Université de Montpellier 2,
I.U.T. de Nîmes,
Département G.E.I.I

Fachhochschule
Braunschweig/Wolfenbüttel
Fachbereich Maschinenbau
Fachbereich Maschinenbau
Campus Wolfenbüttel

RAPPORT DE STAGE en ALLEMAGNE:
stage de fin de 2e année.

Acquisition de déplacement par un microcontrôleur fujitsu.

Année 2005/2006

CHAMBON Mathieu
SIONG François
Tuteur de stage : Prof. Dr.-Ing. Rolf Roskam

Sommaire:

Sommaire suite:

Préambule :

Ce compte rendu est destiné à présenter le travail que nous avons accompli durant les trois mois de notre stage en fin de seconde année de DUT GEII, effectué dans l'université allemande de Wolfenbüttel.

Il retrace donc toutes les démarches que nous avons effectué, pour mener à bien le projet que nous a confié notre tuteur de stage, le professeur Roskam.

Le but de notre stage est d'apprendre à mener à bien un projet, sous la tutelle d'un professeur, mais en totale autonomie, avec ici une nouvelle barrière, celle de la langue.

Le projet que nous a confié le professeur Roskam a pour principal but de de nous faire découvrir les utilisations concrètes d'un microcontrôleur.
Notre travail a été de concevoir de A à Z un système embarqué pour gérer le déplacement d'un objet déplacé par un moteur électrique.

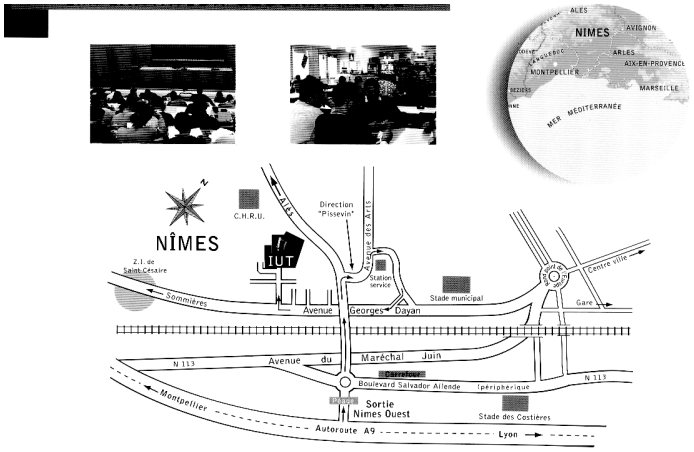
Nous sommes donc passés par les étapes de conception et de réalisation d'une carte électronique construite autour d'un microcontrôleur, ensuite nous avons réalisé la programmation de ce microcontrôleur, afin d'acquérir et de traiter les données en provenance des capteurs de déplacement.

Nous avons aussi réalisé un site Internet disponible en trois langues, pour présenter aux étudiants qui passeront après nous à la ville de Wolfenbüttel.

Nous ouvrirons ce rapport avec une présentation de notre université d'origine : l'I.U.T de Nîmes, puis nous présenterons l'université qui nous a accueilli en Allemagne : la Fachhochschule de Wolfenbüttel ainsi qu'une présentation rapide mais essentielle de la région de Wolfenbüttel.

Présentation de l'I.U.T. de Nîmes:

L'I.U.T. de Nîmes est situé dans le quartier de St Césaire, il est depuis 35 ans un grand pôle de formation d'étudiants.



Il comprend cinq départements de formation : **G**énie **E**lectrique et **I**nformatique **I**ndustrielle créé en 1968, **G**énie **C**ivil et **G**énie **M**écanique et **P**roductique créés en 1969, plus récemment: **G**estion des **E**ntreprises et **A**dministrations en 1991 et **S**cience et **G**énie des **M**atériaux en 1996.

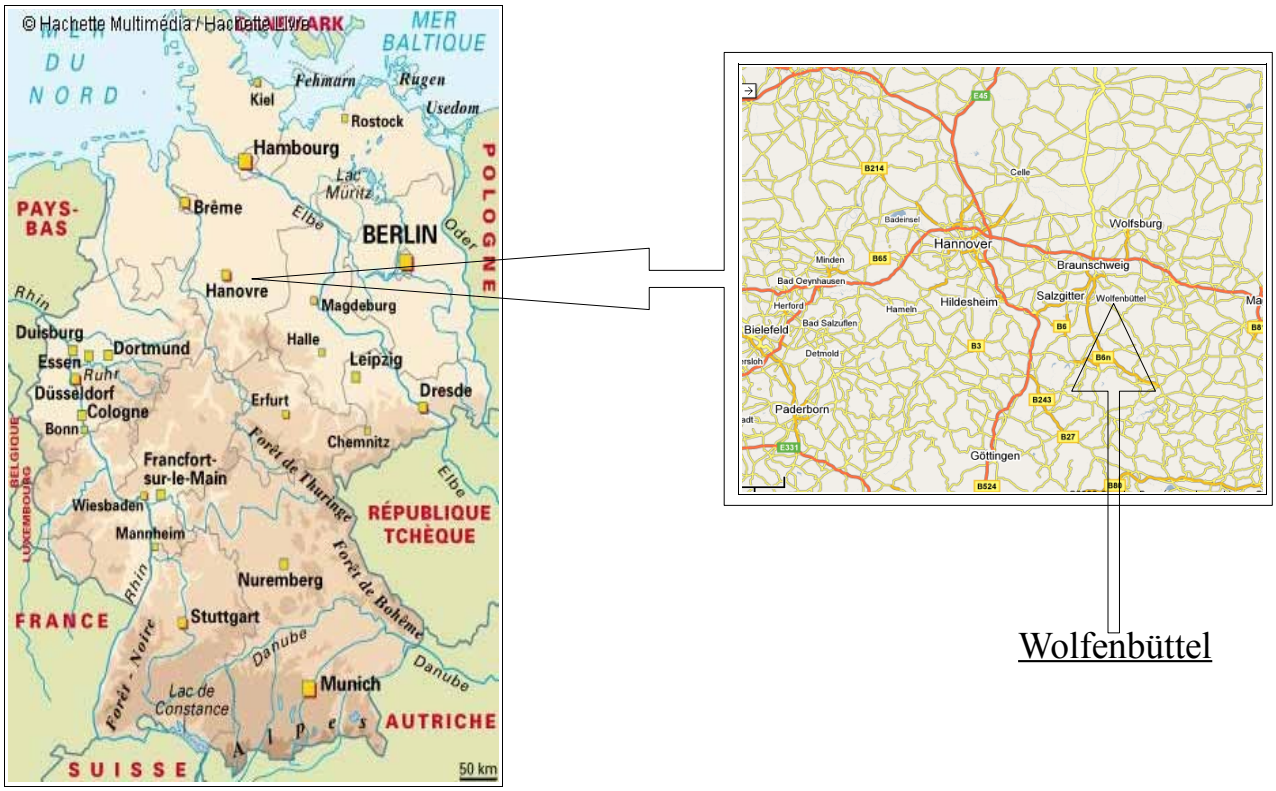
Rattaché à l'université Montpellier 2, l'établissement Nîmois possède cependant une autonomie administrative et financière et accueille plus de 1100 étudiants.

L'I.U.T. en quelques chiffres:

Il emploi **45 enseignants de l'enseignement supérieur , 46 de l'enseignement secondaire** et fait appel à **161 vacataires du monde professionnel.**

De plus, **43 IATOS** assurent la bonne marche de l'établissement qui dispose d'un budget **2,5 millions d'euros.**

Nous avons donc atterri à Wolfenbüttel, située en plein milieu de l'Allemagne dans les les faux bourgs de Hanovre, à 250 kilomètres à l'ouest de Berlin.



Présentation de wolfenbüttel et de sa région:

Wolfenbüttel est une ville de taille moyenne, située au coeur d'un des plus grand bassin industriel d'Allemagne, après celui de la célèbre Ruhr allemande.

En effet, on retrouve dans cette région de nombreuses grandes multinationales comme VolksWagen, fujitsu-Siemens ou encore Bosh.

Ces entreprises ont toutes implantées de grandes usines dans la région, Fujistsu Siemens est implantée a Braunschweig et les usines VolksWagen ont même nécessité la construction d'une ville pour leurs employés à 40 km de Wolfenbüttel cette ville s'appelle Wolfsburg. C'est dans cette ville que se trouve la plus grosse usine de production de la marque.

Ces entreprises participent activement au développement de l'université en fournissant du matériel, des fonds et de nombreux stages aux étudiants allemand dans l'optique d'une embauche.

Présentation de la Fachhochschule :

À gauche la statue de la Fachhochschule.

A droite le bâtiment principal où se trouvent, des bureaux administratifs et des salles de cours.

	Université des sciences appliquées de Braunschweig/Wolfenbüttel
Adresse:	Salzdahlumer Street 46 / 48
Code postal :	38 302
Ville:	Wolfenbüttel
Pays :	Germany
Numéro de téléphone :	0049 5331 939 6001
Numéro de fax :	0049 5331 939 6002

La Fachhochschule n'est pas implantée qu'a Wolfenbüttel, comme son nom l'indique, elle est aussi présente à Braunschweig mais aussi à Wolfsburg et à Salzgitter qui a été comme Wolfsburg construite pour les besoin d'une usine VolksWagen.

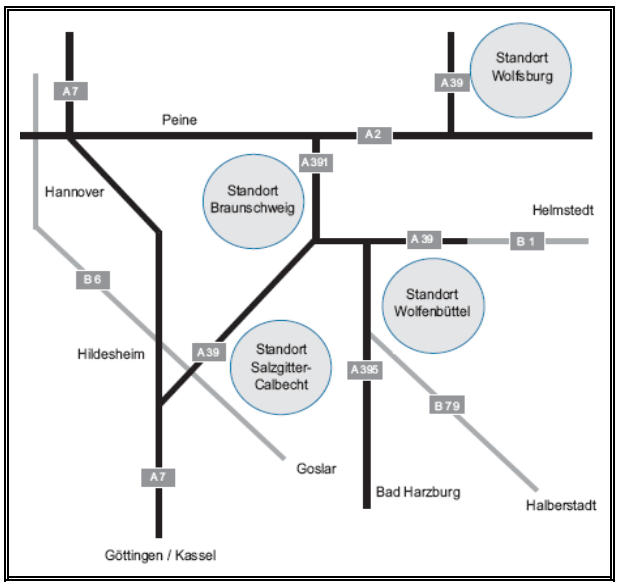


Schéma 1: Les quatre campus de la Fachhochschule le Braunschweig/Wolfenbüttel

Les différents départements :

Chaque campus de l'université possède ses propres départements :

On retrouve sur le Campus de Braunschweig:

- Le département des affaires sociale

sur le Campus de Salzgitter:

- Le département du transport, des sports, du tourisme et des médias

sur le Campus de Wolfenbüttel se trouvent:

- Le département de génie électrique.
- Le département des sciences de l'ordinateur.
- Le département de génie mécanique.
- Le département de droit.
- le département de maschinenbau dans lequel nous avons été accueillis et que nous présenterons dans quelques pages.

enfin sur le Campus de Wolfsburg:

- Le département des affaires sociales.
- Le départements de Productique.
- Le département de gestion des entreprises.

Histoire de la Fachhochschule:

La Fachhochschule de Brauschweig/wolfenbüttel a été crée après les débats politiques sur l'éducation tenus dans les années 60 en Allemagne.

En effet, la nécessité d'aider l'industrie allemande à maintenir sa compétitivité dans le domaine international a mené à une demande croissante d'un personnel mieux qualifié, avec des connaissances pratiques dès la sortie du cursus universitaire.
Cette demande a donc donné son point de départ à la Fachhochschule.

Nous allons maintenant retracer l'histoire de la Fachhochschule depuis 1905 jusqu'en 2004 dans le tableau suivant.

Chronologie:

1905 :	Fondation de l'école chrétienne pour des femmes à Braunschweig (Christlich-Sozialen Frauenschule) (1939 école provinciale pour Welfare public, 1947 université d'état providence de la basse-saxe, 1966 universitésupérieure pour le travail social de la basse-saxe)
1928 :	Fondation de l'école d'ingénieurs (technique), un établissement privé qui offre 5 semestres de cours avec un examin final reconu par l'état dans l'électrotechnique mécanique à Wolfenbuttel.
1971 :	L'académie de technologie a été convertie en université de la Science appliquée ; Base du département des affaires sociales par fusion avec l'universitésupérieure pour le travail social à Braunschweig
1972 :	Création du département d'approvisionnement technologique .
1988 :	Création de l'institut pour la construction de véhicule dans Wolfsburg, nouveau campus de l'université de la Science appliquée
1991 :	Création du département de la gestion d'entreprise ; Approbation des instituts pour la Réutilisation et l'Informatique de Production (plus tard l'informatique industrielle)
1993 :	Création du département des Services de santé publique: dans les bâtiments de l'ancienne caserne de Northampton dans Wolfenbüttel
1993 :	Création d'un quatrième campus à Salzgitter ; Formation du département des transports et de la la gestion du trafic
1994 :	Création du département des Services de santé publique .
1995 :	Ouverture du département de maschinenbau.
2000 :	Création du département de droit.
2004 :	Le département de la gestion de médias, de sport et de tourisme se transforment en Transporter-Sport-Tourisme-Médias de Salzgitter de Karl-Scharfenberg-Corps

L'activité principale de l'université est de constituer un personnel compétant pour le marché du travail de la région. Mais le travail de l'université ne s'arrête pas là:

- ✔ Suivi des étudiants,
- ✔ Possibilités de fin d'études a l'étranger, avec un réseau très développé avec une multitude de coopérations internationales dans l'enseignement supérieur.
- ✔ Le sport universitaire est mis en avant, avec des aménagements pour les sportifs de haut niveau.
- ✔ La recherche est également importante dans l'établissement. Cette recherche est basée sur des applications très concrètes. L'établissement accompagne des projets du développement pour l'industrie comme pour le gouvernement.

La Fachhochschule de Brauschweig/Wolfenbüttel est un employeur important de la région, et emploie 526 personnes : membres du corps enseignant et personnel.

Le département machinebau :

Nous avons été accueillis dans le département machinebau, dans les laboratoires de mechatronik de la Fachhochschule de Wolfenbüttel.
La traduction de machinebau directement en français est impossible.
Cependant on enseigne dans ce département la mécanique, électricité générale, l'électronique, les réseaux locaux, la pneumatique et l'automatique.

Les étudiants de ce département peuvent choisir certaines options pour se spécialiser en fonction de leurs préférences.
Parmi les options possibles, il y a des TPs de mechatronik,le département de machinebau possède deux laboratoires de mechatronik.
Cet enseignement réunnit les compétances des mécaniciens, des électrotechniciens et des électrotecnicien autour de l'étude des moteur,et de leurs asservissement par des microcontroleurs.

Les laboratoires de mecharonik :

Voici le laboratoire du professeur Roskam en images :


Il s'agit d'un espace de travail d'environ 35m carrés avec deux rangées de postes informatiques,sur chacun se trouvent le travail d'un binôme allemand et est relié à la maquette correspondant à leur projet.
Le professeur Roskam donne des cours trois fois par semaines dans ce laboratoire, et les étudiants peuvent avoir accès à leur guise, il suffit de lui demander et chercher les double des clefs au "Schüsselausgabe" à traduire par bureau des clés.
On n'y trouve les doubles de toutes les portes de l'université, classé dans un immense coffre fort à tiroirs.
Nous avons réalisé l'intégralité de notre stage dans ce laboratoire, de l'étude du projet à la programmation, en passant par la soudure de la carte électronique. En plus des postes informatiques, nous avions à notre disposition des oscilloscopes et un poste de soudure, que nous présenterons sommairement dans la partie pratique du rapport.

Introduction :

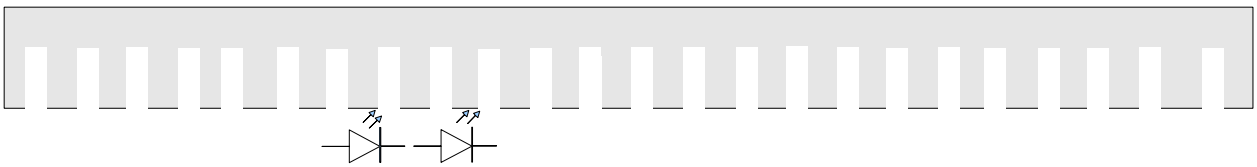
Dans le programme de notre deuxième année d'études en DUT GEII il est inclus une période de trois mois réservés aux stages.

Ces stages peuvent s'effectuer dans une entreprise en France ou à l'étranger, ou alors dans une université partenaire de l'IUT de Nîmes comme l'est la Fachhochschule de Wolfenbüttel .
Nous sommes quatre de notre département à avoir choisi d'effectuer un stage dans cette université allemande.

À notre arrivée dans le département de mechatronik,, nous avons formé deux binômes et nos tuteurs nous ont présentés les sujets de nos stages respectifs.
L'autre binôme a pour sujet d'étude le réseau de terrain Ethernet PROFIBUS.

Quant à nous, nous devons concevoir un système embarqué construit autour d'un microcontrôleur fujitsu.
Un système embarqué est un produit dont l'encombrement est réduit au maximum pour pouvoir faire partie d'un système plus évolué comme par exemple dans le cadre de l'automobile, où l'électronique est de plus en plus omniprésente.

Le principe de fonctionnement de notre système est simple : nous avons à traiter des signaux de nature rectangulaire en provenance de capteurs de déplacement, placés devant une règle crantée de un mètre articulée par un moteur électrique.



Capteur A // Capteur B

Avec deux signaux nous devons être capable de mesurer le déplacement établi, le sens de déplacement, ainsi que les changements éventuels de direction.
Ces signaux proviennent de deux capteurs positionnés l'un à côté de l'autre, les signaux ainsi obtenus sont donc décalés l'un par rapport à l'autre, ce qui nous permettra de définir le sens de déplacement.

Dans cette partie nous allons dans un premier temps présenter le cahier des charges de notre projet, ensuite nous parlerons du matériel qui a été mis à notre disposition par la Fachhochschule. Enfin avant d'en arriver au projet en lui-même, nous expliquerons les principes de fonctionnement des logiciels essentiels à ce projet.

I/Cahier des charges du projet :

-Conception et réalisation de la carte électronique autour du microcontrolleur :

1_Etude du microprocesseurs et de sa carte.

2_Création du schéma électronique du système à réaliser.

3_Commandes des composant.

4_Fabrication de la carte: soudure des composants.

5_Test de la carte.

-Programmation du microcontrolleur :

1_Compréhension des logiciels utilisés et familiarisation avec ceux-ci.

2_Programmation en langage C .

3_Test du programme.

4_Localisation des erreurs.

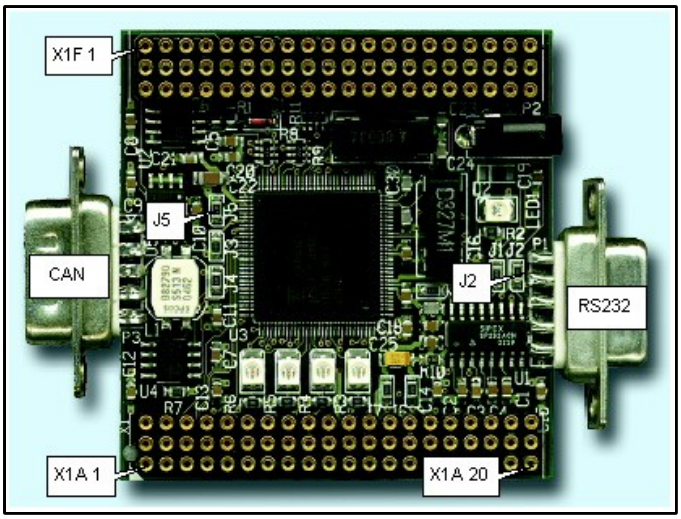
5_Programme fini : ajout d'améliorations.

II/Présentation du matériel.

Nous n'allons pas faire ici une présentation détaillée de tous les instruments mis à notre disposition par la Fachhochschule, comme les ordinateurs, les oscilloscopes, ou encore le fer a souder, mais nous avons ici présenté les composants électroniques qui nous ont été fournis pour mener à bien notre projet.

La starter board du microcontrôleur MB91364G :

Voici donc l'élément principal autour duquel c'est articulé notre stage : le microcontrôleur Fujitsu MB91364G sur son support appelé starter board .

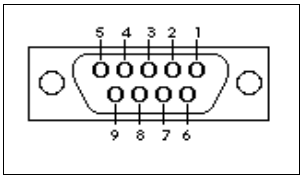


La photo à gauche étant à taille réelle, on comprend l'intérêt d'implanter ce microcontrôleur sur une carte, afin de pouvoir réaliser des soudures dans un montage électronique.

En effet le microprocesseurs M.B. 91364 est pourvu de 120 pattes, réparties autour d'un composant mesurant à peine plus d'un cm².

De plus, d'autres composants ont été ajoutés afin de rendre cette carte plus optionnelle.

Les deux ports série de types COM de la carte permettent une interface facile avec n'importe quel ordinateur équipé d'un port COM.



C'est la société Accemic basée à Munich qui fabrique une large gamme de starter board et met aussi à disposition les programmes d'interface avec le microcontrolleur dont nous parlons dans le chapitre suivant.

Un starter board coûte à lui seule 110 €, mais sur commande par Internet, avec six mois de garantie, et un environement logiciel complet, son prix est alors de 800 €.

Le microcontrôleur MB 91F364 G de fújitsu:

La documentation de se microcontrolleur fait plus de 800 pages, nous ne rentrons donc pas ici dans les détails, car l'essentiel est de présenter ses caractéristiques principales.
Vous pouvez consulter la documentation du constructeur en allant dans les ANNEXES:

Le MB 91F364 G est un Microcontrôleur 32 bits de la série FR, avec une architecture RISC.
C'est un produit du milieu de la gamme de microcontrôleurs fújitsu.
Mais il n'en reste pas moins un outil de travail puissant avec une fréquence de travail à 64 Mhz.

Sa technologie 32-bits permet également de traiter des informations de manière précise, plus rapidement, grâce à une mémoire RAM de 12KB et une mémoire flash de 256 KB.

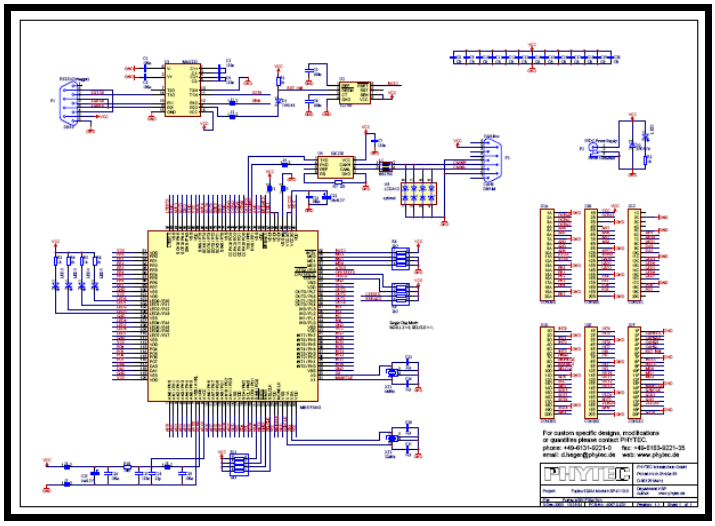


Schéma de la Starter board articulée autour du microcontrôleur

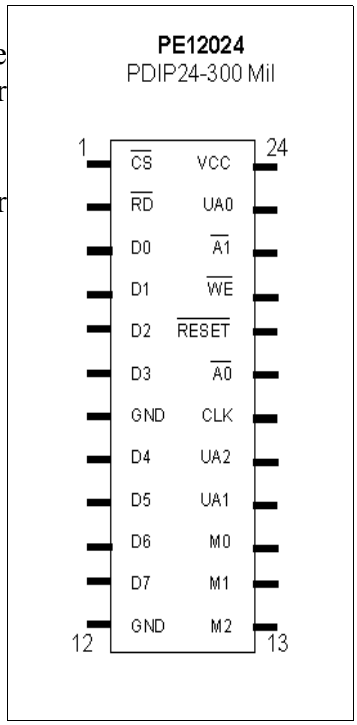
Le microcontrôleur MB91F364 G est donc tout à fait adapté à nos besoins, sa puissance de calcul est même largement supérieure à nos besoins, en effet se composant est utilisé pour des taches bien plus complexe que la nôtre niveau industriel, ses capacités ne seront donc pas mises en défaut lors de notre projet. Mais le professeur Roskam a pour objectif de faire évoluer notre travail au cours des prochaines années.

CONTROLLER FEATURES :

- 32-bit Fujitsu FR50 RISC Core **64MHz,**
- Compatible avec les logiciels FR30,
- 12 KB de mémoire RAM,
- 4 KB F-bus RAM ,
- 256 KB de mémoire Flash ,
- 2 KB de mémoire ROM ,
- trois modes différent d'horloge,
- Mode d'économie de puissance
- 4 canaux 16-bit pour le timer PWM,
- 12 canaux ADC avec une resolution de 10-bits,
- 2 canaux DAC avec une resolution de 10-bits,
- Un Convertisseur Analogique Numérique CAN,
- Un port I2C
- 4 canaux 16-bit d'entrée,
- 4 canaux 16-bit de comparaison de sorties,
- Entrées séries: I/O,
- 2 canaux UART avec l'option LIN,
- UART
- RTC
- 8 sorties pour des LED

L'encodeur incrémental PE12024 :

L'encodeur incrémental 24 bits PE12024 nous permet de réaliser un comptage des signaux d'entrée, et fournis au microcontrôleur des données stockées sur 24 bits à l'aide d'un port 8 bits.



Pour y accéder, le microcontrôleur doit donc signaler à quel octet il veut avoir accès, à l'aide des deux entrées /A0 et / A1 en appliquant la règle suivante :

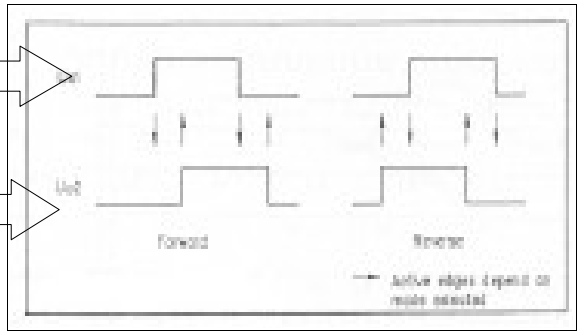
> (/A0=/A1=HIGH), followed by the LSB+1 (/A0=LOW, /A1=HIGH) and then the MSB (/A0=HIGH, /A1=LOW).

Ce qui signifie : quand /A0 et /A1 sont a l'état haut, on lit l'octet de poids faible (LSB) , pour l'octet du milieu (LSB+1) on met /A0 a l'état bas, et pour l'octet de poids fort, /A0 =1 et / A1=0.

L'interface de l'encodeur incrémental PE12016/12024 peut déterminer la direction ou le déplacement d'un dispositif mécanique a l'aide de deux signaux d'entrées provenant des capteurs.
En effet grâce a son architecture, il permet en plus d'un comptage, de déterminer le sens de déplacement du mobile, si les deux signaux sont comme le montre le schéma ci dessous, un peut décalés l'un par rapport a l'autre.

Signal A :

Signal B :



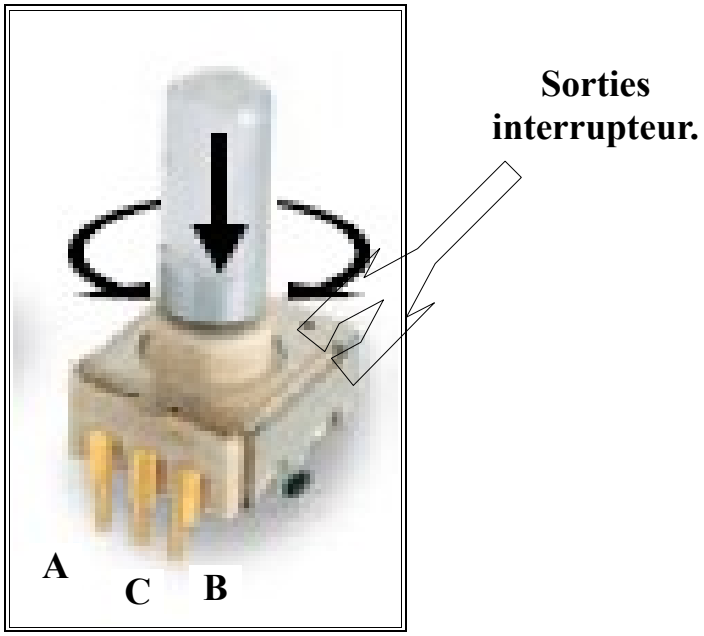
Ce dispositif est conçu pour l'usage dans beaucoup de systèmes sur microprocesseur.

L'encodeur incrémental série E27 :

Ce composant qui ressemble à un potentiomètre, permet de remplir deux fonctions.

La fonction interrupteur : en appuyant sur le haut du composant, on agit sur un contact à fermeture, ce que relie les deux sorties a l'arrière du composant.

La fonction génération de signaux rectangulaires : si on tourne l'axe du composant, on effectue une alternance d'ouvertures et de fermetures entre les deux sorties(A et B) et la patte C qui est reliée au +5V.
Cela permet de générer des signaux, qui même s'ils ne sont pas périodiques, peuvent être traités par le compteur.
Seulement, le contact entre une les deux sorties et la patte C est légèrement retardé par rapport à l'autre en fonction du sens du mouvement.
Cela permet au compteur de définir le sens de déplacement.



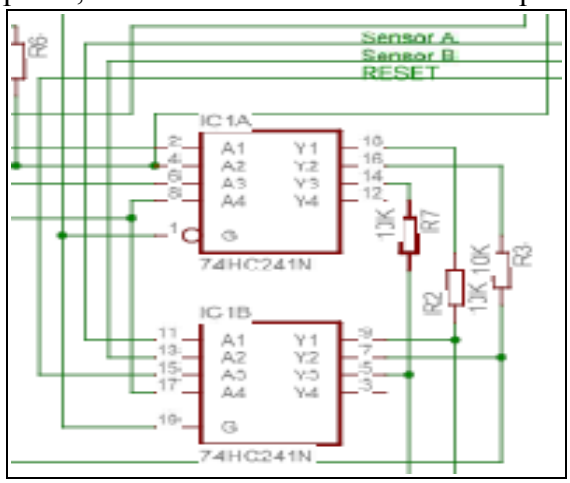
Dans notre montage, la partie bouton poussoir de ce composant sert à la remise à zéro du compteur par l'entrée A0.

Et les signaux A et B générés par une rotation de l'axe sont utilisable pour simuler les futurs signaux provenant des capteurs pour mettre au point la maquette, ce seront donc dans la phase réalisation nos signaux d'entrée.
Cependant, cela servait uniquement pour les tests de vérification du bon fonctionnement de notre montage et de notre programmation.

A l'avenir, cet encodeur incrémental permettra de donner une valeur au compteur afin de positionner un axe hydropique avec une précision de 24-bits, soit pour un axe d'un mètre, une précision au micromètre près.

Le CD74HCT241E :

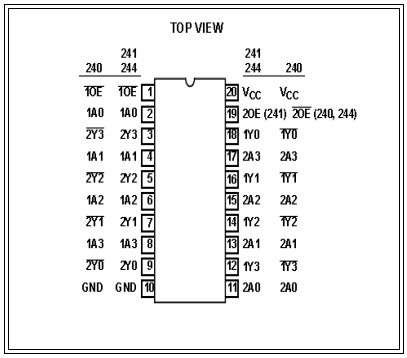
Tout d'abord, le circuit logique CDHCT241E permet d'aiguiller des données. Nous allons vous expliquer son fonctionnement pour notre projet.
Ce circuit logique est composé de deux parties séparées : deux fois quatre entrées et une entrée de validation. Sur l'une des deux partie, cette entrée de validation est complémentée .



Cela permet de choisir quelles entrées seront reliées aux quatre sorties, en utilisant seulement un seul bit de sortie du microcontrôleur.
Cette particularité du circuit se révèle d'une grande efficacité car il nous permet de choisir de quel dispositif nous allons compter les signaux.

Dans notre cas, nous avons deux signaux à récupérer(A et B), ils seront donc reliés aux entrées 2 et 4 du CD74HCT241E pour ceux provenant de l'encodeur incrémental et 11 et 13 pour les signaux externes.
En sortie, nous retrouvons les même des signaux A et B qu'en entrée, donc ce circuit n'apporte aucune modification sur les signaux d'entrée.
Nous choisissons quels signaux utiliser a l'aide d'un bit du port J du microcontrôleur.

Nous avons également utilisé l`entrée 6 et 15 du CD74HCT241E pour pouvoir faire la remise à zéro du compteur. Le signal de remise á zéro ce fait par le signal de l'encodeur incrémental. Ce signal est obtenu en appuyant sur le bouton du composant.
Il peut être utilisé dans les deux cas car il est relié aux deux parties du circuit logique.



II/Présentation des logiciels:

Tous les logiciels que nous avons utilisés ont été développés pour aider la programmations des microcontrôleurs Fujitsu.

Le premier programme Easy code est un compilateur C spécialisé dans la réalisation de programme court a faible utilisation de mémoire, en fait, il est le seul logiciel dont nous avons pu télécharger une version d'évaluation, mais celle-ci ne permet pas d'enregistrer son travail.

Nous avons donc pu nous familiariser un peu avec le logiciel chez nous, mais toute l'écriture du code a dû être faite à la Fachhochschule.

Le deuxième programme Softune est fourni par la société Fujitsu lors de l'achat du microcontrôleur, il permet de convertir le langage C provenant d'easy code en langage assembleur ce qui rend donc le programme utilisable par le microcontrolleur.

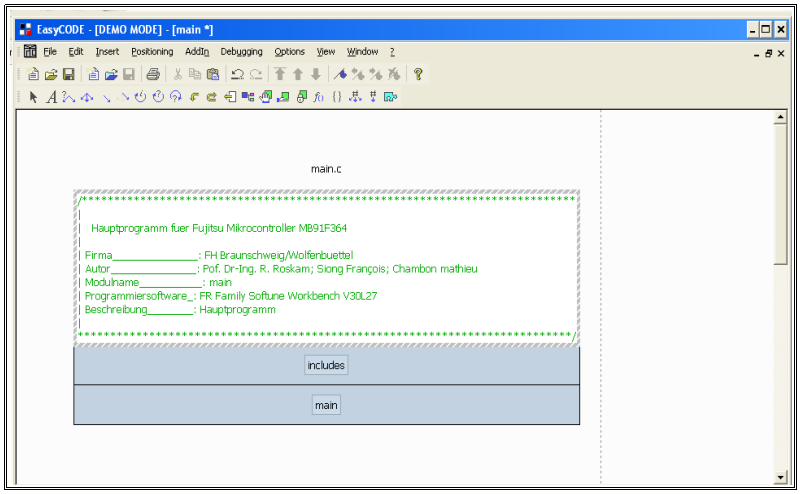
Le troisième, Accemic permet lui de transférer le code assembleur venant du logiciel Softune dans la mémoire du microcontrôleur et de contrôler toutes les broches du composant.

Enfin le dernier programme, MMF Visua, permet d'avoir un affichage graphique des données renvoyées par le microcontrôleur, il s'utilise avec un programme écrit en C par le Professeur Roskam, pour lire les valeur renvoyées par la maquette sur le port COM.

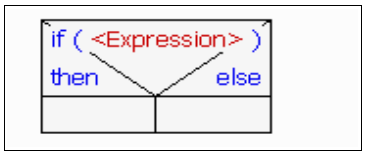
- Easy code: le compilateur C.:

Easy code est un des principaux logiciels sur lequel nous avons travaillé car nous avons tapé tout le code en C grâce à lui.

Il apporte une touche graphique à l'écriture en présentant tout le code C sous forme de blocs.



La compréhension d'un programme en est alors réellement simplifié.
Par exemple la condition if se représente comme ceci :



Cela peut rendre le code aux plus lourds pour de longs programmes mais comme la compréhension en est facilité dans le cadre de programmes courts, il est donc tout a fait approprié pour la programmation de microcontrôleurs.

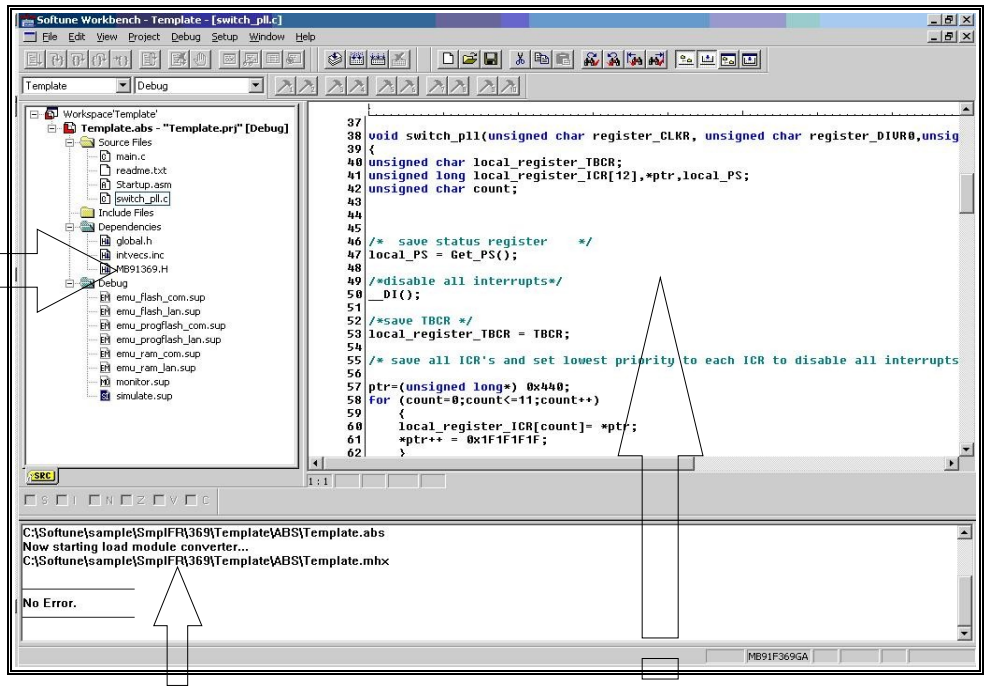
De plus nous avons besoin pour effectuer notre programme de librairies et de fonctions déjà programmées que nous devrons modifier pour faire fonctionner notre montage en les adaptant aux programme que nous allons réaliser.

Et ces fonctions qui ont été écrites par le professeur Roskam, sont beaucoup plus compréhensibles sous ce logiciel grâce à ses représentations graphiques des fonctions, que si le code été écrit en C. pur.

- FR_familie Softune Workbench:

Ce logiciel a été entièrement créé par fújitsu, il est livré avec le microcontrôleur, car il réalise une fonction essentielle pour celui-ci : la traduction du code C en un langage compréhensible pour le microprocesseur: le langage assembleur.
Ce logiciel permet donc de compiler les lignes de code en C, mais il sert aussi de véritable plate-forme pour le groupement des codes: avec d'un côté tous les programmes à extension « .c » et d'autre part toutes les bibliothèques « .h » nécessaires à la compilation d'un programme.
Le logiciel en lui même est assez facile à utiliser une fois que tout le projet est bien mis en place.

Fenêtre d'assemblage des différents modules du projet:
Il faut mettre ici tous les fichiers de librairie(.h), et de code(.c).



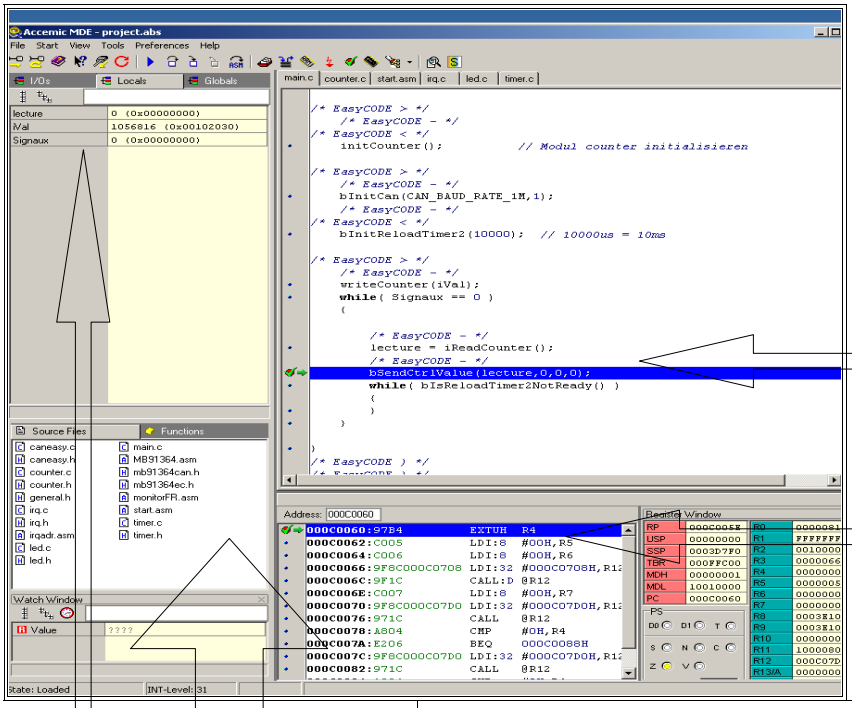
Fenêtre d'état du mode compilation et make, affichage des éventuelles erreurs.

Fenêtre principale :
Elle affiche tous les codes provenant d'easy code et les erreurs dans le script.
Cette fenêtre peut servir d'éditeur de code source, mais son utilisation est beaucoup plus lourde que celle d'easy code, son utilité est donc restreinte à des changements de faible importance, après une compilation quand on a juste fait une petite erreur de syntaxe.

Le code C présent dans Softune est du C pur, il y a juste des annotations servant à Easy code, mais celles si se distingues du code.
Une fois tous les paramètres entrés, nous pouvons assembler tous les codes sources et ainsi créer un fichier make, qui porte l'extension « .abs ».
Ce fichier est prêt à être chargé dans le microcontrolleur cible, par Accemic, puisqu'il s'agit en fait du programme traduit en assembleur.

- Accemic MDE (MDE = Monitor Debugging Environment) :

Accemic est le logiciel qui permet de transférer le code assembleur provenant de Softune directement dans la mémoire du microprocesseur.
Il propose une large gamme d'outils pour interagir avec le microprocesseurs, ce qui permet de tester ses programmes, ou seulement une partie de programme en temps réel.
Il nous a aussi permis de vérifier le bon fonctionnement de tous les composants électroniques présents sur la carte que nous avons réalisée.



L'ensemble de programmes et sous programmant C peuvent être visualisés dans cette fenêtre, on peut aussi définir un point d'arrêt(break point)dans le programme, pour vérifier son bon fonctionnement, ou pour calculer le temps mit par une tâche pour s'effectuer.

Programme en langage assembleur, les programmes peuvent être effectués selon un mode pas à pas, break point, ou en run.

Ici toutes les bibliothèques utilisées.

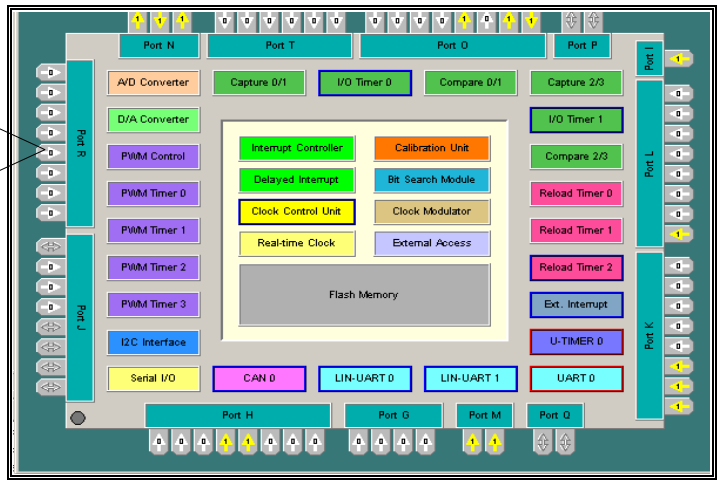
Nous avons ici un affichage de toutes les variables utilisées par chaque programmes, et chaque sous programme du projet.
Nous avons un accès sur ces variables et nous pouvons les modifier pendant que le programme s'exécute.

Accemic permet de sauvegarder un programme dans la mémoire du microcontrôleur, et en étant relié à la carte du microcontrôleur par un port COM, il permet d'interagir avec lui, et donc de tester le comportement du programme et des composants utilisés facilement.
Mais on ne peut en aucun cas y modifier le programme, car celui ci est enregistré dans la mémoire du microprocesseurs.

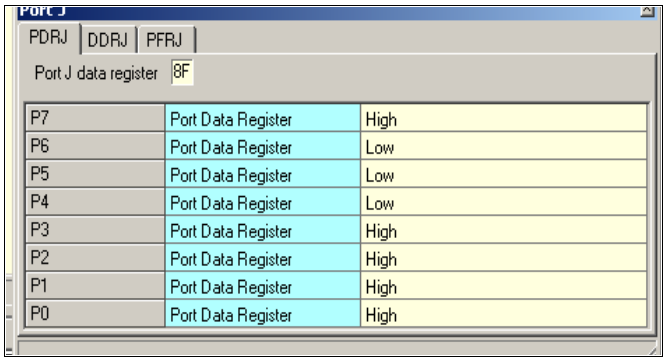
Il faut donc apporter des modifications soit avec Softune soit avec Easy code au programme, traduire le tout en assembleur et le transfert et dans la mémoire du microprocesseur, après chaque modification.

Par contre nous pouvons intervenir sur toutes les broches du microcontrôleur grâce à cette fenêtre :

Ici l'état et la direction de chaque port est affiché.



En cliquant sur un port, on voit apparaître une fenêtre :

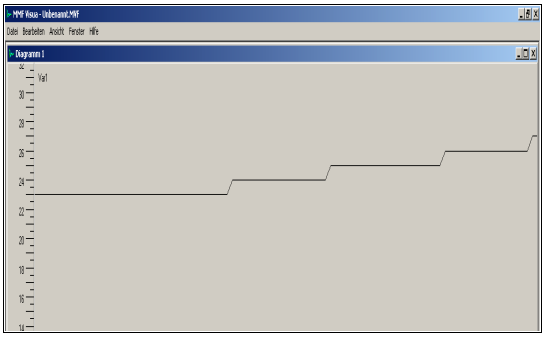


Elle permet de changer en quelques clics, la valeur, la fonction, ou encore la direction de toutes des sorties du port J en l'occurrence.

Cette fenêtre permet aussi d'agir sur quasiment toutes les structures internes du microcontrôleur comme : les registres, les timers et les piles entre autres.

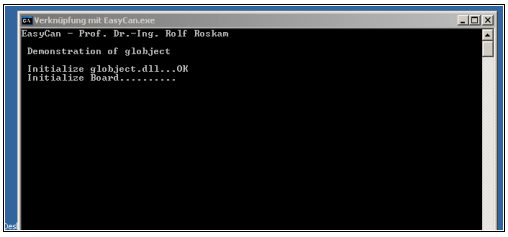
- MMF Visua :

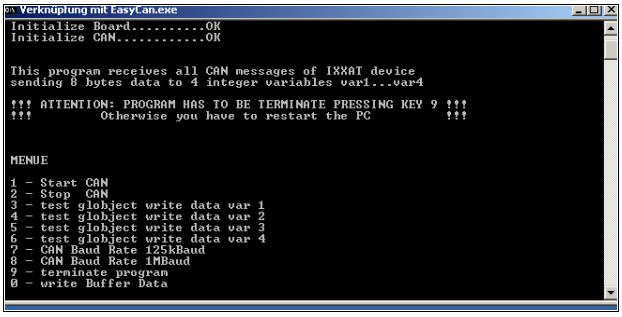
Ce logiciel nous a permis de récupérer les données renvoyées par le microcontrôleur et de les afficher a l'écran sur un graphique.
Pour cela, nous avons dû ajouter de nouvelles fonctions et de nouvelles librairies à notre programme, pour que le microcontrôleur puisse dialoguer avec le PC par le deuxième port COM .



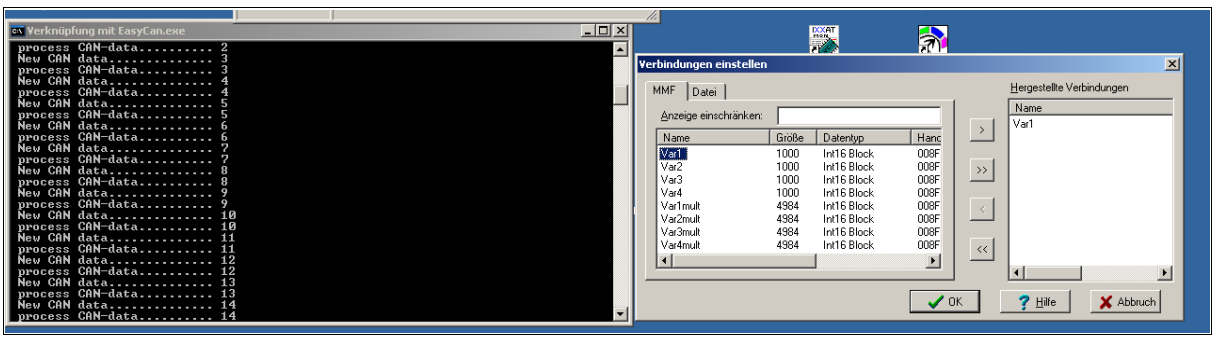
On visualise ici la variable de sortie de notre programme.

Pour acquérir les données dans MMF Visua, nous avons d'abor besoin de lire les valeurs revoyées par notre programme sur le port COM.
Pour ce faire nous avons utilisé un programme écrit en C++ par le professeur Roskam, et qui et compatible avec MMF Visua.
Ce premier programme a été programmé en C++ par monsieur Roskam et permet de lire les données sur le port COM.





Ensuite elles sont envoyées dans MMF Visua qui stocke les valeurs prises par les différentes variables de sortie, et qui affiche le graphe de leurs signaux.



IV/Le projet:

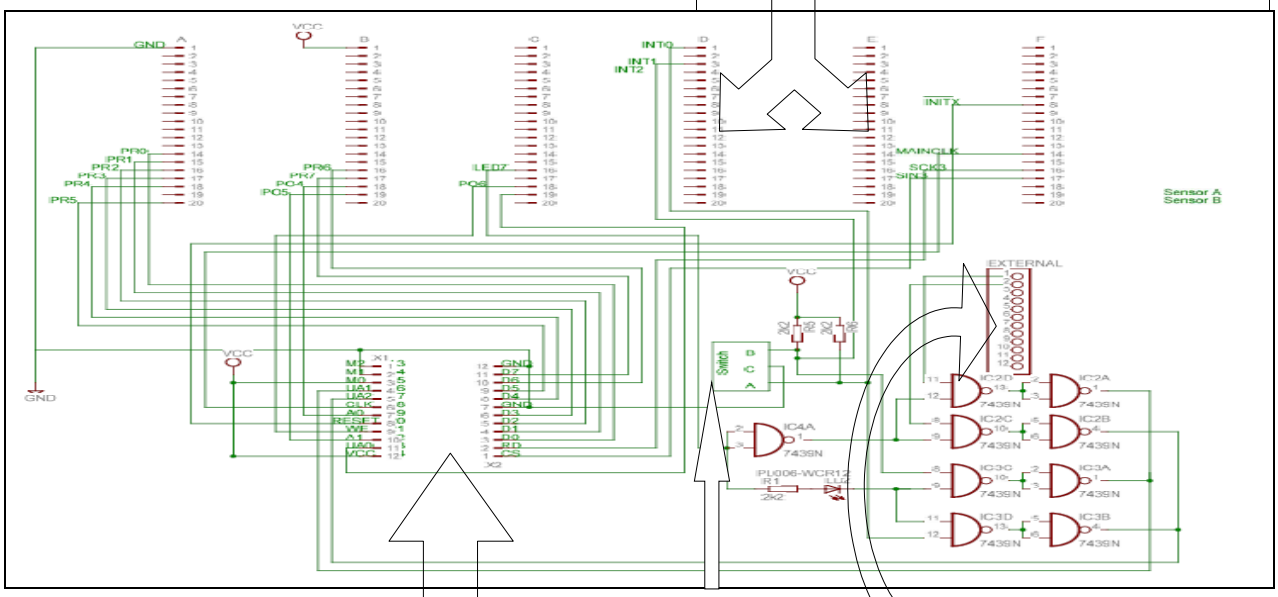
Nous attaquons maintenant le stage en lui même, nous allons parler ici de toutes les démarches que nous avons effectuées dans les laboratoires de Mr. Roskam pour remplir le cahier des charges qu'il nous a expliqué.
Nous traiterons du stage dans l'ordre chronologique c'est-à-dire en commençant par expliquer la conception de la carte électronique, puis sa fabrication, ensuite nous présenterons les programmes que nous avons réalisés dans la partie programmation du stage.

- Conception de la carte électronique :

Pour commencer, nous avons dû réaliser plusieurs schémas électroniques.
Nous avons conçu le premier schéma, juste après que M. Roskam nous ait expliqué les tâches à faire accomplir par notre carte. Mais il y a eu des changements pour les branchements du compteur après avoir montré notre travail à notre tuteur.

Mais dans un premier temps, nous allons expliquer le fonctionnement de notre premier schéma et les problèmes que celui-ci posait.

Ces 6 broches de vingt emplacements sont celles de la starter board du microcontrôleur numérotés de A1 à F20.



Le compteur PE12024

l'encodeur incrémental série E27
Il peut générer des signaux carrés et sert aussi de bouton poussoir.

Bornier d'entrée de la carte, on reçois ici les signaux A, B et RESET, provenant des capteurs.

Afin de faire marcher le compteur, nous devons tout d´abord récupérer les signaux de déplacement A et B provenant de deux sources différentes :

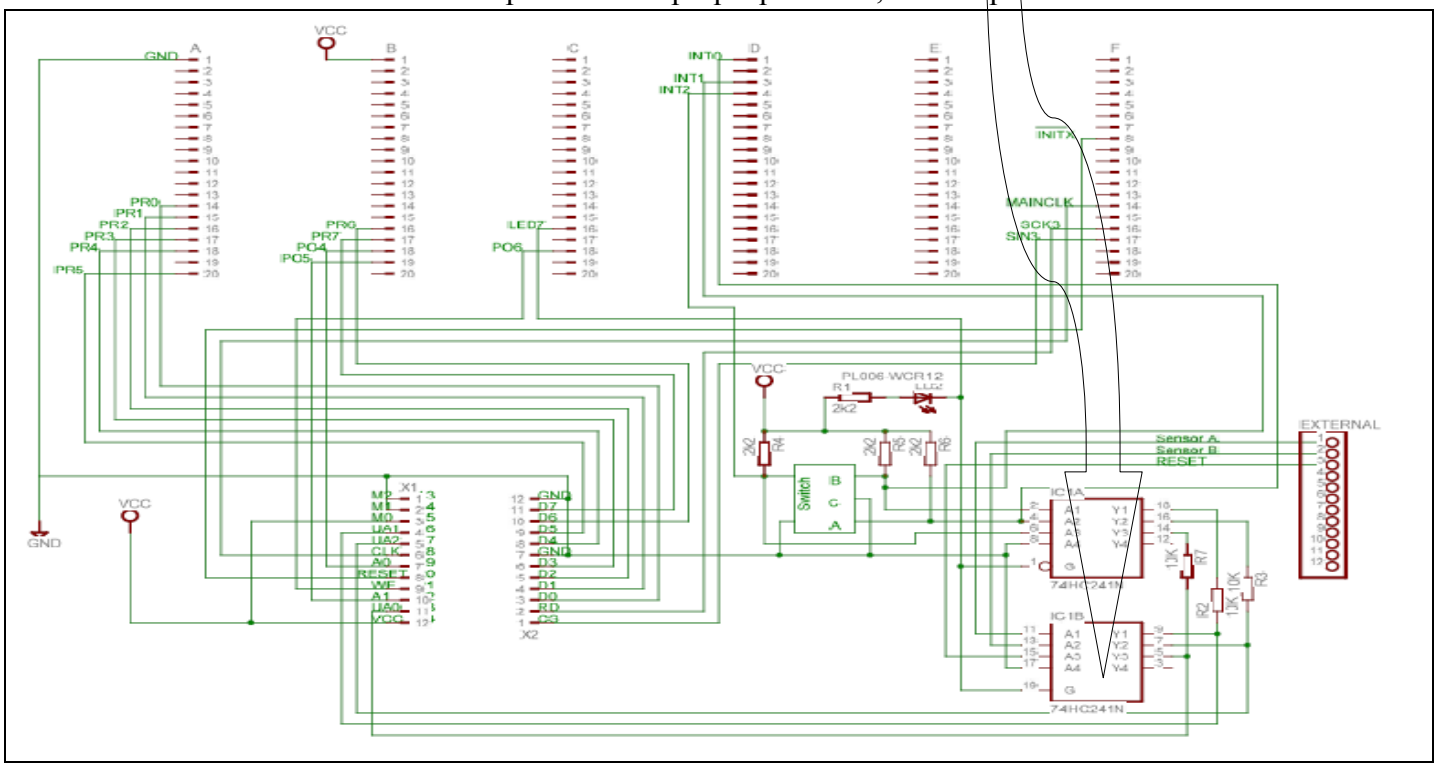
- La première source est l'encodeur incrémental série E27 qui permet de générer deux signaux carrés, avec un léger décalage entre eux, et qui nous a permis de tester dans un premier temps notre programme.

- La deuxième provient des deux capteurs externes à la carte qui génère aussi des signaux carrées sur les sorties en destination de UA1 et UA2 du compteur.

Le montage avec les neuf portes nand permet donc de choisir entre un de ces deux signaux, en activant seulement une des deux paire de signaux A et B.

Comme vous pouvez le constater, sur le schéma il y a plusieurs portes NAND, exactement neuf portes.
Le problème réside alors dans l'espace occupé par toutes ces portes logiques. En effet, pour avoir nos neufs portes NAND, il nous fallait trois circuits de quatre portes logiques chacun, car l'université ne dispose pas d'autres modèles de porte NAND.
Il ne resterai alors pas assez de place pour positionner tous les composants sur la future maquette.
Le professeur Roskam nous a donc soumis sa proposition:
Le remplacement des portes NAND par un seul circuit : Le CD74HCT241E :

Le fonctionnement de ce composant est expliqué plus haut, dans la présentation du matériel.



Explication du montage :

Pour faire le choix entre les dispositifs de comptage, il se trouve que notre nouveau circuit dispose de deux entrées; une complémentée par rapport à l'autre(voir schéma, broche. 1 et 19 du circuit logique CD74HCT241E).
On peut voir à partir de quel dispositif le comptage ce fait grâce à une D.E.L.
Elle s'allume lorsque le comptage ce fait par le biais de l'encodeur incrémental série E27.

Cette L.E.D n'est en fait que le reflet de l'état de la sortie du microcontrôleur que nous utilisons pour choisir la source de notre comptage.

Explication des connections autour du microcontrôleur:

Dans notre microcontrôleur, il y a plusieurs ports, dont le Port R qui nous sert à rentrer des données sur le compteur ou à récupérer la donnée; nous avons pris ce port car il est le seul à disposer de 8 entrées: ce qui facilite la tâche plus tard dans la programmation.

Il a aussi le Port O, il est utilisé pour le choix des octets lors de la lecture ou de l'écriture et aussi pour activer ou désactiver l'entrée /WR du compteur.

Ensuite nous avons le Port N qui active ou désactive les entrées /CS et /RD du compteur .

Nous utilisons aussi le Port J dédié aux leds, ici on l'utilise pour voir par quel dispositif on compte.

Pour finir on utilise le Port K. Ce port est spécialement dédié aux interruptions.

Connections des broches du compteur:

Les sorties D0-D7 sont reliées au Port R du microcontrôleur.

Les entrées M0, M1 et M2 nous permettent de choisir un mode de fonctionnement du compteur, il y en tout six mode de fonctionnements pour le comptage.

Nous avons choisi le premier mode: le compteur est synchronisé sur les fronts montants de UA1.

Les entrées UA1 et UA2 servent pour recevoir les signaux A et B.

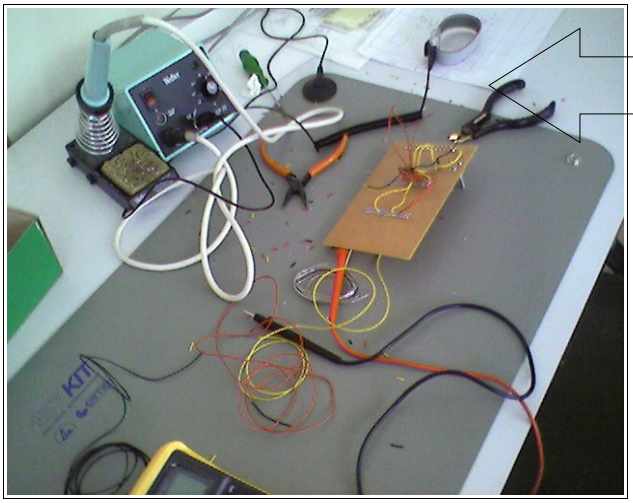
UA0 sert pour le reset du compteur(attention cette entrée est la seule des remises a zéro qui n'est pas complémentée).

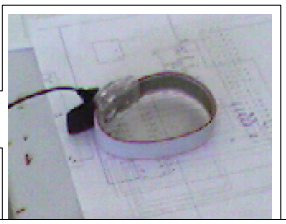
On récupère l'horloge à partir de l'entrée „CLK“ que l'on récupère à la sortie du microcontrôleur „MAINCLK“(broche F14 du schéma).

Et puis il faut surtout pas oublier l'alimentation et la masse du compteur.

- Réalisation de la carte électronique

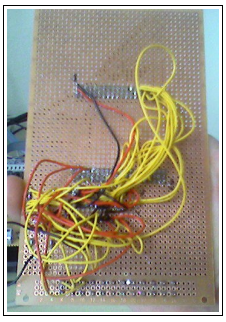
Voici donc ci dessus le poste de soudure avec lequel nous avons réalisé notre carte électronique.



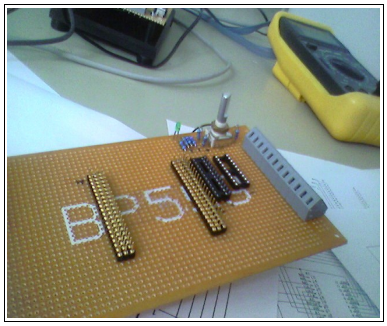


Ce petit bracelet permet de manipuler des composants électroniques sans risquer de les abîmer car il décharge le corps humain de son électricité statique en le mettant à la masse.

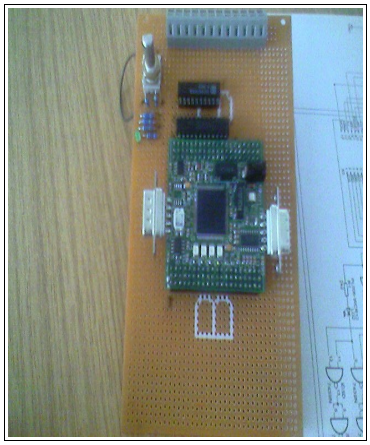
Et ci dessous quelques photos montrant sa progression:



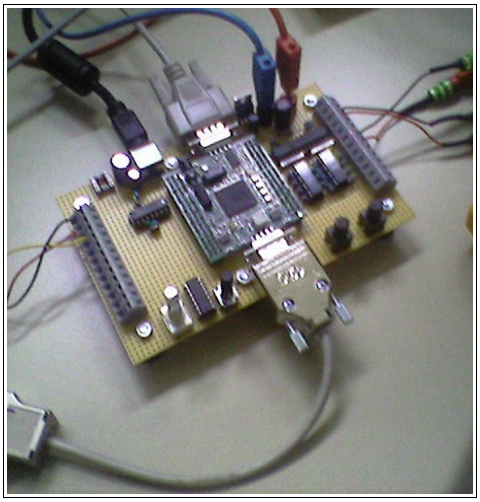
Vue de dos.



De face sans les circuits imprimés.

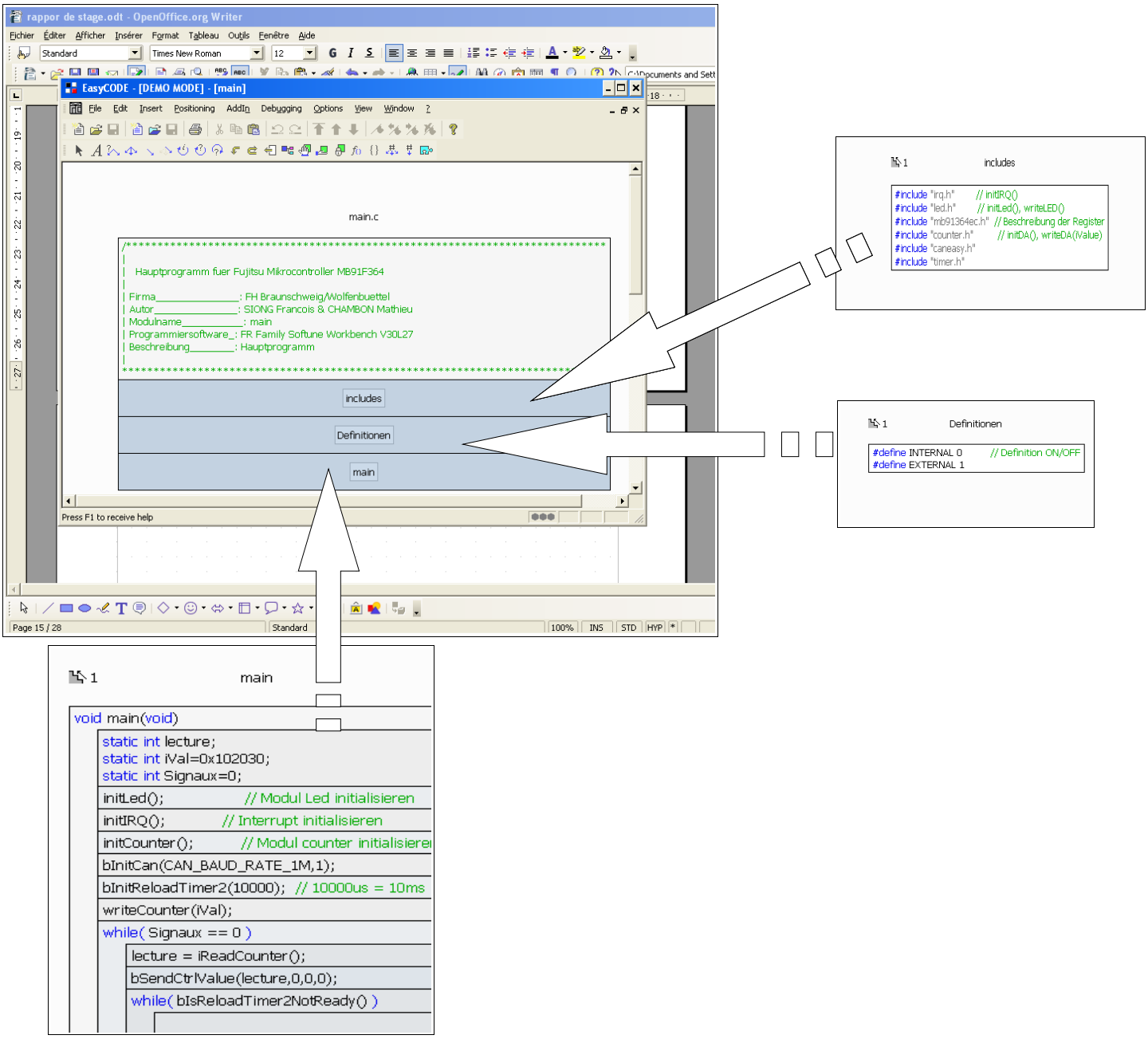


Enfin la carte finie:



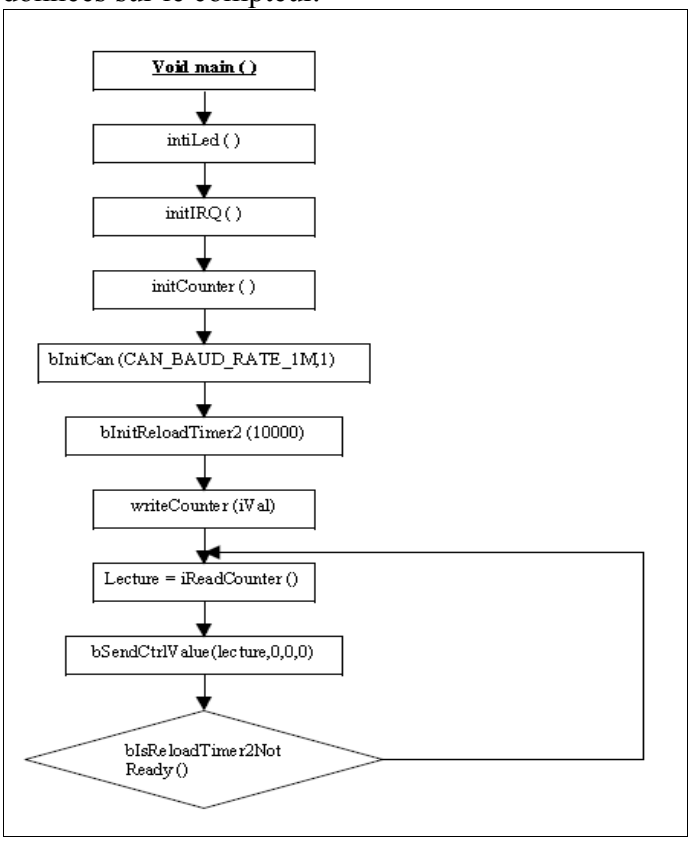
Et voilà un premier essai de la carte sur laquelle notre projet va être implanté.

- Programmation du microcontrôleur
  - **Les différents programmes**
    - Le programme principal : main ():



Le programme « void main ( ) » est l'endroit où nos sous-programmes vont être appelés, c'est le programme principal de notre projet.

Donc dés le début du programme principal, il y beaucoup d'initialisation tel les « led », les interruptions « IRQ », le compteur 24-bits et le port « CAN » qui nous permet de relever les données sur le compteur.



- ✔ InitLed () : cela permet de faire une initialisation de toutes les led , ici nous les éteignions.
- ✔ InitIRQ () : permet d'initialiser toutes les interruptions pour ne pas dérégler la mise en marche du programme entier lors de la mise sous-tension de toute la carte électronique.
- ✔ InitCounter () : Cette fonction initialise le compteur, et est décrite dans Counter.c.
- ✔ InitCan () : initialise le bus « CAN » pour avoir une bonne fréquence d'émission des données, pour pouvoir communiquer avec l'ordinateur.
- ✔ bInitReloadTimer2 (10000) : permet de définir la temporisation ReloadTimer2 à 10ms.
- ✔ writeCounter (iVal) : on appel le sous-programme pour écrire une valeur « iVal » sur le compteur cette fonction est aussi écrite dans Conter.c.
- ✔ Lecture = iReadCounter( ) : appel du sous-programme de lecture, qui renvoi la variable « lecture », qui est la valeur lue sur le compteur.
- ✔ bSendCtrlValue(lecture,0,0,0) : Cette fonction permet d'envoyer la donnée lue sur le compteur dans le programme extérieur pour l'afficher dans un graphe en fonction du temps.
- ✔ **La condition à la fin du programme vérifie bien si les 10ms de « ReloadTimer2 » se sont bien écoulées pour pouvoir refaire une lecture, et ainsi de suite, cela permet de réaliser une attente, pour que le port COM ne soit pas submergé de données.**

- [Le programme du compteur Counter.c]

Counter.c

```
/*****************************************************************************
|
| Firma_____________: FH Braunschweig/Wolfenbuettel
| Autor_____________: SIONG Francois und CHAMBON Mathieu
| Modulname_________: Counter
| Programmiersoftware_: FR Family Softune Workbench V30L27
| Beschreibung______: Counter fonctions
|
*****************************************************************************/
includes
void initCounter(void)
void writeCounter(int Value)
void SetSwitch(int iValue)
int iReadCounter(void)
```

1 includes

```
#include "mb91364ec.h"      /* Zugriff auf CPU-Register*/
#include "counter.h"
```

Voilà les bibliothèques utilisées par notre programme, la première est celle du microcontrôleur, la seconde est celle de notre programme, et nous y avons déclaré toutes les fonctions que nous avons créés.

Nous allons maintenant détailler toutes les fonctions que nous avons programmées dans ce fichier:

- [La fonction InitCounter();]

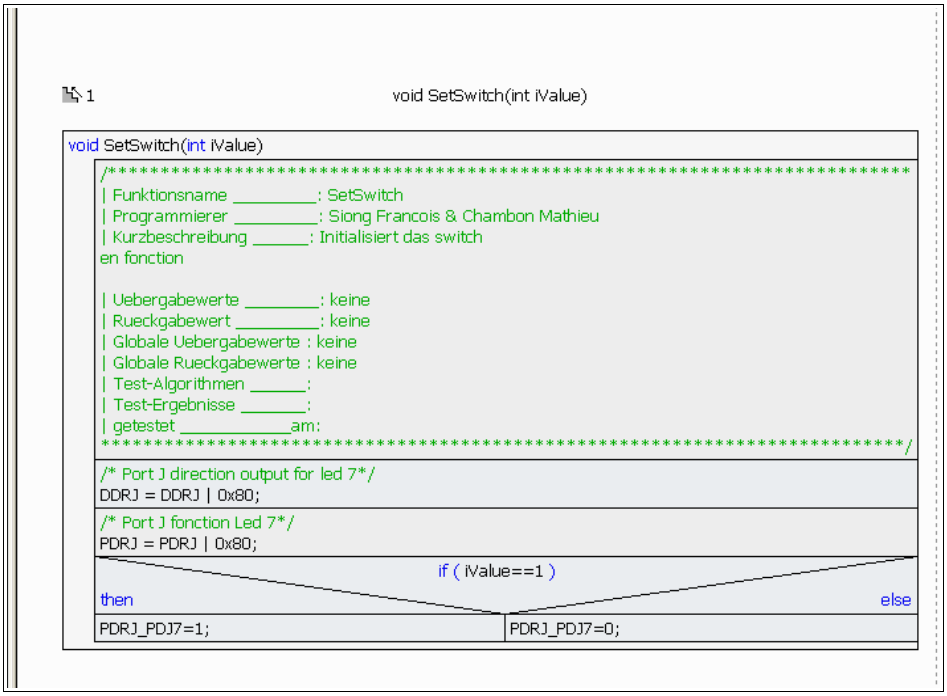
Elle permet d'initialiser le compteur, en définissant la direction des ports O,N et J qui permettent l'accès au compteur.
Ensuite, on appelle la fonction writeCounter, pour écrire une valeur d'origine dans la mémoire du compteur. Ici nous initialisons à 0x00.

1 void initCounter(void)

```
void initCounter(void)
{
  /*****************************************************************************
  | Funktionsname _________: InitCounter
  | Programmierer _________: Siong Francois & Chambon Mathieu
  | Kurzbeschreibung ______: Initialisiert das  Counter
  |                          Port PFRJ (Port Function Register)
  |                                  Bit 0 bis 3 auf 1: LED Function
  |                          LED ausschalten, ueber Port PDRJ (Port Data Register)
  |                                  Bit 0 bis 3 auf 1: LED aus
  | Uebergabewerte ________: keine
  | Rueckgabewert _________: keine
  | Globale Uebergabewerte : keine
  | Globale Rueckgabewerte : keine
  | Test-Algorithmen ______:
  | Test-Ergebnisse _______:
  | getestet _____________am:
  *****************************************************************************/
  /* Port O direction output for P06,P05,P04*/
  DDRO = DDRO | 0x70;
  /* Port J direction output for led 7 :out*/
  DDRJ = DDRJ | 0x80;
  /* Port J fonction Led 7*/
  PFRJ = PFRJ | 0x80;
  /* Port N direction output PN5 et PN3 : out*/
  DDRN = DDRN | 0x28;
  writeCounter(0x00);
  SetSwitch(0x00);
}
```

- La fonction SetSwitch (iValue),

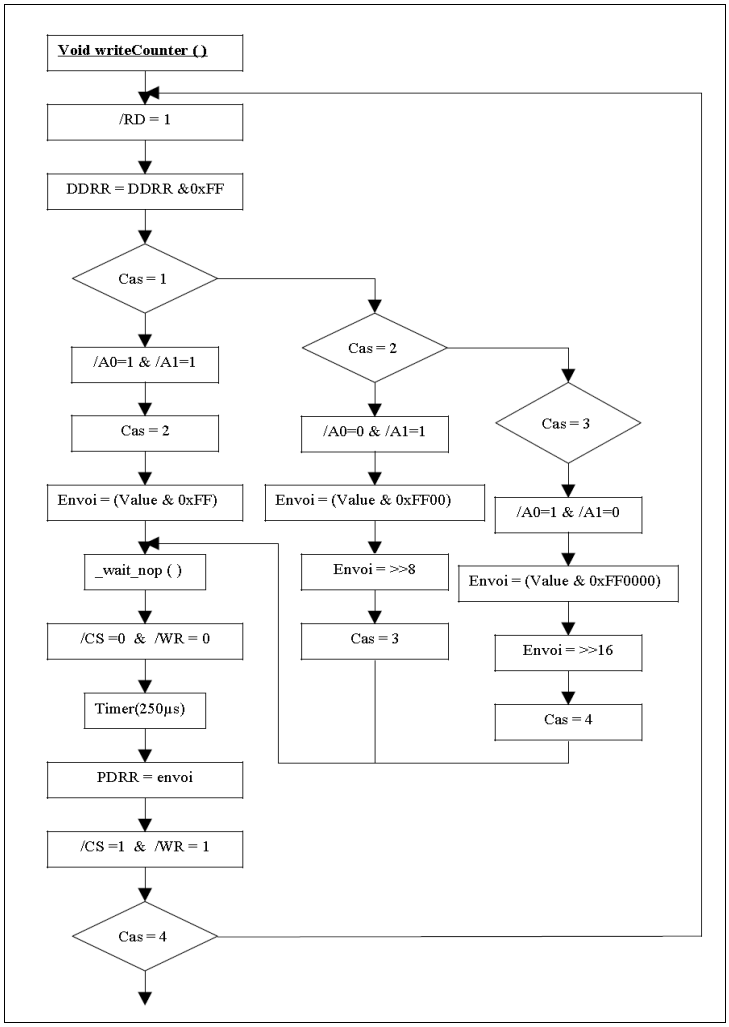
Cette fonction nous permet de choisir quels signaux d'entrée nous voulons traiter.
Selon la valeur de la variable iValue, une des deux paires de signaux se retrouve en entrée du compteur.



L'état du port J7 dépend donc de la valeur de iValue, et la diode électroluminescence indique l'état de J7.

- La fonction writeCounter();

Ce sous-programme, comme son nom nous l'indique, permet de faire une écriture sur le compteur. A quoi servira-t-il ? Et bien dans notre projet, cela faisait parti de notre cahier des charges. Nous avons eu la fonctionnalité de ce sous programme de la part de notre tuteur : que ce sous-programme va lui permettre, à l'aide d'un vérin hydraulique le, de positionner une plate forme en envoyant une donnée sur le compteur.



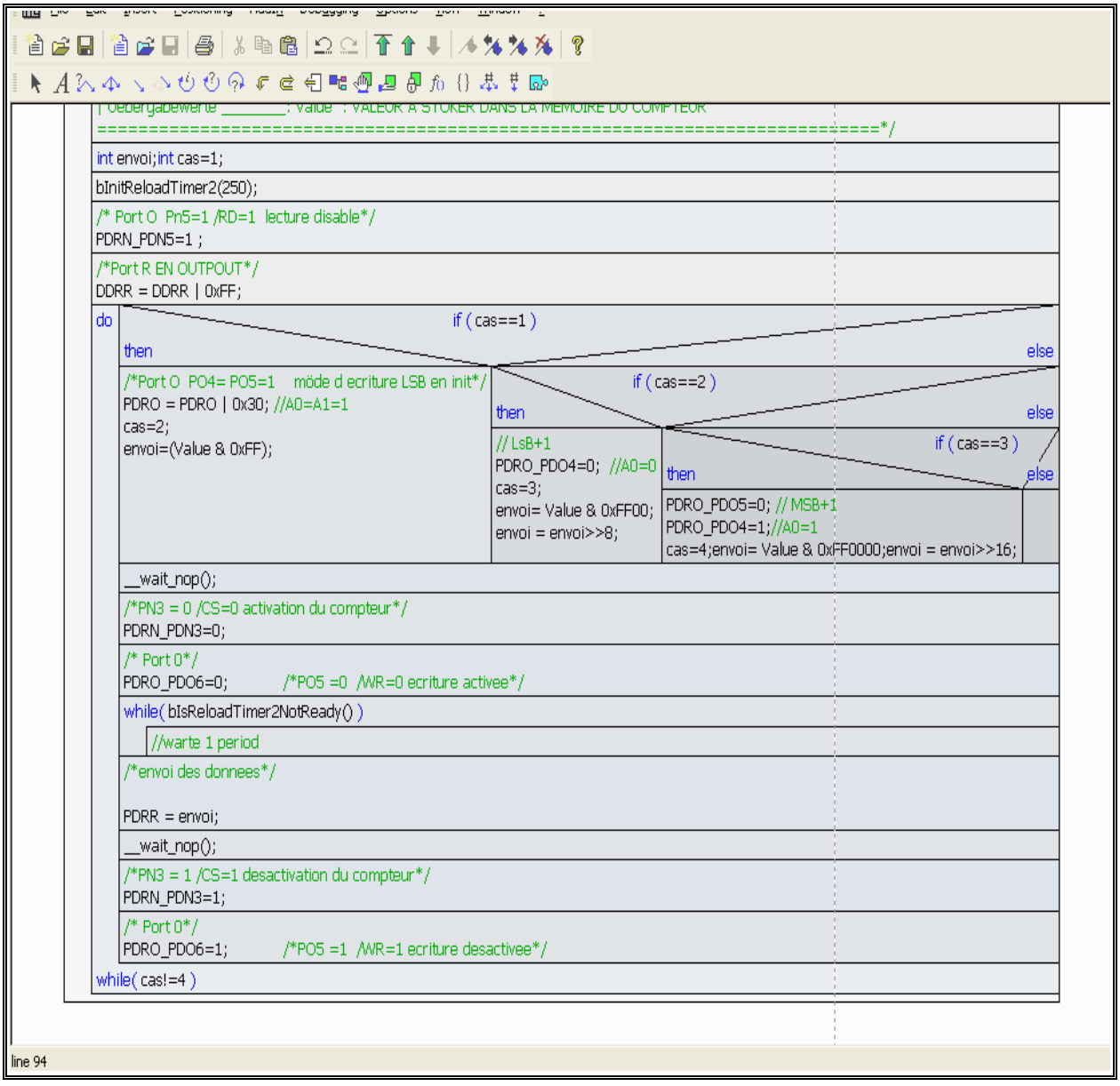
- ✔ Tout d'abord on met l'entrée /RD à l'état « 1 » car cela permet de désactiver le mode lecture du compteur.
- ✔ On met les entrées /A0 = 1 et /A1 = 1, on choisi le premier octet à envoyer le LSB.
- ✔ Ensuite on règle les broches port R en tant que sortie, de ce fait nous pourrions envoyer le chiffre prédéfini sur le compteur ou une valeur quelle conque.
- ✔ Dès le début de notre sous-programme, notre variable « Cas » est égale à 1. Ceci permet d'avoir une boucle tant que « Cas » n'est pas égale à 4. De ce fait nous envoyons une donnée sur le compteur de 3 octets.
- ✔ Nous procédons ensuite à un masque « Value & 0xFF » qui nous permet de prendre que en compte les donnée du premier octet. On stock cette valeur dans la variable « Envoi », on récupère cette valeur plus tard.

- ✔ Nous attendons _wait_nop() qui permet de faire une temporisation de 16,5µs, nous sommes obligés car c'est le fonctionnement de l'écriture sur le compteur.
  En effet nous devons effectuer les actions sur le compteur en respectant des temps d 'attente, car notre microcontrôleur travaille a une vitesse telle qu'il réagit plus vite que le composant.

- ✔ Nous mettons /CS = 0 et /WR = 0 pour les activer. Cela va nous permettre d'écrire sur le compteur. Après cela nous faisons une temporisation de 250µs avant l'envoi de la donnée.

- ✔ Après les 250µs, nous procédons à l'envoi de la donnée sur le compteur par la fonction « PDRR = envoi ». Notons ici que nous faisons appel à la variable précédemment utilisé « Envoi ». Tout juste après l'envoie, on remet /CS et /RD à l'état inactif « 1 ».

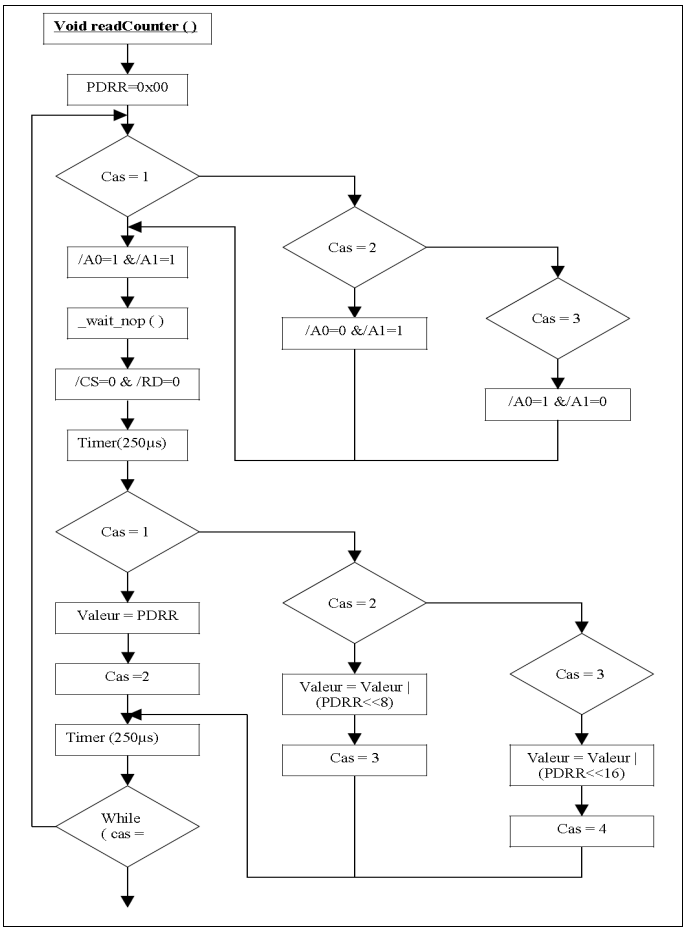
- ✔ On peut voir dans l'algorithme que dans le « Cas = 1 » que cette variable passe à 2. Ceci nous permet, à la toute dernière condition, d'envoyer le 2e octet et ensuite le 3e. Pour enfin sortir de cette boucle, la variable « cas » doit être égale à 4, c'est ce qu'on trouve enfin du « Cas = 3 ».



- La fonction iReadCounter();

Donc voici l'algorithme du sous programme iReadCounter ().

Tout ce sous programme permet de lire les différentes données dans le compteur. Cette lecture ce fait en temps réel et nous permet de voir l'évolution du compteur.



Le fonctionnement de ce sous-programme :

- ✔ La première chose à faire est de choisir la nature des broches du portR. Pour ce faire nous avons le choix entre les forcer à être des sorties ou des entrées. Dans ce cas où nous sommes dans la lecture de donnée, nous allons placer ces broches en tant que « entrée ». Ceci permet de faire rentrer les données du compteur dans notre microcontrôleur et de les traiter.

- ✔ La deuxième étape consiste à choisir qu'elle octet choisir à lire sur le compteur. Nous sommes en présence d'un compteur de 24-bits et de 8 entrées (ou sorties) des données. Donc il ne permet pas de lire directement les 24-bits d'un seul coup. Pour cela, le compteur est muni de 2 entrées qui nous permettent de lire chaque octet du compteur. L'entrée /A0 et /A1, lorsque /A0=1et /A1 =1, nous lisons le premier octet du compteur. Lorsque /A0=0 et /A1=1, c'est la lecture du deuxième octet. Puis quand /A0=1 et /A1=0, nous lisons le dernier octet du compteur. Cet ordre de lecture doit être respecté.

- ✔ Lorsque les cas choisis, nous mettons une opération qui se nomme _wait_nop ( ). Ceci nous permet de faire une temporisation de 16,5µs. Ceci est nécessaire car les entrées /A0 et /A1 ont une priorité de mise à niveau avant les entrées /CS et /RD.

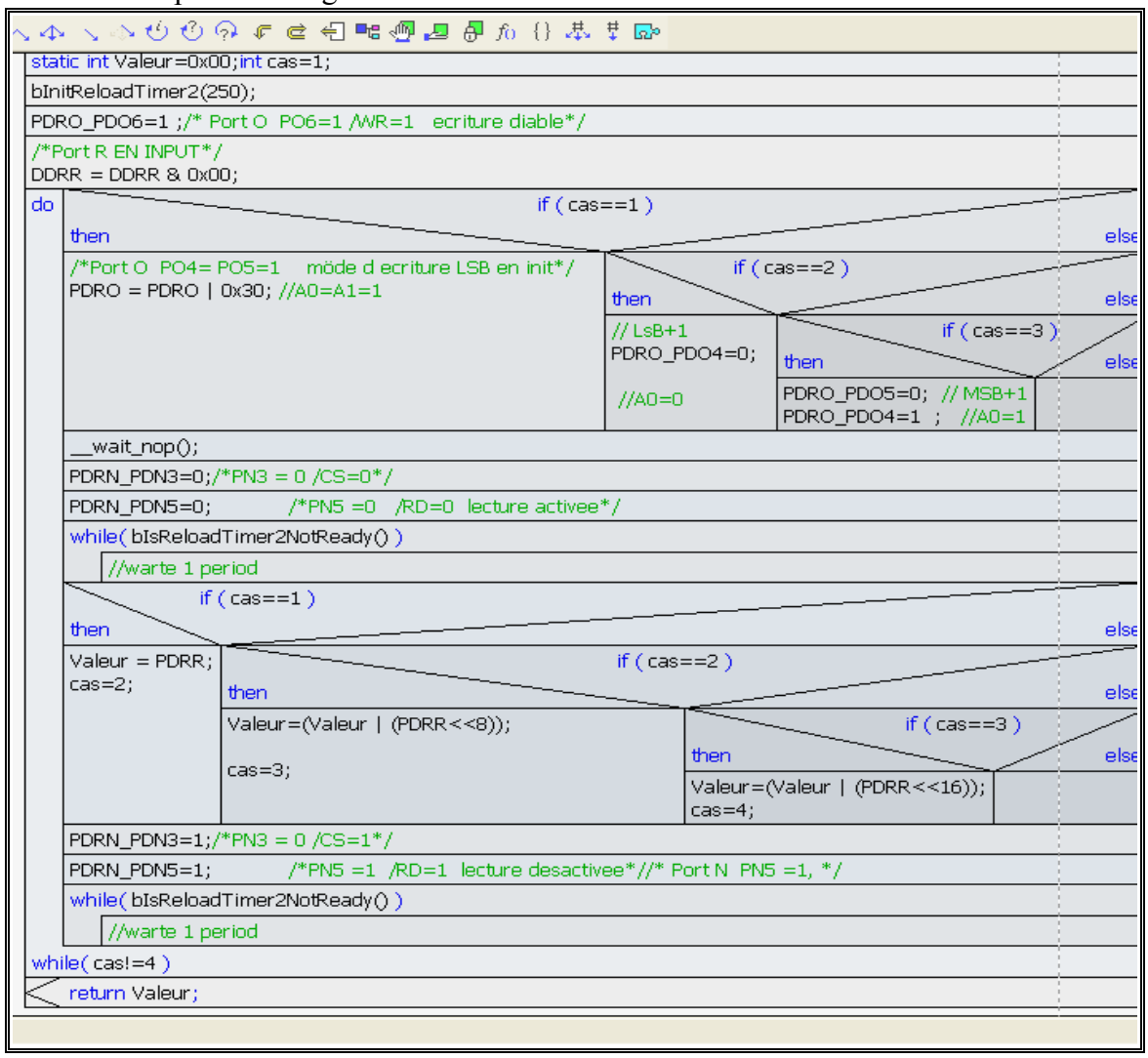
- ✔ La suite est logique, nous mettons /CS et /RD à l'état 0 pour les faire fonctionner car ils sont complémentées à leur entrée. /CS permet de choisir (ou de activer la puce) et /RD permet de mettre le mode lecture. Puis cas =2 nous permet de faire avancer dans l'algorithme.

- ✔ Le timer vient juste après l'activation de /CS et /RD car on doit attendre que les données du compteur se stabilise car sinon nous aurons des données faussées due à une instabilités des sorties du compteur (c'est-à-dire lors des changement d'état des sorties lorsque notre compteur incrémente ou décrémente). A la fin de cette première lecture, nous avons rebouclé juste après la mise en état de sortie du port R car nous avons 24-bits à lire. Or « Cas = 2 », nous passons dans une autre branche autre que celui emprunté dans la première lecture. Cette fois-ci « /A0=0 et /A1=1 ». L'état dans laquelle sont entrées nous permet de lire le deuxième octet du compteur.

- ✔ Pour faire une lecture du deuxième octet, nous devons faire un décalage de la donnée lue, car nous sommes sur la position du 1er octet. Pour ne pas effacer la première lecture effectuée, le décalage ce fera de la manière suivante :

  « Valeur = Valeur | (PDRR<<8)

- ✔ Et puis ainsi de suite, nous rebouclons vers la début. Ce qui change ici est le décalage, elle ne se fera pas 8 bits à gauche mais de 16 bits.



NOTE : Tant que « la variable Cas est différente de 4 » nous continuerons à faire la boucle afin de lire toute la donnée de 24-bits.

➢ Les problèmes rencontrés

- ✔ Le tout premier problème était la compréhension des logiciels. En effet, afin de comprendre les logiciels, ils nous fallait modifier des programme existants et voir les changement à effectuer.

- ✔ Le deuxième problème était la compréhension des documentations des composants. Seule la lecture ne suffisait pas à comprendre le bon fonctionnement de chaque composants. Pour remédier à ce problème nous avons fait des schéma électronique et proposé à notre tuteur des solutions. Après validation du schéma final, nous avons commençé les soudures des composants sur la maquette mise à notre disposition.

- ✔ Le plus gros problème du projet, c'est de ne mélanger les fils et leurs emplacements car une seule erreur serait fatale pour notre microcontrôleur. Mais pour ne pas détruire notre microcontrôleur, après les soudures, nous avons fait les tests pour chacun des fils.Il se trouve que après cette étape, qu'il y avait des erreurs en effet. Ces erreurs étaient surtout des erreurs d'emplacements ou d'inversion entre les fils. Enfin de compte au bout d'une semaine de soudure et les tests, nous avons réussit les soudures sans aucun mélange de fils ni des faux contacts entres les pattes des composants.

- ✔ Après les soudures, les problèmes étaient toujours présents. En effet, après cette semaine de soudure, nous avons attaqué la programmation de notre microcontrôleur . Après avoir fait ces algorithmes, nous sommes passé à la programmation sur logiciel. Puis les tests sur l'ordinateur nous ont aidé énormément. Mais malheureusement, il avait toujours un problème dans la programmation. Ce problème persistait sur l'incrémentation du compteur du premier octet. Ce qui veut dire en d'autre terme que le compteur nous affichais la valeur précédente et non la valeur au moment de la lecture elle-même, ce qui représentait un énorme problème car notre carte permet le positionnement d'une plate forme. Au bout de quelques jours le problème était réparé et le compteur marchait à la perfection sans aucun affichage des valeurs précédentes lors des tests.

Sans la résolution des ces problèmes, notre projet n'aurait pas été achevé convenablement.

V/Le site internet:

Pour présenter la ville de Wolfenbüttel aux étudiants qui passeront après nous, nous avons décidés de réaliser un petit site Internet.
Notre idée était de créer des pages Web décrivant l'atmosphère de la ville qui nous a accueilli.
Mais nous voulions aussi que ce site se disponible en plusieurs langages car l'université allemande s'est aussi intéressée à notre travail.
Le résultat final sera donc un site Internet avec des photos et la possibilité de choisir entre trois langues : le français, l'allemand, et l'anglais.

Architecture du site :

- accueil : page d'accueil
- la ville : présentation de la ville
- l'université: présentation d'université
- les habitants: présentation des Allemands
- les soirées: petit aperçu des soirées allemandes
- les auteurs: information sur les auteurs
    - Mise en ligne du rapport de stage.
    - Liens vers les sites web des deux universités

Pour le design de ce site nous avons utilisé le modèle d'un site Web déjà construit durant les cours à l'IUT.
Et nous avons réadapté son architecture pour qu'il puisse être disponible en trois langues.
Ensuite nous avons ajouté une bannière qui donne la météo du jour de Hanovre, nous n'avons pas pu avoir celle de Wolfenbüttel car les sites allemands de météo sont payants. Mais nous avons remarqué que les informations données pour Hanovre correspondent souvent avec le temps de Wolfenbüttel.

Nous nous sommes ensuite attachés à ajouter les photos que nous avions prises depuis le début de notre séjour sur le site, en les commentant.
Il a ensuite fallu traduire toutes nos rubriques en anglais et en allemand.

Une fois tout ceci fait nous avons corrigé les derniers bugs et nous avons cherché un serveur pour héberger notre site.
Mettre le site en ligne nous a pris énormément de temps par rapport au temps que nous aurions dû y passer, car les plates formes de mise en ligne gratuites sont volontairement plus difficiles à utiliser que celles payantes.

VI/Conclusions:

- Conclusion générale:

Ce stage en Allemagne nous aura permis de découvrir de nombreuses choses :
L'autonomie pendant trois mois dans un pays étranger, les méthodes de travail des Allemands, en plus de toutes les compétences techniques que nous avons dues développer pour mener à bien notre projet.

Le projet que nous avons eu été très intéressant et enrichissant.

Partir de la base de la conception d'une carte nous montre les difficultés que cela peut parfois entraîner.

Comme les différentes manière à concevoir notre schéma: nous avons eu deux schéma possible; la notre avec l'utilisation des portes logiques N-AND et celui avec le CD74HCT241E.

Ou encore les soudures qui fût un temps très difficile pour nous car si les emplacements étaient mis au mauvais endroit il fallait chercher l'erreur avec un multimètre.

Mais avec persévérance nous avons abouti a la fin de notre projet dans le temps imparti de notre période de stage.

Ce stage nous a permis d'avoir une vision de ce qu'est la conception de cartes électronique. Nous avons vu les différents chemins qui aboutissent au résultat final. Tel que par le nombre possible de choix des composants à choisir car il y a une centaine de possibilités voir plus pour faire notre carte.

La motivation était présente car notre carte été en réalité une partie de la maquette de notre tuteur. Et qu'il fallait le terminer avant notre retour en France.

Pour finir cette partie, on voudrais dire que nous avons vécu de bons moments durant ce séjour. On souhaite bon courage aux futur étudiants qui passeront après nous.

- Conclusions personnelles:

SIONG François

Tout d'abord le projet que nous a mis en main m'a beaucoup plu. En effet, les missions que l'on devait accomplir étaient très intéressant. D'une part la recherche du montage a été un dur passage car même si je suis bilingue, la documentation des composants n'est pas toujours très compréhensible. Mais on a tenu bon et avons respecté le cahier des charges à la lettre.

Mais le moment le plus difficile est pendant la soudure des composants car une fois tout soudé, on ne savait si les connections étaient bien à leur place respective. C'est pour cela que nous avons tout dessoudé et recommencé de nouveau. Enfin de compte la deuxième fois fût la bonne car il n'y avait pas d'erreurs dans les connections.

Ce fût une expérience formidable car nous étions confrontés à nous même. En effet, notre tuteur était rarement derrière nous donc il a fallut fournir beaucoup plus de travail d'équipe pour mener à bien notre projet.

Cette expérience n'est pas seulement professionnelle mais aussi culturelle. Nous avons vécu en Wolfenbüttel pendant deux mois et demi et nous avons pu voir comment les gens vivaient au quotidien. De plus, dès notre arrivé, nous avons rencontré les colocataires de Morgan et Mathieu. Ce sont des Finlandais ce qui permet des échanges entre nous.

Pour finir cette partie, la seule langue qui permet de communiquer entre nous les étudiants était l'anglais.

CHAMBON Mathieu : Partir pendant trois mois en Allemagne m'a bien sûr permis de faire progresser mon niveau en allemand, mais aussi en anglais car je me trouvais dans une collocation où il n'y avait qu'un seul allemand.
La langue la plus utilisée entre nous était alors inévitablement l'anglais.$ La seule chose pour laquelle je n'ai pas progressé est mon inébranlable accent français, que je n'arrive pas à perdre.

J'ai aussi découvert une partie d'un pays que je ne connais que trop mal, et j'ai découvert l'univers de l'éducation allemande.

Mener à bien notre projet n'a pas été une chose facile, mais nous sommes parvenus a remplir le cahier des charges qui nous a été fournis, en mettant en place les compétences développées pendant nos deux années d 'études à l'IUT de Nîmes.

Je retiens donc de ce séjour un agréable souvenir, grâce a toutes les personnes qui on su nous guider tout au long de ce stage.

VII/Remerciements :

Nous tenons à remercier toutes les personnes grâce à qui se stage en Allemagne à été possible :

Tout d'abord un grand merci au professeur Roskam, notre tuteur de stage, qui nous a accueilli dans son laboratoire, guidé et nous aider lorsque cela était nécessaire.

Un merci pour M. Zemmiri qui était la pour nous aider dans les premiers pas á l'université.
Nous voudrions, de manière générale, remercier l'ensemble de la Fachhochschule pour le travail accompli dans l'accueil des étudiants étrangers et en particulier le bureau des relations Internationale Mlle Katrin Müller et Mlle Kristina Rübenkamp.
Enfin remercier tous ceux qui nous ont permit de rendre ce stage en Allemagne une expérience mémorable.

Nous tenons aussi à remercier certaines personne en France qui ont amplement contribué à notre réussite, nous pensions notamment à Mlle Julien qui nous a fourni les contacts et aussi l'ensemble du Bureau des Relations Internationales.

IX/ANNEXES :

Les fichiers d'annexes ci dessous sont présents sur un CD car ils sont trop volumineux sur support papier..

- ✔ Datasheet CD74HCT241E
- ✔ Datasheet PE12016_24
- ✔ Datasheet MB91F364G
- ✔ EvalBoard MB91F364G

Sources

Les principales sources de notre étude et de mon rapport

- ✔ http://www.accemic.com
- ✔ http://www.fme.fujitsu.com
- ✔ http://www.easycode.com

- ✔ Documentation techniques présentent en annexe de ce rapport