14] C.Azevedo – R.Pissard-Gibollet, « Le contrôleur du robot BIP2000 » N°0249, INRIA, 04/2001.

7- Annexes

- 7-1. Logiciels utilisés
- 7-2. Programme PIC
- 7-3. Technique du wrapping
- 7-4. Schémas de la carte
- 7-5. Documentation UNIDIG-I-O-24I/O partielle
- 7-6. Documentation variateurs PARVEX SBS partielle

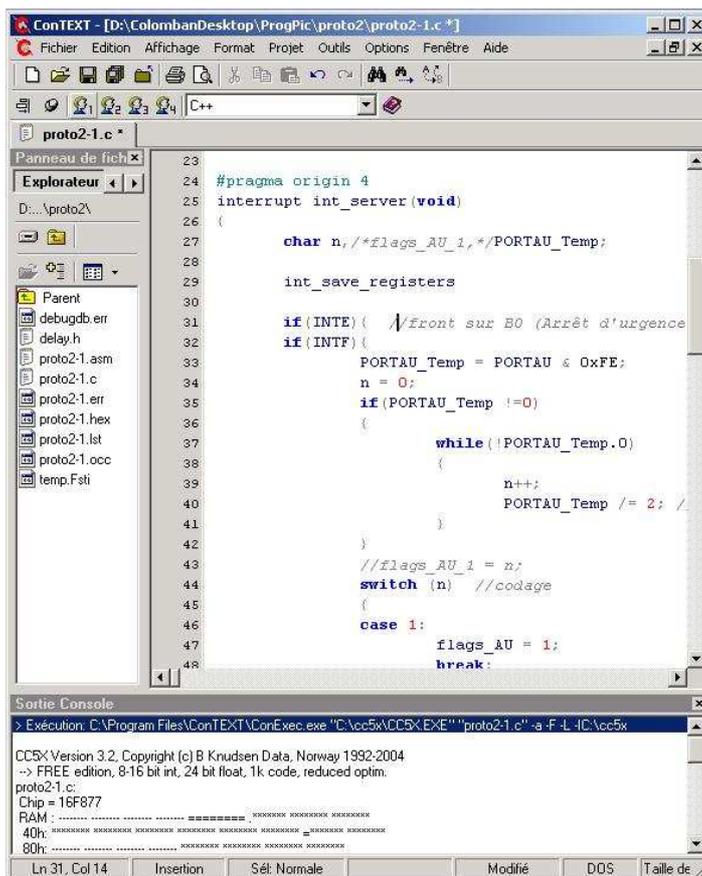
7-1. Logiciels utilisés

Bootloader BigoPic V1 ou V2 : Bigonoff : www.bigonoff.org



Compilateur C CC5x version gratuite : www.bknd.com/cc5x

Editeur de texte orienté programmation : ConTEXT : www.fixedsys.com/context



7-2. Programme PIC

```

/*      Version finale du programme de la carte de sécurité :
*      Détection front montant sur RB0 et lecture du portB
*      -> activation arrêt d'urgence et affichage codé
*      2 modes de fonctionnement dans le main
*      -> 1° : voyant clignotant pour réinitialisation
*      -> 2° : voyant fixe pour fonctionnement normal
*/
#pragma chip PIC16F877
#include "int16cxx.h"
    //définition des entrées-sorties
#pragma char PORT_S @ PORTD //port de sorties
#pragma char PORT_E @ PORTC //port d'entrées
#pragma char PORT_AU @ PORTB //port d'entrée des défauts

#pragma bit AR_UR @ PORT_S.4 //sortie d'arrêt d'urgence
#pragma bit voy_marche @PORT_S.5 //sortie voyant
#pragma bit rearm @ PORT_E.0 //entrée de réarmement

char flags_AU;

#pragma origin 4
interrupt int_server(void)
{
    char n, PORT_AU_Temp;

    int_save_registers

    if(INTE){ //front sur B0 (Arrêt d'urgence global)
    if(INTF){
        PORT_AU_Temp = PORT_AU & 0xFE;
        n = 0; /////// Test entrée active /////
        if(PORT_AU_Temp !=0)
        {
            while(!PORT_AU_Temp.0)
            {
                n++;
                PORT_AU_Temp /= 2; //décalage à droite
            }
        }
        switch (n) /////// Codage sorties /////
        {
        case 1:
            flags_AU = 15; //normalement rien
            break;
        case 2:
            flags_AU = 1; //Arrêt logiciel
            break;
        case 3:
            flags_AU = 2; //Drive Enabled
            break;
        case 4:
            flags_AU = 4; //Arrêt d'urgence
            break;
        case 5:
            flags_AU = 9; //FDC_jGauche
            break;
        case 6:
            flags_AU = 10; //FDC_jDroite
            break;
        }
    }
}

```

```

        case 7:
            flags_AU = 12;        //FDC_Pelvis
            break;
        default:
            flags_AU = 15;        //mauvaise détection
            break;
    }
    PORT_S = flags_AU; /////// Mise à jour sorties /////
    AR_UR = 1;
    INTE = 0;
    INTF = 0;
}}
int_restore_registers
}

#include "delay.h"

void main(void)
{
    OPTION = bin(11000000);
    ADCON1 = 0x06;                //pas d'analogique

    PORTA = 0;
    TRISA = 0;
    PORTB = 0;
    TRISB = 0xFF;
    PORTC = 0;
    TRISC = 0x01;
    PORTD = 0;
    TRISD = 0;

    AR_UR = 0;
    flags_AU = 0;
    INTE = 1;                    //interruption RB0 active
    GIE = 1;                    // Global interrupt enable

    if(PORT_AU !=0)                //test au démarrage
        {INTF = 1;}
    while(1)
    {
        if(!flags_AU)                /////// Arrêt d'urgence ? /////
        {
            //oui
            if(rearm)                /////// Réarmement ? /////
            {
                //oui
                if(!PORT_AU)        /////// Plus de défauts ? /////
                {
                    //oui
                    flags_AU = 0; /////// Redémarrage du robot /////
                    PORT_S = 0;
                    INTF = 0;
                    INTE = 1;
                }
            }
            //non
        }
        //non
        voy_marche=!voy_marche; /////// Clignotement voyant /////
        DelayMs(255);
    }
    else                            //non(normal)
    {
        voy_marche = 1;                /////// Allume voyant /////
    }
}
}

```

7-3. Technique du wrapping (connections enroulées)

La réalisation de cartes par la méthode du wrapping consiste à réaliser les connections entre les différents composants par du fil enroulé sur les broches voulues (photo 1).

Cette technique de liaison sans soudures présente de nombreux avantages pour la réalisation de prototypes ou non :

- Forte résistance aux vibrations, longue durée dans le temps
- Modification aisée en dé-wrappant le(s) fil(s) sans détériorer la qualité du circuit.

Le wrapping se fait par des outils spécifiques, électriques (photo 2) ou manuels (photo 3) sur des supports de composants spéciaux ayant de longues pattes côté cuivre (photo 4).

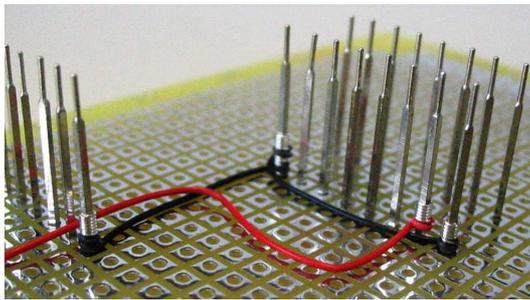


Photo 1

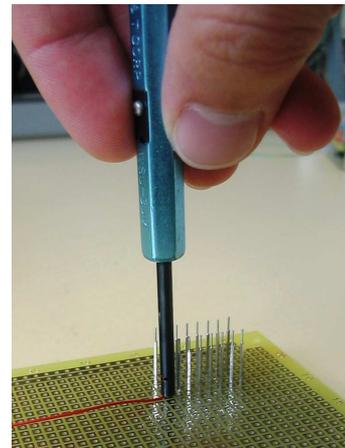


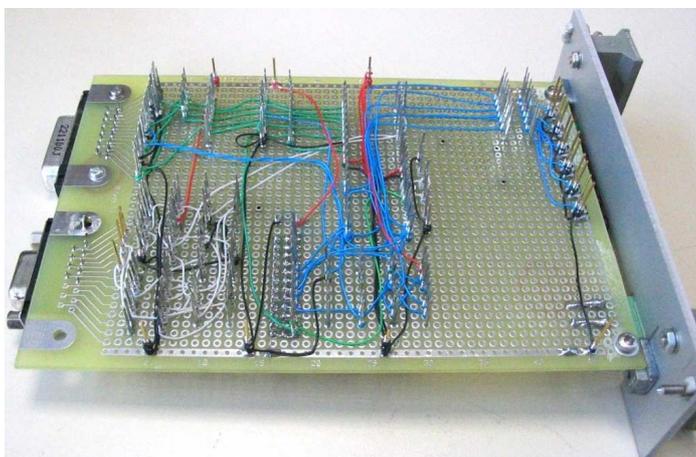
Photo 3



Photo 2



Photo 4



La carte de sécurité wrappée

7-4. Schémas de la carte

