

Projet e-kart 50m départ arrêté

Projet e-kart 50m départ arrêté

Remerciements

Je remercie messieurs Thierry LEQUEU, Daniel ALQUIER et tout le personnel de l'IUT m'y avoir accueilli.

Je remercie tout particulièrement mon maître de projet monsieur Thierry LEQUEU de m'avoir accordé un peu de son précieux temps, de ses idées, et connaissances.

Introduction

Pour ce projet de licence il s'agissait de réaliser les programmes de microcontrôleur des cartes pré-réalisées part des projets précédents. Afin que sur la borne de départ soit affiché le plus petit temps de parcours et sur la borne d'arrivée la vitesse final. De plus le système devait pouvoir fonctionner en sans fil (du moins si j'avais le temps de mettre en œuvre une solution correspondante).

Sommaire

Introduction	5
Sommaire	6
Présentation du candidat.....	7
Présentation de l'association E-kart	8
Présentation du projet.....	9
1. Objectif.....	9
2. Cahier des charges.....	11
3. Les méthodes employées.....	14
4. Code vision AVR	19
5. Target 3001	20
6. Note oscilloscope	28
7. Exemple de test : fonctionnement des E/S	28
Résumé.....	32
ABSTRACT.....	33
Annexe 1 : dossier de réalisation de la carte wifi.....	34
Annexe 2 : 1 ^{er} programme de test de la borne d'arrivée	38
Annexe 3 : 1 ^{er} programme de test de la borne d'émission réception	43
Annexe 4 : 2 ^{ème} programme de test de la borne d'arrivée	47
Annexe 5 : 1 ^{er} programme de test de la borne départ.....	51

Présentation du candidat

Je m'appelle Denis LEGER, je suis en licence de microélectronique et électronique analogique. J'ai passé un baccalauréat STI génie électronique ainsi qu'un BTS système électronique et ai participé pendant quatre ans aux trophées robotique E=M6. J'ai choisi cette voie car tout ce qui se rapporte à la science m'intéresse et je me suis découvert un vif intérêt pour l'électronique au collège dans une option facultative que j'ai suivie pendant les deux années proposées, nommée NTA (nouvelle technologie appliquée) qui portait sur l'électronique, l'électrotechnique, la pneumatique, et la conception de projets techniques.

LEGER Denis

Appt n° 235
 12 allée de la Rochefoucauld
 71100. TOURS
 Tel : 06.87.88.76.04.

Chez parents :
 Rue des Bayards
 71150. PARIS L'HÔPITAL
 Téléphone 03.85.91.14.56

Informations personnelles	Situation de famille : célibataire Nationalité : Française Age: 21 ans. Date et lieu de naissance : 18/11/1988 à Saint-Rémy (71)
objectif	Obtention d'un projet de licence en électronique
Formations	Actuellement en licence de microélectronique analogique BTS (systèmes électroniques) au lycée N Niepce à Chalon-sur-Saône.(13/20) Bac STI (génie électronique) (mention AB) deux ans de NTA (nouvelle technologie appliquée) en 4ème et 3ème, ainsi que des cours supplémentaires de programmation en Terminale (langage étudié Smalltalk).
Expériences professionnelles	Février 2008 : 4 jours au Domaine Les vignes Blanches (tirage de sarments, mouchage de vignes) (71) Fin août 2007 : 3 jours de vendanges à St Aubin (21) Juin & juillet 2006 : 15 jours d'accolage chez Mr Demangeot (71), 2 mois au Domaine Les Vignes Blanches, une semaine d'effeuillage chez Mr Duchemin (21). Juillet 2005 : vendange en vert chez Mr Pierre Brazey (71). Juin 2005 : Domaine Les Vignes Blanches (71) et chez Mr Éric Duchemin (effeuillage vigne) (21). Emplois d'été au Domaine Les Vignes Blanches, à PARIS L'HÔPITAL 1 mois de travail en équipe dans les vignes en juin 2004.
Stage	Stage de 9 semaines à COMECA EBT suite à ma 1ère année de BTS Système électronique.
hobbies	Informatique(programmation), dessin, sculpture sur bois, jouer du synthétiseur (2 ans en initiation au piano), Natation (brevet 50m brasse), avion (Brevet d'Initiation à l'Aéronautique + stage découverte de 4H), Lecture (médiéval fantaisie), robotique (participation pendant 4 ans au trophée robotique E=M6), électronique, conception d'image et de clip 3D, musique (rock).
Langues	Allemand (scolaire), Anglais (scolaire), italien (notions).

Présentation de l'association E-kart

L'association E-kart met en compétition plusieurs établissements où les étudiants réalisent des karts entièrement électriques.

Les articles du règlement qui suivent, résument assez bien l'épreuve.

Article 2.1 Objectifs des épreuves

- Evaluer le travail pédagogique des étudiants qui ont travaillé sur le kart électrique.
- (Tester les performances du kart lors du challenge).

Article 2.2 Les sujets d'études

- Fabrication et montage mécanique du kart électrique.
- Recherche de sponsors.
- Organisation et travail en équipe.
- Etude technique en relation avec le kart électrique : comportement dynamique, performance, justification des choix technologiques, études économique, études de marchés...
- Réalisation du variateur et/ou du chargeur.
- Réalisation de l'électronique embarquée : afficheur de vitesse, tension courant, température,...
- Transmissions de données à un PC fixe.
- Développement de logiciel utile au kart électrique (site web, simulateur, ...).
- ...

e-Kart 2010 Site web : www.e-kart.fr

du jeudi 27 mai au samedi 29 mai 2010

Challenge de kart électrique organisé par l'Association e-Kart, l'Association ASTECH, la société Kartmasters, et l'Association Kartelec

Sur la piste extérieure du
Parc des Expositions de Vierzon
à Vierzon (Région Centre, Cher 18)
Site : www.e-kart.fr/2010/

Association e-Kart - Thierry LEQUEU - 152, rue de Grandmont - 37550 Saint AVER

Présentation du projet

1. Objectif

Mettre en œuvre l'épreuve de 50m départ arrêté qui comporte deux bornes une aux départ l'autre à l'arrivée. Commençons par la borne de départ :

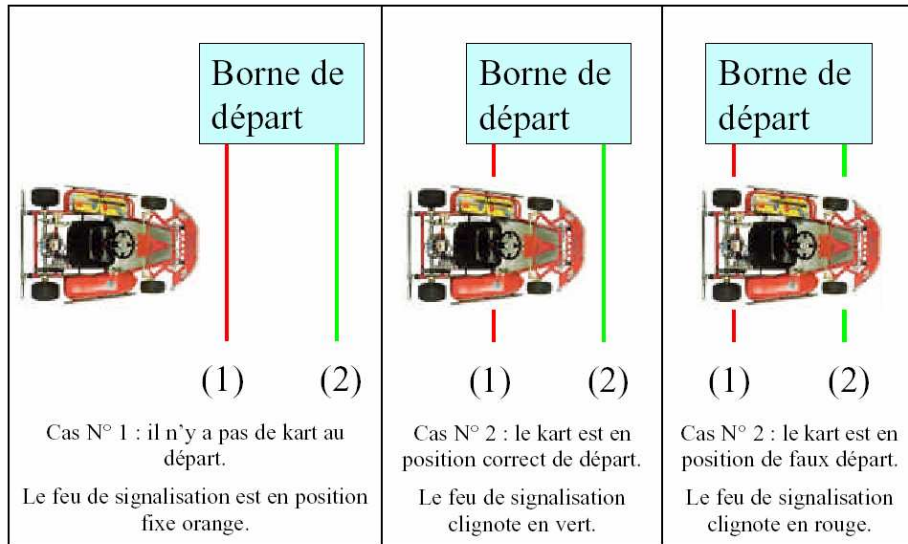
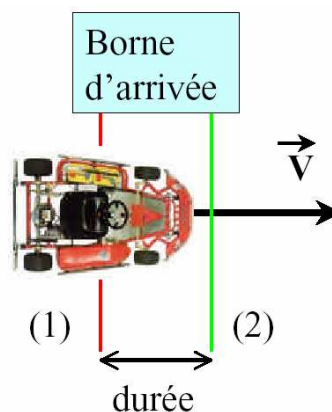


Figure 1.2. Schéma de principe de la mesure du temps pour l'épreuve de 50 mètres départ arrêté (Projet-50m-DA.ppt - Page 1).

Cependant l'utilisation des 3 feux de départ a changé dans le cas n°1 c'est le feu rouge et non orange qui est allumé. Puis dans la première solution du cas n°2 une courte attente est demandée avant le passage du feu rouge au feu vert afin d'éviter les départs intempestifs, et dans la seconde solution le feu orange s'allumera.

Ceci a été choisi afin de se rapprocher du mode de signalisation des feux de ville. De plus cette borne assure l'affichage de la durée de parcours la plus courte qui commence en cas de bon départ à la coupure de (2) et s'arrête à la réception du signal d'arrêt envoyé par la borne d'arrivée.

Continuons avec la borne d'arrivée :



La distance entre (1) et (2) étant connue (30cm) le temps entre la coupure de (1) et celle de (2) permet de déterminer la vitesse du kart à l'arrivée qui est alors affichée sur la borne. De plus lors de la coupure de (1) est envoyé à la borne de départ le signal de fin du parcours servant à arrêter le comptage de la durée du parcours.

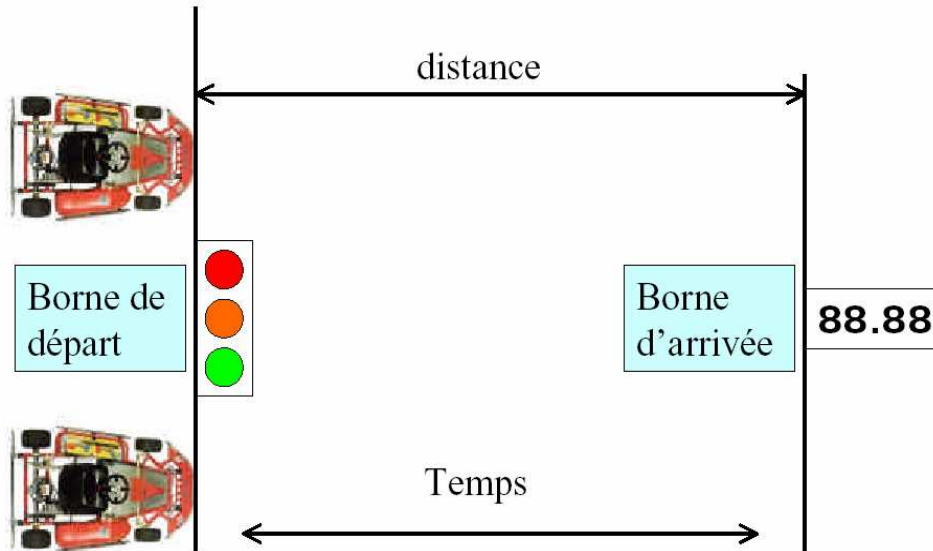


Figure 1.1. Schéma de principe de la mesure du temps pour l'épreuve de 50 mètres départ arrêté
(Projet-50m-DA.ppt - Page 1).

2. Cahier des charges

I°) Schéma fonctionnel

1°) premier degré

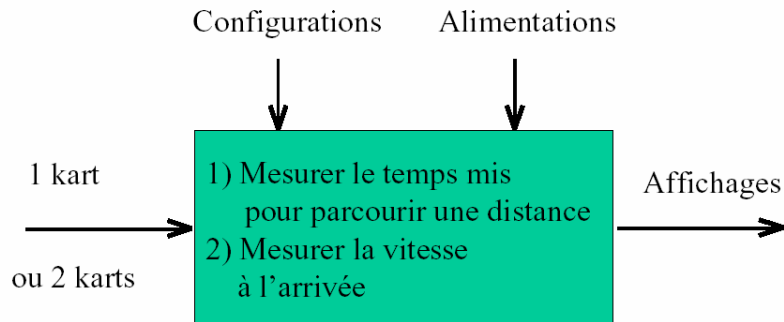


Figure 2.2. Synoptique de niveau 1 du projet électronique
 (Projet-50m-DA.ppt - Page 5).

2°) troisième degré de la 1^{ère} borne

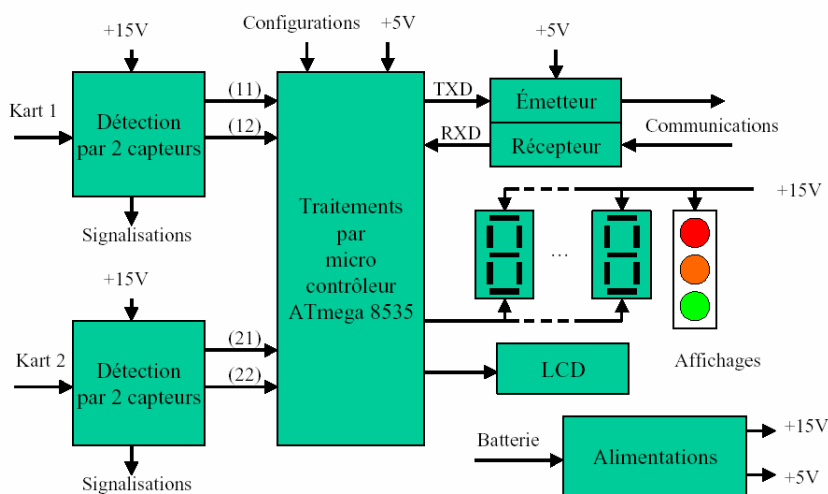


Figure 2.4. Synoptique de niveau 3 d'une borne
 (Projet-50m-DA.ppt - Page 7).

II°) expression du besoin

1°) Contraintes liées à l'architecture matérielle ou logicielle :

Des carte type ont déjà réalisée il faudra les adaptées par les logiciels pour effectué les différentes taches et applications que l'on effectuera selon les besoin des différents niveaux d'application ou ce situeront ces cartes. Il y a, a gérée un à deux affichages par écrans de six digits mit sur le panneau d'arrivée (ou les deux) une commande pour l'arbitre ainsi que des feux de signalisation.

2°) Contraintes spécifiques liées à l'environnement :

L'utilisation d'antenne a été recommandée afin d'éviter les câbles pouvant entraver la circulation des karts. De plus une possible communication de donnée avec un ordinateur a été évoqué et sera a réalisée.

3°) Contrainte économique :

Il est nécessaire de rester dans les moyens du club d'e-kart, et de ne pas trop empiéter sur le budget.

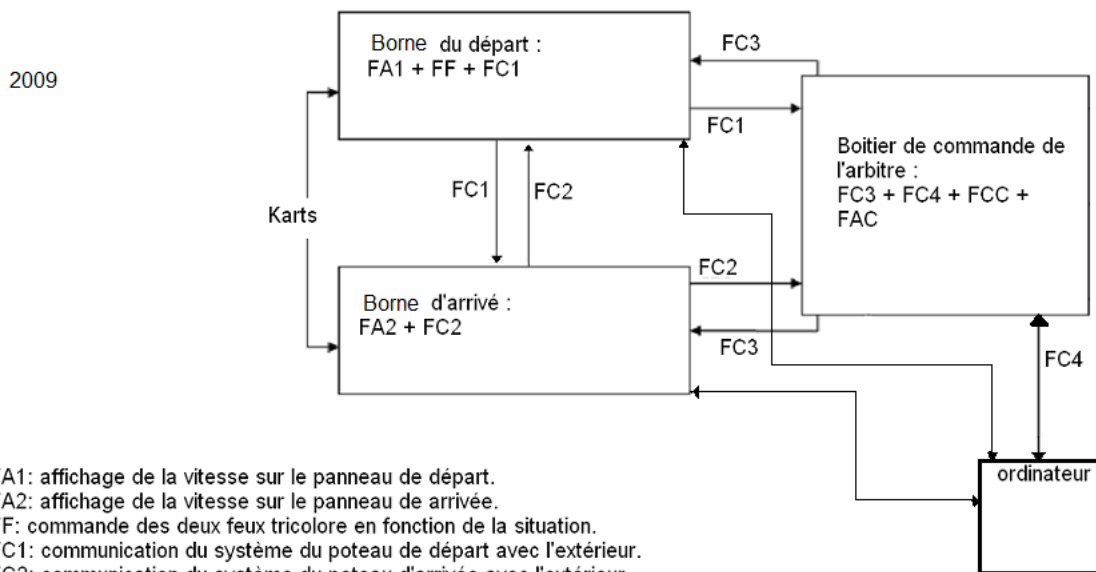
4°) Documents et moyens technologiques mis à disposition :

2 salles de l'IUT ont été mises à disposition ainsi que le matériel contenu dans celle-ci.
Matériel existant.

Dossier du projet 2 document (16 et 3 pages) entier, des rapports des étudiants, ainsi que les documentations techniques des cartes électroniques.

III°) fonctions

Schéma fonctionnel du système :



FA1: affichage de la vitesse sur le panneau de départ.
FA2: affichage de la vitesse sur le panneau de arrivée.
FF: commande des deux feux tricolore en fonction de la situation.
FC1: communication du système du poteau de départ avec l'extérieur.
FC2: communication du système du poteau d'arrivée avec l'extérieur.
FC3: communication du système de commande de l'arbitre avec l'extérieur.
FC4: communication du système de commande de l'arbitre avec un ordinateur.
FCC: Contrôle des différentes communications.
FAC: commande de l'arbitre au système.

Description des fonctions :

FA1: affichage du temps sur la borne de départ.

Sortie des signaux par la broche 20 pins de la carte du poteau a un afficheur composée à 6 digits (évité les delles fortes luminosité sur ce poteau).

FA2: affichage de la vitesse sur la borne d'arrivée.

Sortie des signaux par la broche 20 pins de la carte du poteau a un afficheur composée à 6 digits.

FF: commande des deux feux tricolore en fonction de la situation comme pour l'afficheur.

FC1: communication du système de la borne de départ avec l'extérieur.

Entré et sortie par la liaison RS 232 du microcontrôleur allant a une carte de conversion en signaux 485 (a ma charge).

FC2: communication du système de la borne d'arrivée avec l'extérieur.

Entré et sortie par la liaison RS 232 du microcontrôleur allant a une carte de conversion en signaux 485 (a ma charge).

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

FC3: communication du système de commande de l'arbitre avec l'extérieur.

Entré et sortie par la liaison RS 232 du microcontrôleur allant a une carte de conversion en signaux 485 (a ma charge).

FC4: communication du système de commande de l'arbitre avec un ordinateur.

Entré et sortie par la liaison RS 232 du microcontrôleur allant a une carte de conversion en signaux 485 (a ma charge).

FCC: Contrôle des différentes communications.

Cette fonction sera entièrement prit en charge par le microcontrôleur de l'arbitre qui assurera une rotation des communications.

FAC: commande de l'arbitre au système.

Bouton que l'arbitre activera ou non lui permettant :

- De passer le relais des commandes au microcontrôleur.
- De dire s'il y a 1 ou 2 kart (et sur quel voie est le kart dans le cas ou il n'y en a qu'un).
- De signalé au système un problème sur l'une des voie (arrêt de la gestion de cette voie).
- Lancer le départ.

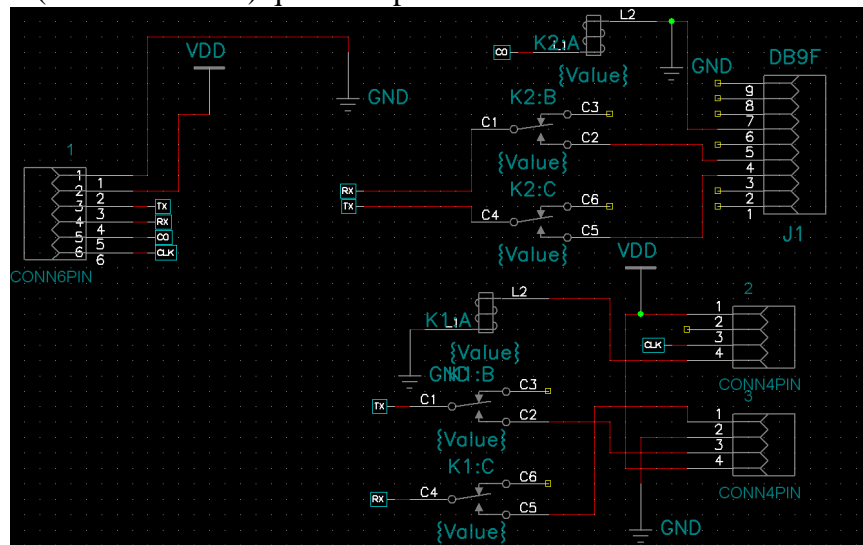
3. Les méthodes employées

Tout d'abord j'ai commencé par m'intéresser au moyen de mise en œuvre absolue c'est-à-dire, de la méthode de communication parfaite (afin de savoir si certaines notions seraient à prendre en compte dans la réalisation des programmes).

Deux solutions ont été retenues. La première utiliserait un système d'émetteur récepteur AM et l'autre du wifi.

Tout d'abord la première solution elle est basée sur le QFM-TRX1-24G qui est double émetteur et double récepteur.

Voici le schéma (fait sous PCAD) qui correspond :



Ainsi que le tableau des pins justifiant ce schéma :

Pin Descriptions

Pin No	Pin Name	Pin Type	Description
1	VCC	Power	1.8V-3.6V digital power supply for digital I/O's and for the digital core voltage regulator
2	SI	Digital Input	Serial configuration interface, data input
3	SCLK	Digital Input	Serial configuration interface, clock input
4	SO	Digital Output	Serial configuration interface, data output. Optional general output pin when CSn is high
5	GDO2	Digital Output	Digital output pin for general use: • Test signals • FIFO status signals • Clear Channel Indicator • Clock output, down-divided from • Serial output RX data
6	GND	Ground	GND
7	GDO0	Digital I/O	Digital output pin for general use: • Test signals • FIFO status signals • Clear Channel Indicator • Clock output, down-divided from XOSC • Serial output RX data • Serial input TX data Also used as analog test I/O for prototype/production testing
8	CSn	Digital Input	Serial configuration interface, chip select

Pour cette solution il faut par point d'utilisation (pour chaque appareillage pour lequel le montage est nécessaire) :

- 2 relais à 2 commutateurs
- Une prise DB9
- Un connecteur 6 broches
- Un connecteur 8 broches
- 1 QFM-TRX1-24G

Avec A commandé : le QFM-TRX1-24G au Prix: 5,17 € / U (TTC).

La deuxième solution étant l'utilisation du wifi, cette solution permettrait donc d'insérer un ordinateur dans le système sans appareillage particulier vu que les ordinateurs portables possèdent le wifi.

Après recherche je me suis rendu compte qu'un projet avait déjà été réalisé sur le sujet sous la tutelle de M. LEQUEU, l'étudiant en charge était M. HAHN Jean-Charles.

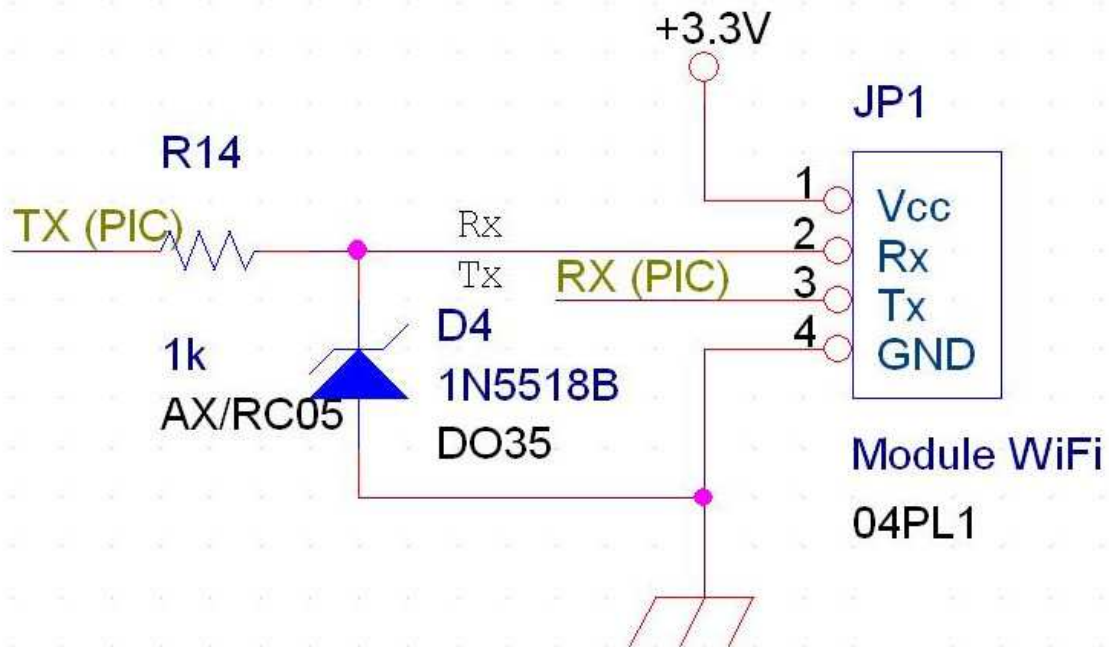
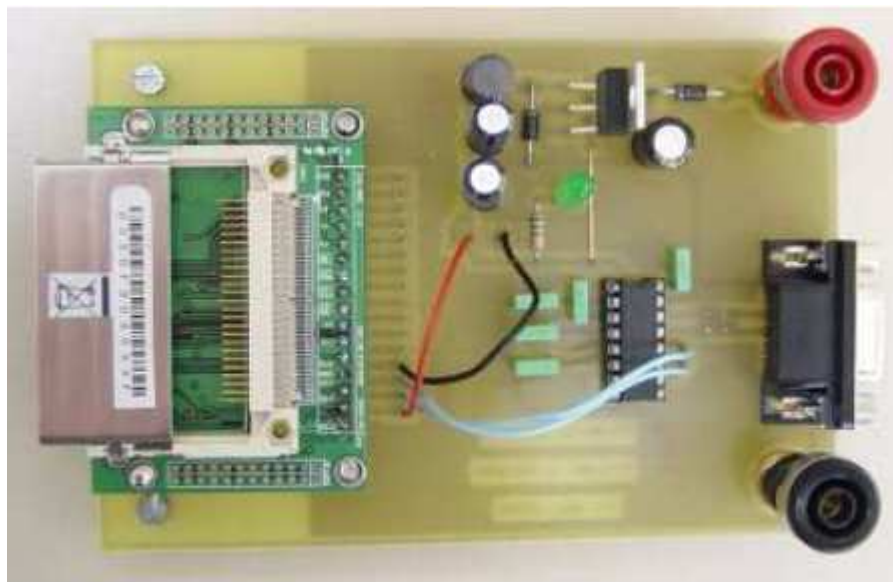


Schéma électrique de l'implémentation du module Wifi



Carte de test réalisé par J-C HAHN

A partir de ce travail j'ai pu aboutir au résultat visible en annexe. Suite à ce travail la solution retenue fut la future utilisation du wifi. J'ai donc imaginé un système de codage des informations correspondant à ce qui suit.

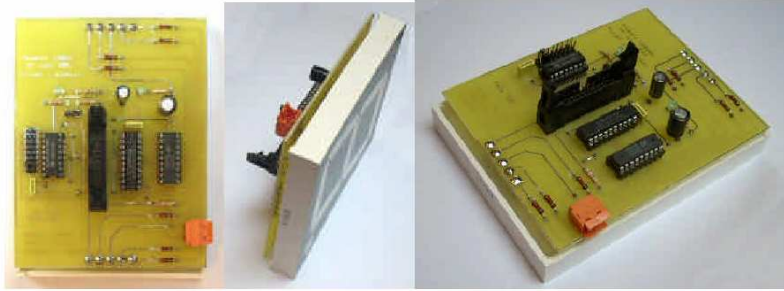
valeurs	signification
De 00 à 0F	Kart1 temps de parcours centième
De 10 à 1F	Kart1 temps de parcours dixième
De 20 à 2F	Kart1 temps de parcours unité
De 30 à 3F	Kart1 temps de parcours dizaine
De 40 à 4F	Kart2 temps de parcours centième
De 50 à 5F	Kart2 temps de parcours dixième
De 60 à 6F	Kart2 temps de parcours unité
De 70 à 7F	Kart2 temps de parcours dizaine
De 80 à 8F	Kart1 vitesse unité
De 90 à 9F	Kart1 vitesse dizaine
De A0 à AF	Kart1 vitesse centaine
De B0 à BF	Kart2 vitesse unité
De C0 à CF	Kart2 vitesse dizaine
De D0 à DF	Kart2 vitesse centaine
De E0 à EF	Commande portable de l'arbitre
De F0 à FF	PC portable de l'arbitre

Tableau des attributions

Ensuite j'ai donc dû étudier les cartes à programmer qui, quelque soit l'utilisation sont toutes les mêmes, il s'agit des Carte microcontrôleur ATmega 8535 réalisées aussi lors d'un projet précédant, ainsi que les afficheurs et les feux.



Carte microcontrôleur ATmega 8535



Afficheur 7segments utilisés

Cette étude m'a permis de connaître le pré-brochage des pins de l'ATmega 8535 sur la carte préconçue.

PDIP

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5
(TXD) PD1	15	26	PC4
(INT0) PD2	16	25	PC3
(INT1) PD3	17	24	PC2
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

ATmega 8535

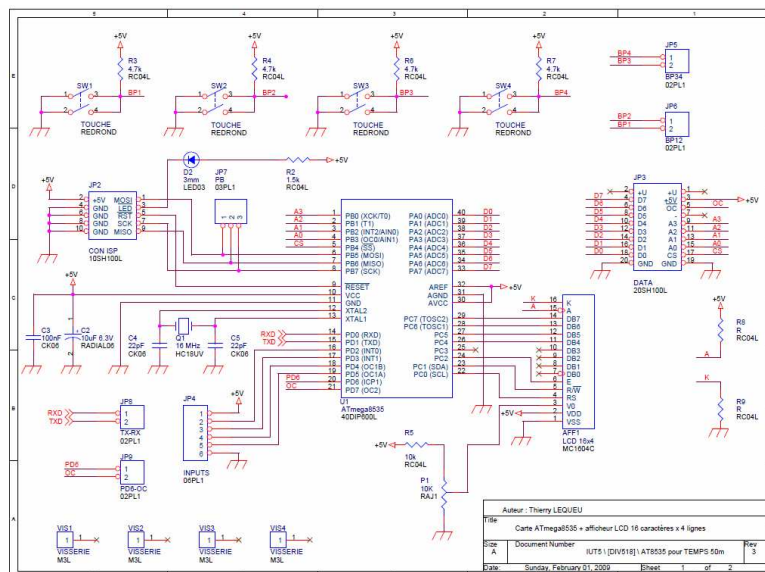


Schéma de la carte de l'ATmega

Les pins choisis sont donc :
 PD6 = rayon (1) de la borne

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle
PD5 = rayon (2) de la borne
PD4 = feu rouge (départ uniquement)
PD3 = feu vert (départ uniquement)
PD2 = feu orange (départ uniquement)

Pour ce qui est du calcul de la vitesse sur la borne d'arrivée la distance entre (1) et (2) étant de 30 cm la vitesse maximum mesurable choisie étant 200 km/h la mesure devant être faite au km/h près, j'ai abouti à ces calculs :

$1 \text{ km/h} = 1000\text{m}/3600\text{s} = 0,27\text{m/s} = 270\text{mm/s}$ pour ce qui est de la précision

$(30*3600)/(199*1000*100) = 5,427\text{ms}$

199km/h => fait que les 30 cm sont parcourus en 5,427ms

200km/h => fait que les 30 cm sont parcourus en 5,4ms

201km/h => fait que les 30 cm sont parcourus en 5,373ms

Le delta T est donc de 27 μs

Le delta T choisi est de 25 μs car la valeur min de fréquence de comptage de l'ATmega est de 5 μs .

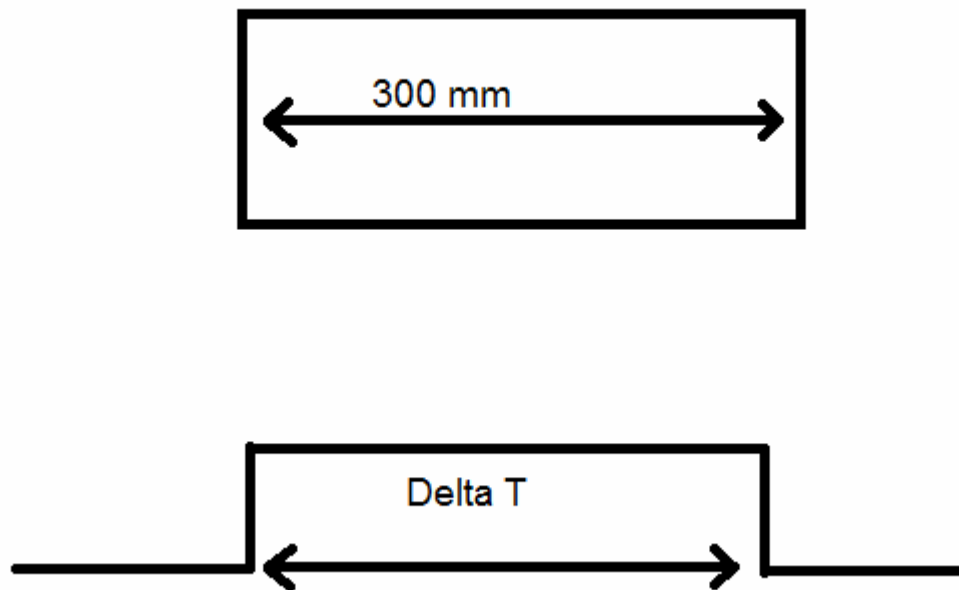


Schéma du système de mesure

4. Code vision AVR

Il s'agit du logiciel de programmation que j'ai eu à utiliser en voici le manuel d'utilisation que j'ai tiré.

Création du fichier de programme :

File => new : Project

File => Générate Save and exit

Ps : « nommer du même noms les fichiers du projet »

Pour programmer :

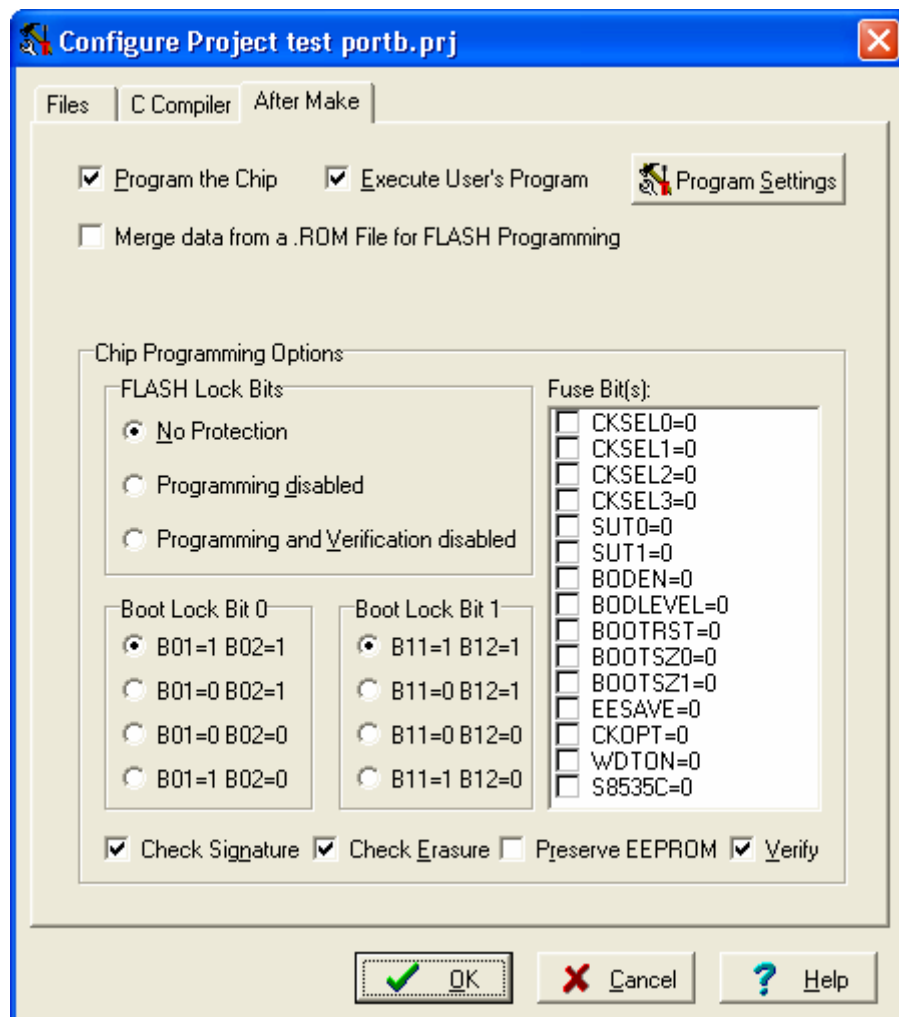
Project => configure => After make

Programme the chip

Run the programme

Setting => Programmer => Utilitaire : Kanda système

Broche : LPT1



validation de la programmation

5. Target 3001

Il s'agit du logiciel avec lequel j'ai réalisé les typons de la carte wifi j'ai utilisé la version 13 qui est gratuite et légal.

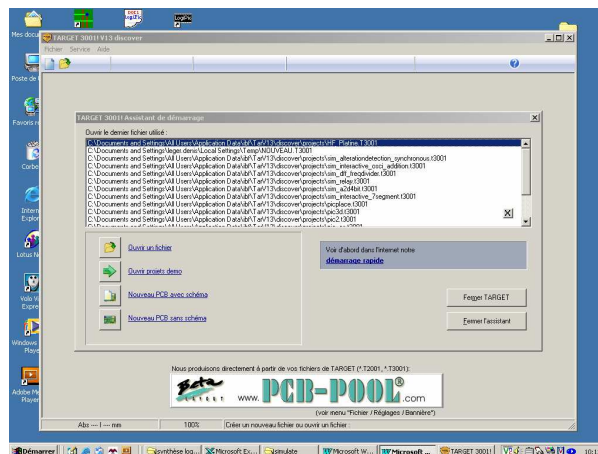
I) Introduction

Il s'agit d'un logiciel de conception de carte sauf qu'ici on peut faire des cartes plus abouties, grâce aux bibliothèques que l'on peut mettre à jours gratuitement en quelque clic dans le logiciel. C'est vraiment plus agréable que de tout devoir chercher sur Internet soi-même. Cependant il existe deux version une payante et l'autre non. La version gratuite est limité à 250 broches et 4 pages. Et voilà les prix :

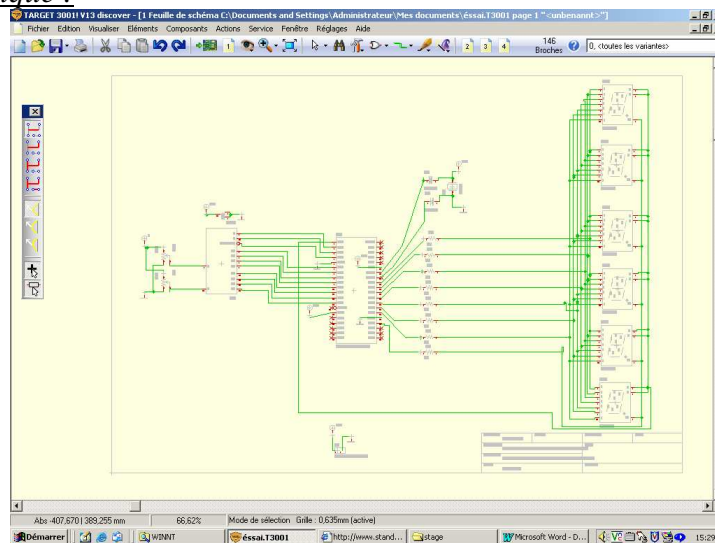
Version	Caractéristiques	Prix (EUR)
TARGET 3001 light	-400 broches/pastilles -2 couches de cuivre -simulation pour 30 signaux -toutes les fonctions actives -accès au serveur du composants -utilisable commercialement -licence monoposte	EUR 59,00
TARGET 3001 smart	-700 broches/pastilles -2 couches de cuivre -simulation pour 50 signaux -toutes les fonctions actives -accès au serveur du composants -utilisable commercialement -licence monoposte	EUR 159,00
TARGET 3001 economy	-1.000 broches/pastilles -4 couches de cuivre -simulation pour 75 signaux -toutes les fonctions actives -accès au serveur du composants -utilisable commercialement -licence monoposte	EUR 569,00
TARGET 3001 professional	-broches/pastilles illimité -100 couches de cuivre -simulation pour 100 signaux -toutes les fonctions actives -accès au serveur du composants -utilisable commercialement -licence monoposte	EUR 1.649,00
TARGET 3001 design station	-broches/pastilles illimité -100 couches de cuivre -simulation pour signaux illimité -toutes les fonctions actives -accès au serveur du composants -utilisable commercialement -licence monoposte	EUR 2.999,00

Je n'ai testé que la version gratuite vous verrez ci dessous les résultats.

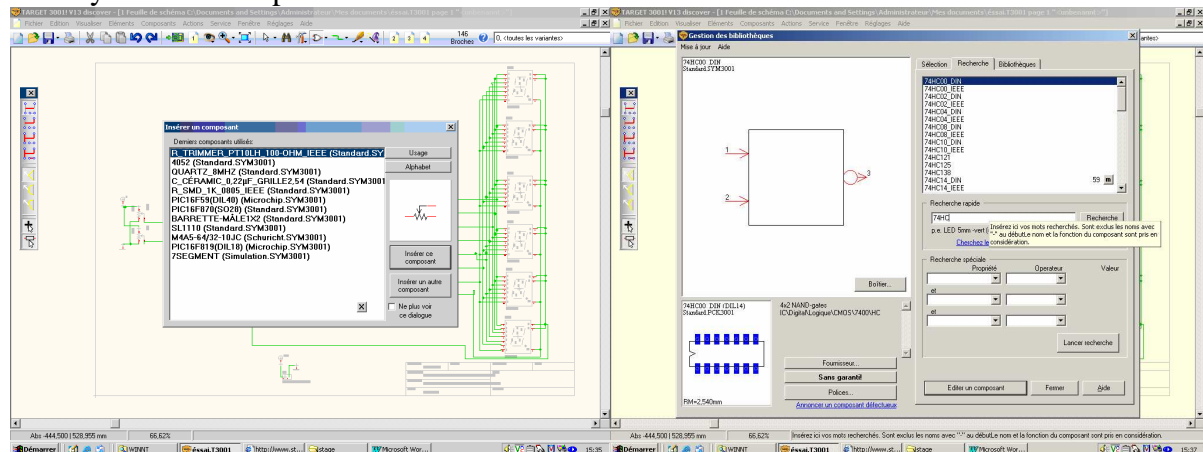
II) accueil logiciel



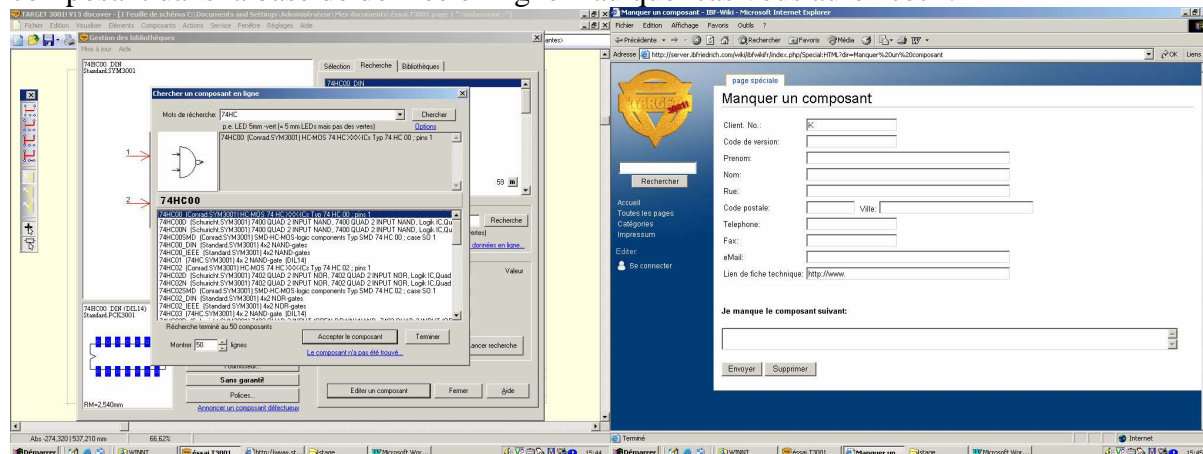
Il s'agit plus d'une fenêtre proposant les possibilités basic.



En cliquant sur l'icône de la porte « ou » on arrive à la page de sélection des composants voici le story board correspondant :

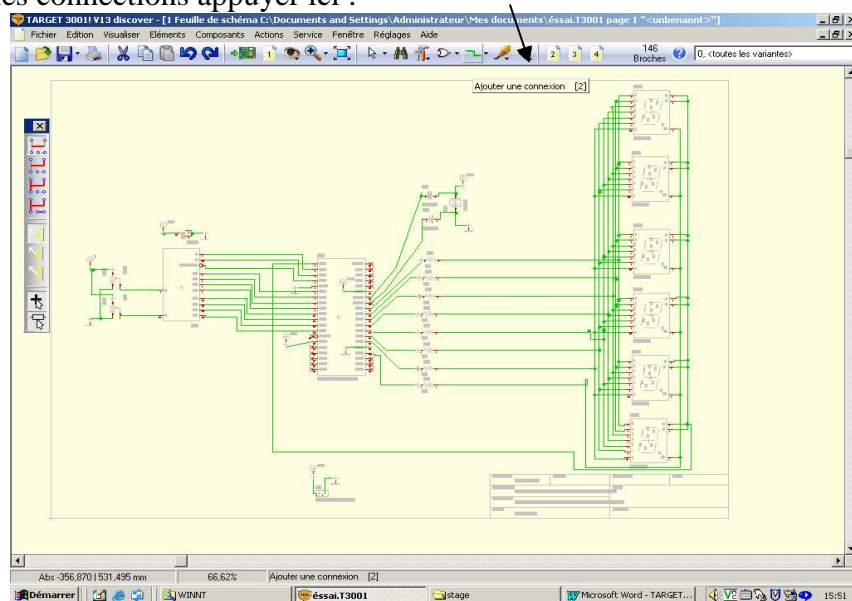


- cliquez sur « insérer un autre composant »
- cliquez sur « éditer un composant » après l'avoir choisi ou cliquez sur « cherchez le composant dans la base de donnée en ligne » au quel cas vous aurez ceci :



- la base de donnée Internet
- si vous n'avez toujours pas ce que vous voulez cliquer sur « le composant n'a pas été trouvé » et demander-le.

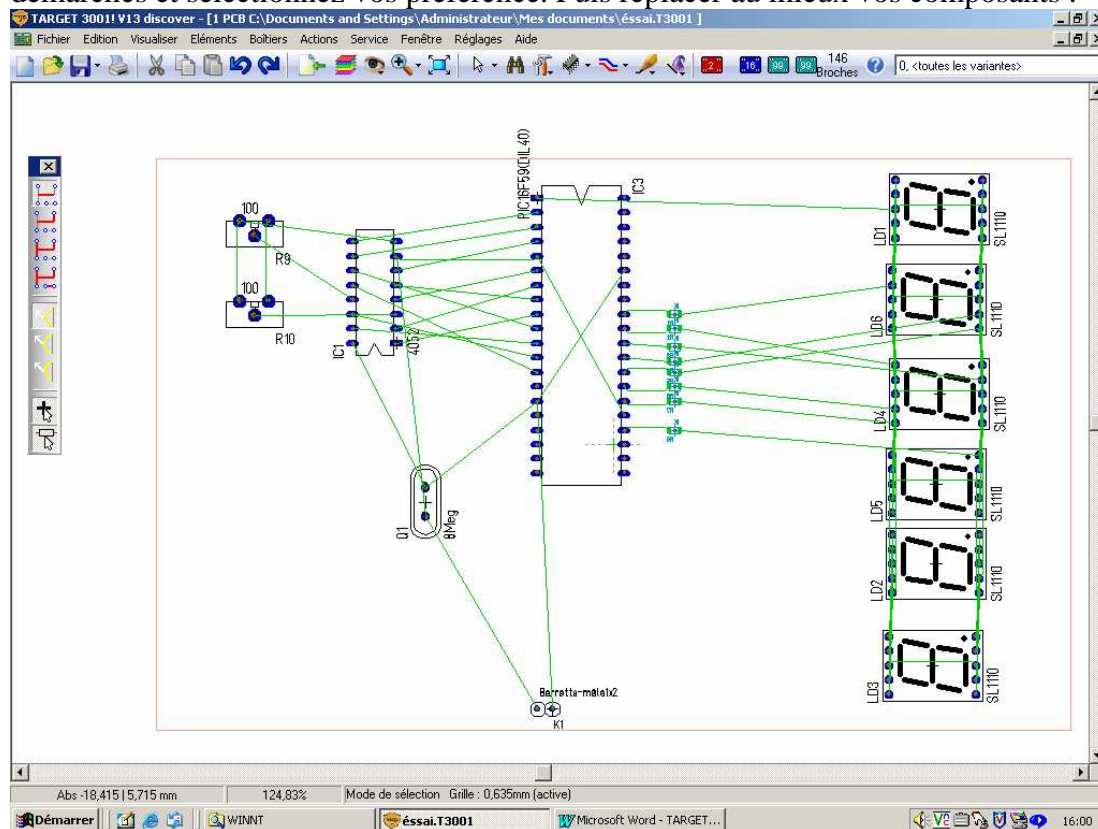
Université François Rabelais
 Institut Universitaire de Technologie
 Département Génie électrique et Informatique Industrielle
 Puis éditer le composant pour le placé le nombre de fois qu'ils vous le faut. Appuyer sur
 « echap » quand c'est fini.
 Pour mettre les connections appuyer ici :



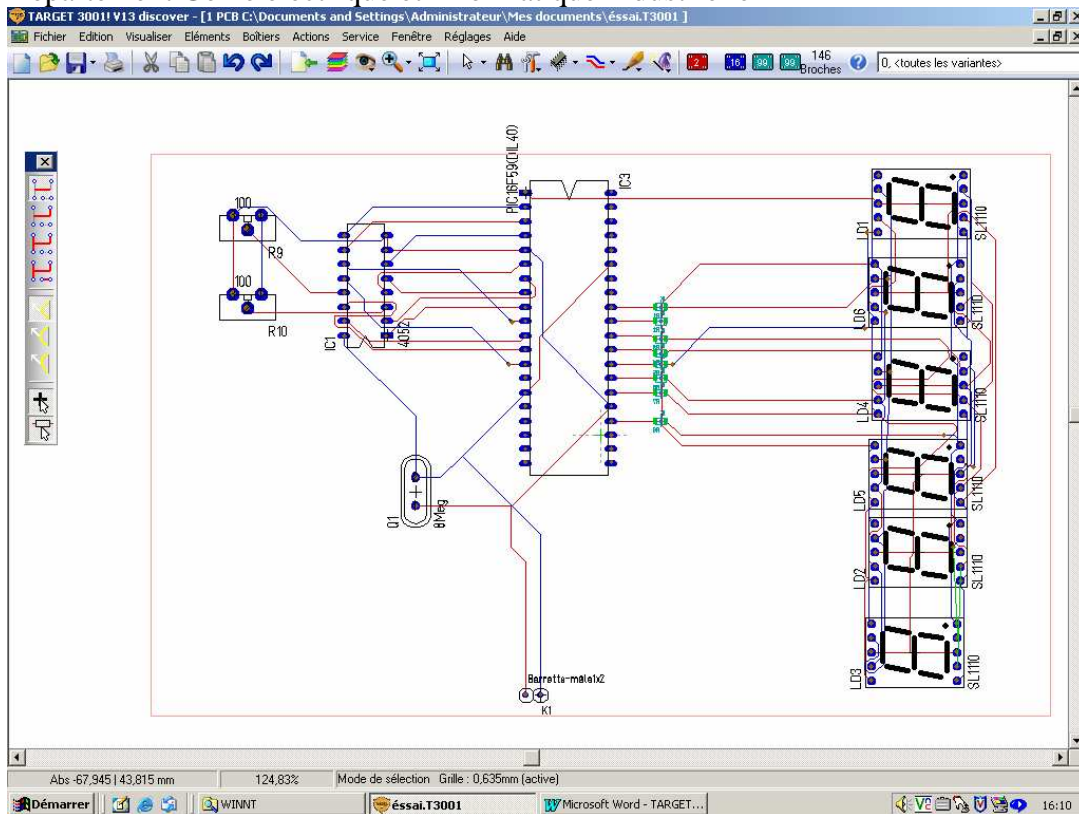
Toute autre démarche m'a été indiquée par le logiciel en cas de nécessité.

IV) le PCB

Cliquez sur le symbole « de la carte pointé d'une flèche verte », et il se lance. Lisez les démarches et sélectionnez vos préférence. Puis replacer au mieux vos composants :

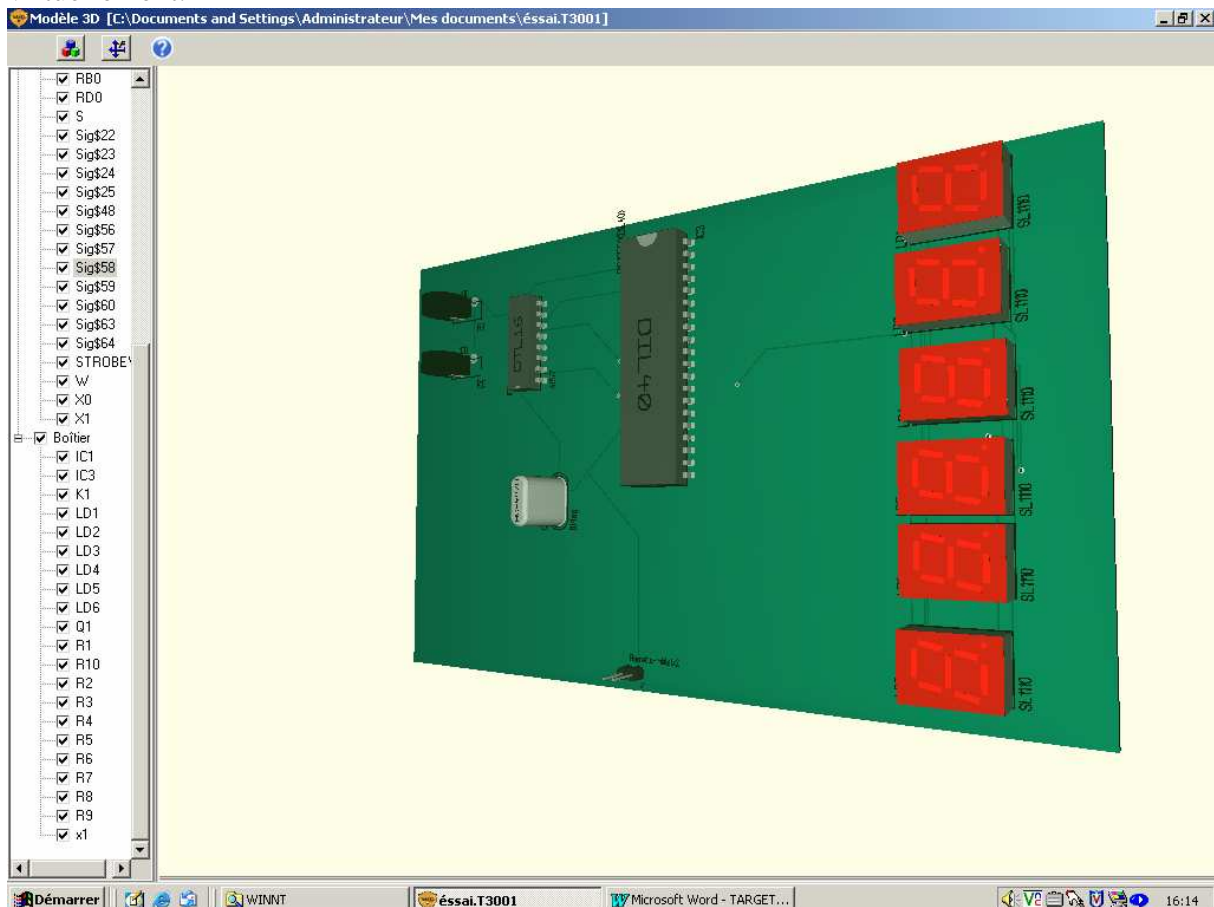


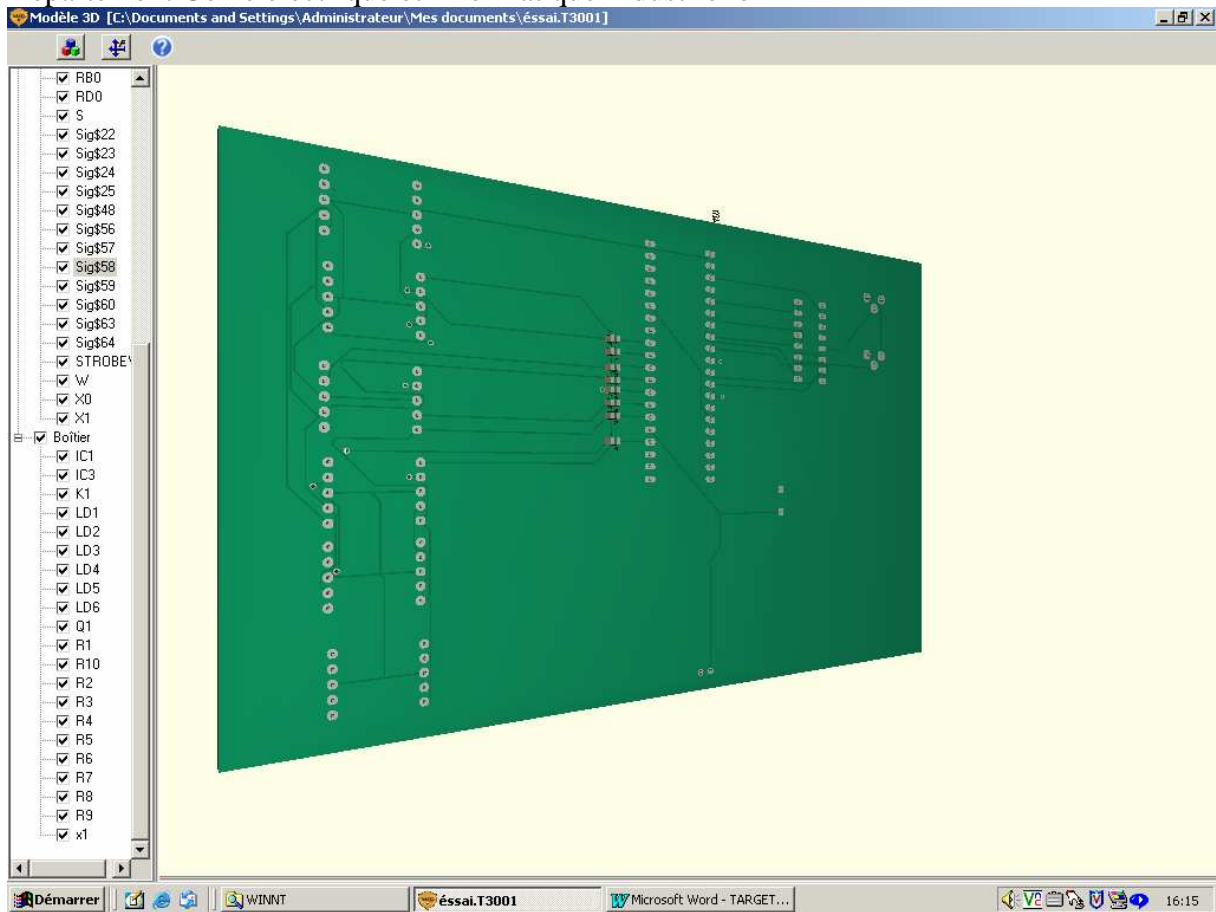
Faite action autorouteur et choisissez celui que vous voulez utilisé le routeur externe est une démo alors attention il est peut être plus pratique mais il n'est pas complet. Et voilà ce que l'on peut avoir :



Avit 3D

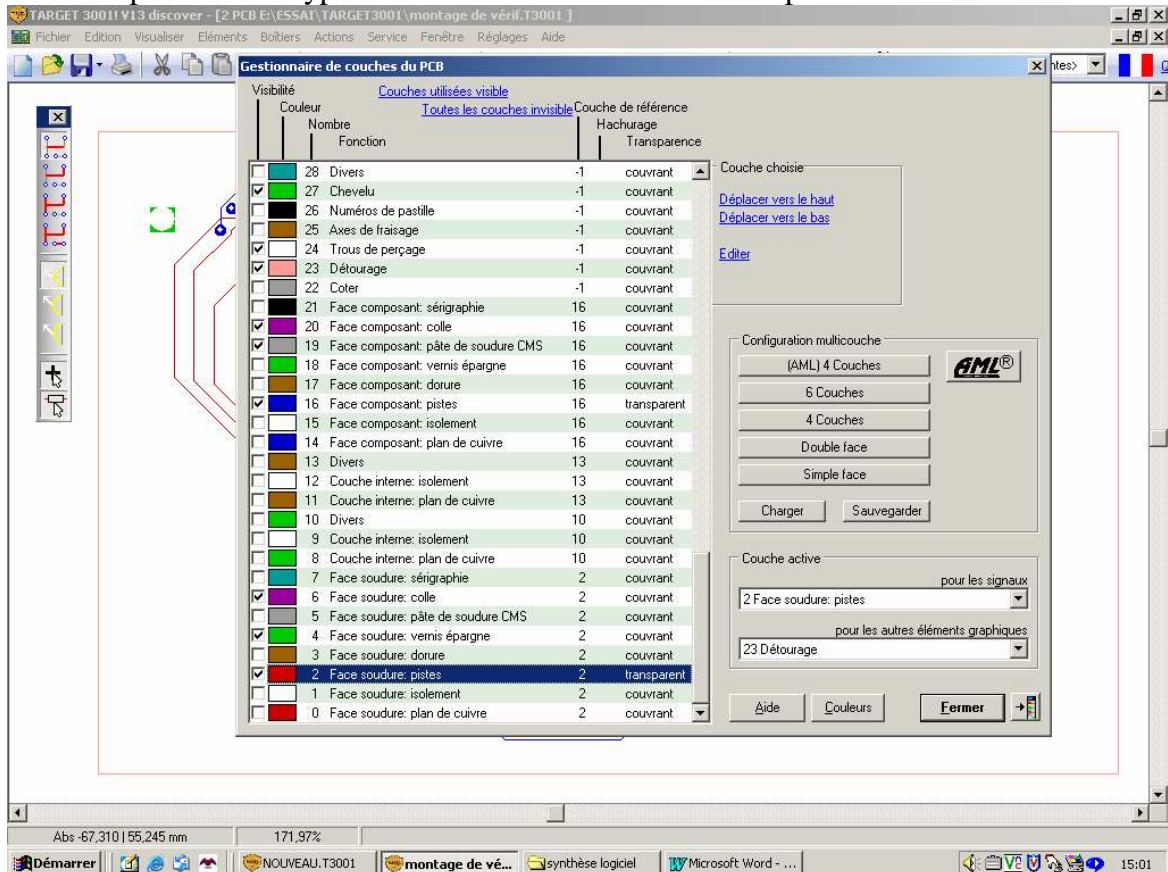
Depuis l'éditeur PCB cliquer sur visualiser puis Avit 3D et regarder votre carte finie virtuellement.



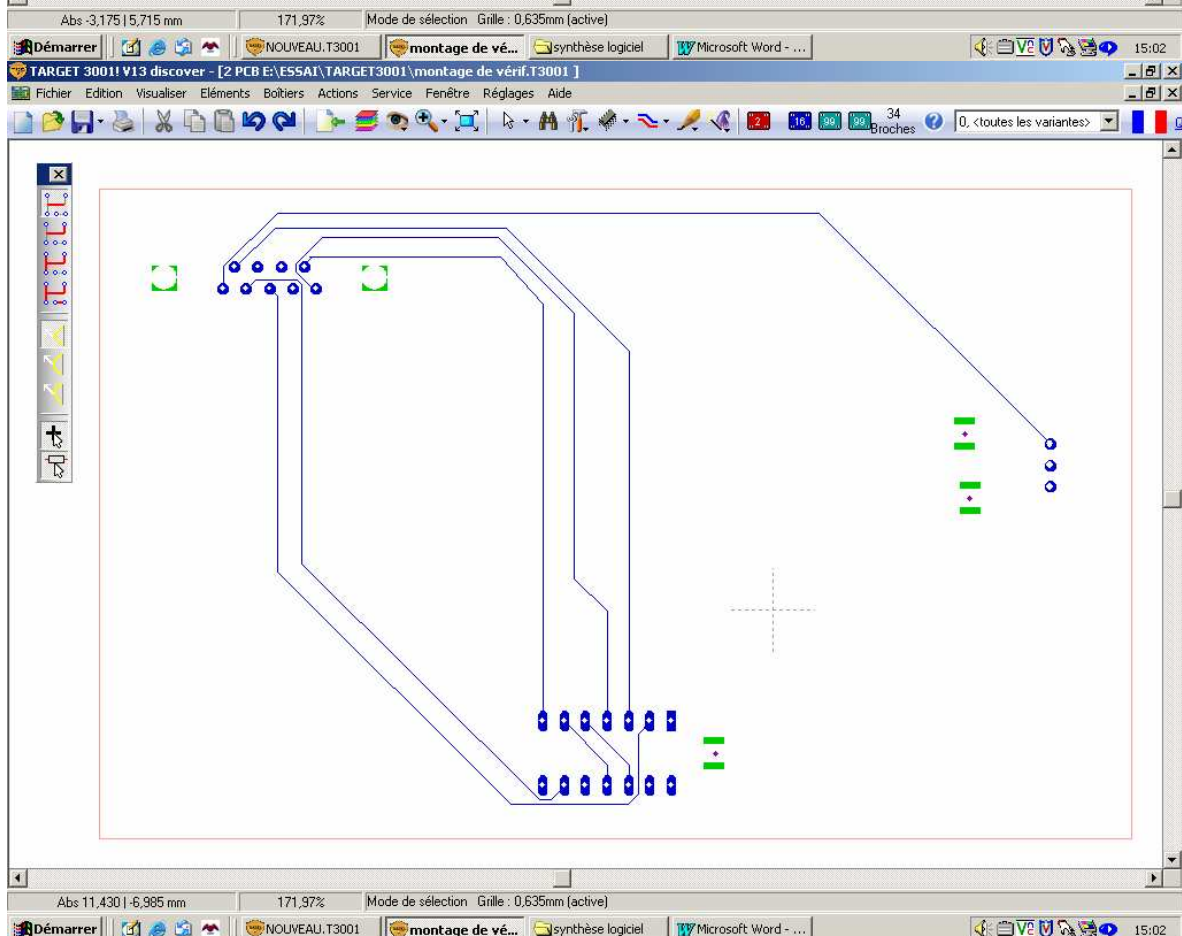
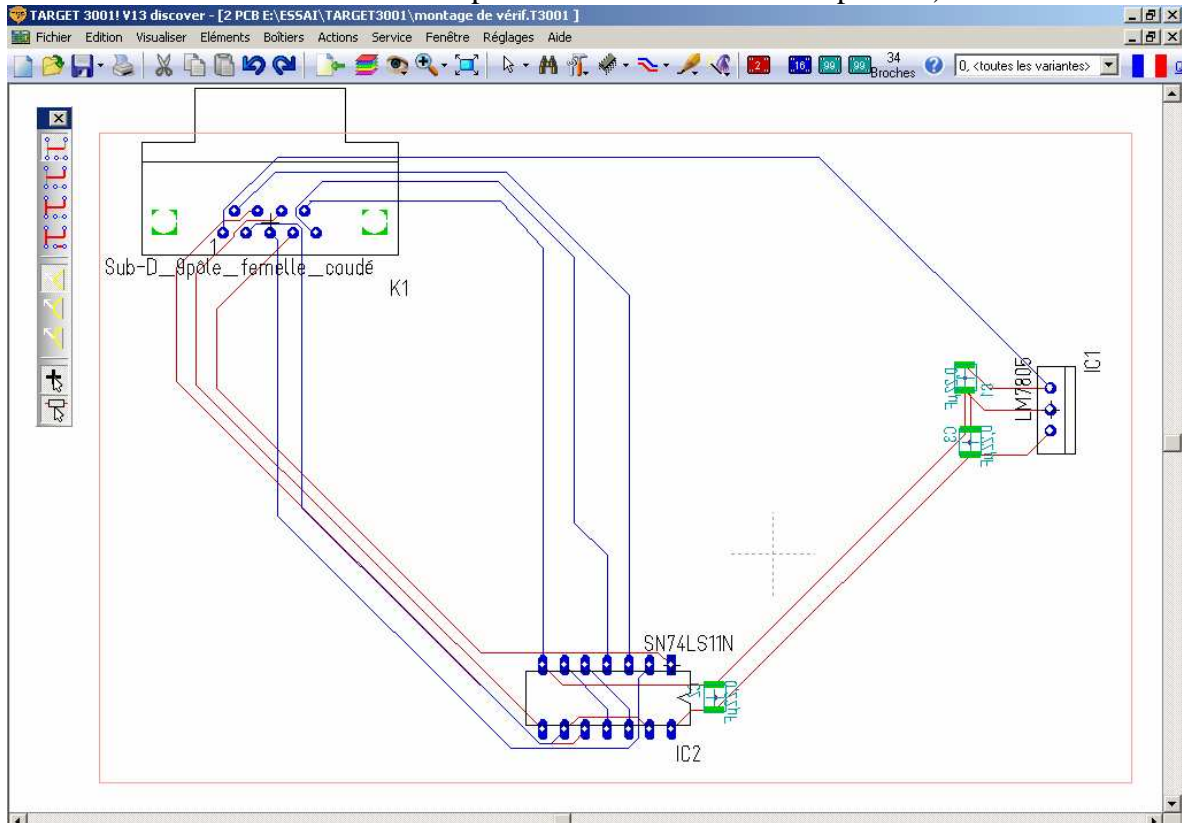


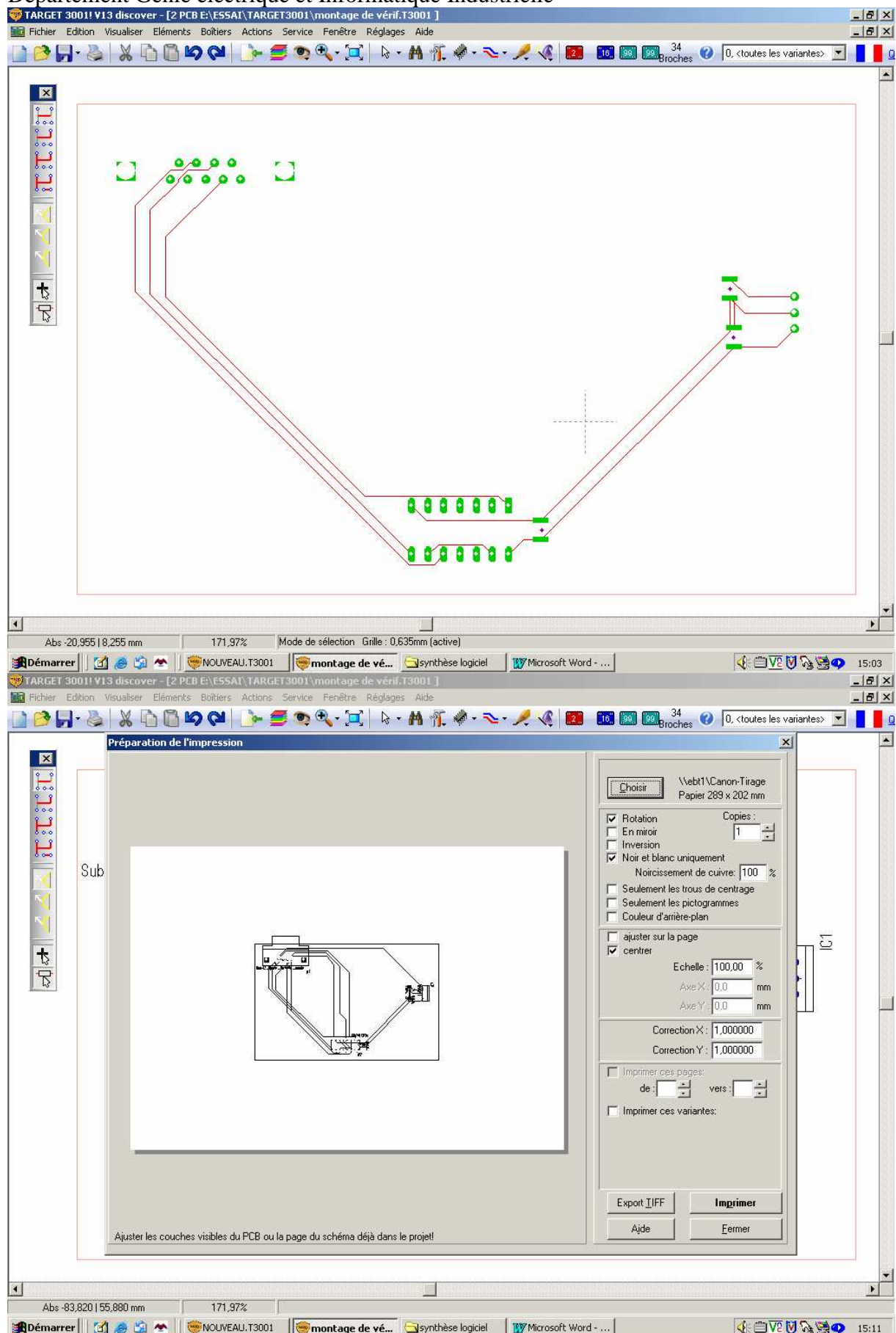
V)typon

Pour l'impression des typons il faut aller dans « visualiser » puis « Couche... ».



Université François Rabelais
 Institut Universitaire de Technologie
 Département Génie électrique et Informatique Industrielle
 Et la fenêtre ouverte sélectionné les données voulut. L'impression finit le travail (ne pas oublier d'inversé le sens du schéma pour la face dessous les composants).





Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

VI)librairie

Internet

VII)revendeur

Ing.-Buero FRIEDRICH

Harald Friedrich, Dipl.-Wirtsch.-Ing.(TU)

Am Schwarzen Rain 1

36124 Eichenzell Allemagne

Tel.: ++49 (0) 6659 / 919444

Fax: ++49 (0) 6659 / 919445

target@ibfriedrich.com ou www.ibfriedrich.com

Coordonnées bancaires:

Genossenschaftsbank Fulda

Bahnhofstrasse 1, D-36037 Fulda,

Germany

IBAN:

DE13530601800002004135

BIC: GENODE51FUL

6. Note oscilloscope

Pour obtenir une capture d'écran des oscilloscopes de la salle voici la démarche :
Util => copie d'écran => GIF fichier (interne)

Puis on récupère par explorer en ftp en mettant [ftp://\(ip de l'oscilloscope\)](ftp://(ip de l'oscilloscope)) dans la barre d'adresse.

7. Exemple de test : fonctionnement des E/S

Programme du test :

```
/******
```

```
This program was produced by the  
CodeWizardAVR V1.25.3 Evaluation  
Automatic Program Generator  
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : Test-LCD  
Version : 1  
Date   : 15/02/2007  
Author : Freeware, for evaluation and non-commercial use only  
Company : Thierry  
Comments:
```

```
Chip type      : ATmega8535  
Program type   : Application  
Clock frequency : 16,000000 MHz  
Memory model   : Small  
External SRAM size : 0  
Data Stack size : 128  
*****/
```

```
#include <mega8535.h>  
/* the LCD module is connected to PORTC */  
#asm  
    .equ __lcd_port=0x15  
#endasm
```

```
/* now you can include the LCD Functions */  
#include <lcd.h>
```

```
// Declare your global variables here
```

```
void main(void)  
{  
// Declare your local variables here
```

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

```
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
```

```
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xFF;
```

```
// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out
Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;
```

```
// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0xFF;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
```

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
```

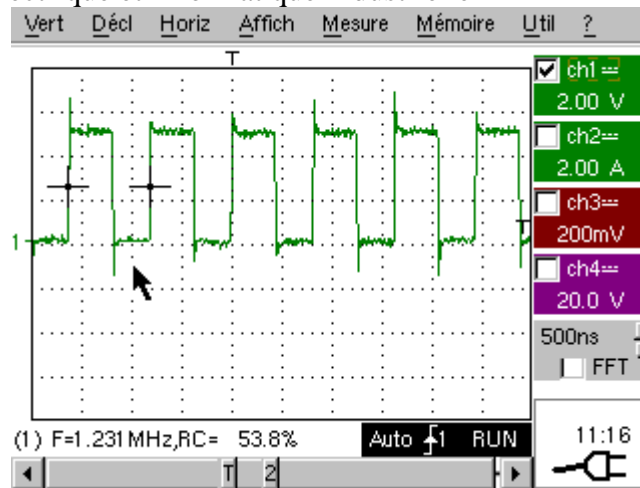
```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
```

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
```

```
/* initialize the LCD for 2 lines & 16 columns */
lcd_init(16);
```

```
/* switch to writing in Display RAM */
lcd_gotoxy(0,0);
lcd_putsf("Projet Programmation1 ATmega8535 - Test de l'afficheur LCD");
```

```
while (1)
{
    // Place your code here
    PORTB=0x00;
    PORTB=0xFF;
};
}
```



Résultat à l'oscilloscope

Résumé

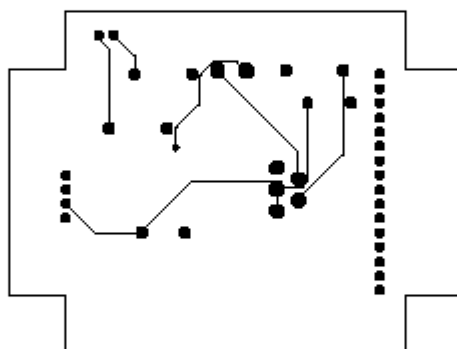
Le travail qui m'a été confié est la réalisation d'un système de chronométrage pour une épreuve de 50m départ arrêté de kart électrique. Le système est composé de 2 bornes une de départ l'autre d'arrivée. La borne de départ affiche le temps de parcours, le chronomètre jusqu'à l'ordre d'arrêt de la borne d'arrivée et gère les feux de départ. La borne d'arrivée envoie le signal de fin de comptage de la borne de départ, calcul la vitesse du kart à l'arrivée et l'affiche.

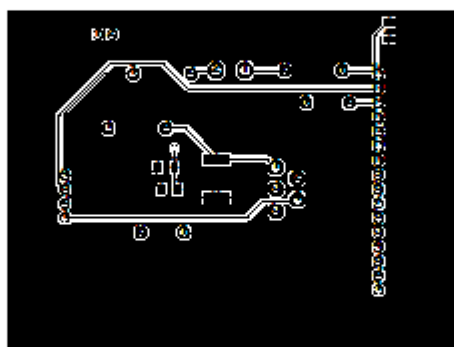
La possibilité grâce au wifi d'intégrer une commande utilisateur ainsi qu'un ordinateur au système a été ajouté.

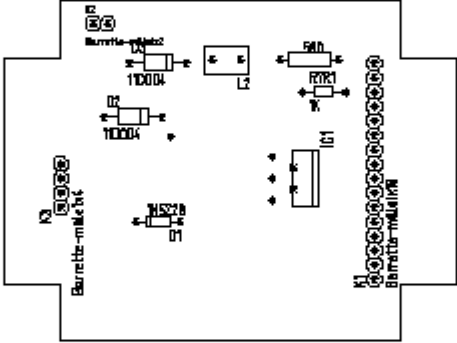
ABSTRACT

The job which was entrusted to me is the realization of a system of timing for a test of 50m standing start of electrical go-kart. The system is composed of 2 demarcations one of departure other one of arrival. The demarcation of departure shows the time of course, the jusqu chronometer 'à the order of stopping of the demarcation of arrival and manages fires of departure. The demarcation of arrival sends the sign of the end of counting of the demarcation of departure, counting the speed of the go-kart at arrival and poster.

Possibility thanks to the wifi to insert an order user as well as a computer in the system was added.







Annexe 2 : 1^{er} programme de test de la borne d'arrivée

/******

This program was produced by the
CodeWizardAVR V1.25.3 Evaluation
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project : Test des afficheurs borne 50 m
Version : 3
Date : 3 juillet 2007
Author : Thierry LEQUEU
Company :
Comments:

Chip type : ATmega8535
Program type : Application
Clock frequency : 16,000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 128

*****/

```
#include <mega8535.h>
#include <stdio.h>
```

```
// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
#include <delay.h>
```

```
#define ENABLE PORTD.7 // sortie.
#define BP1 PIND.6 // entrée.
```

```
// Declare your global variables here
int reception;
int crono;
```

```
#define DIGIT_A 64; { bit b6 }
#define DIGIT_B 128; { bit b7 }
#define DIGIT_C 2; { bit b1 }
#define DIGIT_D 4; { bit b2 }
#define DIGIT_E 8; { bit b3 }
#define DIGIT_F 32; { bit b5 }
```

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle
#define DIGIT_G 16; { bit b4 }
#define DIGIT_P 1; { bit b0 }

```
// Declare your global function here  
unsigned char tampon[20];
```

```
void USART_Transmit( unsigned char data )  
{  
/* Wait for empty transmit buffer */  
while ( !( UCSRA & (0x20)) ); // Test de UDRE bit 5  
/* Put data into buffer, sends the data */  
UDR = data;  
}
```

```
void main(void)
```

```
{  
crono=0;  
// Declare your local variables here
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
```

```
// DDRA=0x00; en entrée
```

```
DDRA=0xFF; // en sortie
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
// DDRB=0x00; en entrée
```

```
DDRB=0xFF; // en sortie
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out Func0=In
```

```
// State7=0 State6=T State5=T State4=T State3=T State2=T State1=0 State0=T
```

```
PORTD=0x00;
```

```
DDRD=0x82; // EN1 BP1 IN IN IN IN TXD RXD
```

```
    // IN7 IN6 IN5 IN4 IN3 IN2 OUT1 IN0
```

```
ENABLE=1; // OC = 1 : pas d'afficheur
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 2000,000 kHz
// Mode: CTC top=OCR1A
// OC1A output: Toggle
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x40;
TCCR1B=0x0A;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x4E; // Base de temps = 2 MHz, soit 0,5 us.
OCR1AL=0x20; // Interruption quand on arrive à 20 000 (0x4E20 soit 10 ms)
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
```


Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle
TIMSK=0x10;

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
```

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;
```

```
// LCD module initialization
lcd_init(16);
```

```
/* switch to writing in Display RAM */
lcd_gotoxy(0,0);
lcd_putsf("Projet arrive.c");
lcd_gotoxy(0,1);
lcd_putsf("Borne 50 m DA");
lcd_gotoxy(0,3);
lcd_putsf("17 decembre 2009");
```

```
PORTB=0xFF;
```

```
// Global enable interrupts
// #asm("sei")
```

```
while (1)
{
    if (BP1==1)
    {
        lcd_gotoxy(0,1);
        lcd_putsf("test bp reussit");
        reception=0xAA;
    }
    else
    {
        lcd_gotoxy(0,1);
        lcd_putsf("test bp echec ");
    }
}
```

```
reception=0x55;  
}  
sprintf(tampon,"TX = 0X%2x",reception);  
lcd_gotoxy(0,2);  
lcd_puts(tampon);  
USART_Transmit(reception);  
delay_ms(500);  
  
}  
}
```


Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

```
while ( !(UCSRA & 0x80) ) // Test de RXC bit7  
;  
/* Get and return received data from buffer */  
return UDR;  
}
```

```
void USART_Transmit( unsigned char data )  
{  
/* Wait for empty transmit buffer */  
while ( !( UCSRA & (0x20) ) ) // Test de UDRE bit 5  
;  
/* Put data into buffer, sends the data */  
UDR = data;  
}
```

```
void main(void)
```

```
{  
// Declare your local variables here
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
```

```
DDRA=0x00;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
DDRB=0x00;
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=Out Func3=In Func2=In Func1=Out Func0=In
```

```
// State7=T State6=T State5=T State4=0 State3=T State2=T State1=0 State0=T
```

```
PORTD=0x00;
```

```
DDRD=0x12; // Bit 4 = sortie = OUT et bit 1 = sortie = TXD
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=FFh
```

```
// OC0 output: Disconnected
```

```
TCCR0=0x00;
```

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle
TCNT0=0x00;
OCR0=0x00;

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```
// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
```

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
```

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

```
// USART Mode: Asynchronous
```

```
// USART Baud rate: 9600
```

```
UCSRA=0x00;
```

```
UCSRB=0x18;
```

```
UCSRC=0x86;
```

```
UBRRH=0x00;
```

```
UBRRL=0x67;
```

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```

```
// LCD module initialization
```

```
lcd_init(16);
```

```
lcd_gotoxy(0,0);
```

```
lcd_putsf("Test RS232");
```

```
while (1)
```

```
{
```

```
    // Place your code here
```

```
    OUT=~OUT;
```

```
    var='0';
```

```
    if (BP1 == 0) { var=0xFE; };
```

```
    if (BP2 == 0) { var='B'; };
```

```
    sprintf(tampon,"BP = %lc",var);
```

```
    lcd_gotoxy(0,1);
```

```
    lcd_puts(tampon);
```

```
    USART_Transmit(var);
```

```
    var=0;
```

```
    var=USART_Receive();
```

```
    sprintf(tampon,"Receive = %lc",var);
```

```
    lcd_gotoxy(0,2);
```

```
    lcd_puts(tampon);
```

```
};
```

```
}
```



```
/* Get and return received data from buffer */
```

```
return UDR;
```

```
}
```

```
void USART_Transmit( unsigned char data )
```

```
{
```

```
/* Wait for empty transmit buffer */
```

```
while ( !( UCSRA & (0x20)) ) // Test de UDRE bit 5
```

```
;
```

```
/* Put data into buffer, sends the data */
```

```
UDR = data;
```

```
}
```

```
void main(void)
```

```
{
```

```
// Declare your local variables here
```

```
// Input/Output Ports initialization
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
```

```
DDRA=0x00;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
DDRB=0x00;
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=Out Func3=In Func2=In Func1=Out Func0=In
```

```
// State7=T State6=T State5=T State4=0 State3=T State2=T State1=0 State0=T
```

```
PORTD=0x00;
```

```
DDRD=0x12; // Bit 4 = sortie = OUT et bit 1 = sortie = TXD
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=FFh
```

```
// OC0 output: Disconnected
```

```
TCCR0=0x00;
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```



```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
```

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
```

```
// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("Test RS232");
```

```
while (1)
{
    // Place your code here
    OUT=~OUT;
    var='0';
    if (BP1 == 0) { var=0xFE; };
    if (BP2 == 0) { var='B'; };
    sprintf(tampon, "BP = %1c", var);
    lcd_gotoxy(0,1);
    lcd_puts(tampon);

    USART_Transmit(var);
    sprintf(tampon, "transmit = %1c", var);
    lcd_gotoxy(0,2);
    lcd_puts(tampon);

};
}
```


Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

```
#define ENABLE PORTD.7
```

```
int SERIE;
```

```
unsigned char RECEPTION;
```

```
int CHRONO;
```

```
flash const unsigned char valeur_constant[] =
```

```
{0xEE,0x82,0xDC,0xD6,0xB2,0x76,0x7E,0xC2,0xFE,0xF6,  
 0,1,2,4,8,16,32,64,128};
```

```
unsigned char tampon[20],seconde,temps,dizaine,unite,dixieme,centieme;
```

```
// Declare your global function here
```

```
unsigned char USART_Receive( void )
```

```
{
```

```
/* Wait for data to be received */
```

```
while ( !(UCSRA & 0x80) ) // Test de RXC bit7
```

```
;
```

```
/* Get and return received data from buffer */
```

```
return UDR;
```

```
}
```

```
void USART_Transmit( unsigned char data )
```

```
{
```

```
/* Wait for empty transmit buffer */
```

```
while ( !( UCSRA & (0x20)) ) // Test de UDRE bit 5
```

```
;
```

```
/* Put data into buffer, sends the data */
```

```
UDR = data;
```

```
}
```

```
// Timer 1 output compare A interrupt service routine
```

```
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
```

```
{
```

```
if(CHRONO==1)
```

```
{
```

```
temps++;
```

```
if (temps>=100)
```

```
{
```

```
temps=0;
```

```
seconde++;
```

```
if (seconde>=60)
```

```
{
```

```
seconde=0;
```

```
};
```

```
};
```

```
}
```

```
if(SERIE==1)
```

```
{
```

```
SERIE=0;
```

```
    seconde=0;
    temps=0;
}
}

// Affiche une valeur sur 8 bits à l'adresse de l'afficheur :
void afficheur1(unsigned char adresse,unsigned char caractere, unsigned char point)
{
    if (point == 1) {
        PORTA=(valeur_constant[caractere] | 0x01); // Un caractère avec le point !
    }
    else {
        PORTA=valeur_constant[caractere]; // Un caractère
    }

    PORTB=(0b00010000 | adresse_constant[adresse & 0x0F]); // PB7 PB6 PB5 CS A0 A1 A2
    A3
    PORTB.4=1;
    PORTB.4=0; // CS = 0
    PORTB.4=1;
    PORTA=0x00;
}

void main(void)
{
    CHRONO=0;
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTA=0x00;
    // DDRA=0x00; en entrée
    DDRA=0xFF; // en sortie

    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTB=0xFF;
    // DDRB=0x00; en entrée
    DDRB=0xFF; // en sortie

    // Port C initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
    PORTC=0x00;
    DDRC=0x00;

    // Port D initialization
```

Université François Rabelais
 Institut Universitaire de Technologie
 Département Génie électrique et Informatique Industrielle
 // Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out Func0=In
 // State7=0 State6=T State5=T State4=T State3=T State2=T State1=0 State0=T
 PORTD=0x80;
 DDRD=0x82; // EN1 BPx IN IN IN IN TXD RXD
 // IN7 IN6 IN5 IN4 IN3 IN2 OUT1 IN0
 ENABLE=1; // OC = 1 : pas d'afficheur

// Timer/Counter 0 initialization
 // Clock source: System Clock
 // Clock value: Timer 0 Stopped
 // Mode: Normal top=FFh
 // OC0 output: Disconnected
 TCCR0=0x00;
 TCNT0=0x00;
 OCR0=0x00;

// Timer/Counter 1 initialization
 // Clock source: System Clock
 // Clock value: 2000,000 kHz
 // Mode: CTC top=OCR1A
 // OC1A output: Toggle
 // OC1B output: Discon.
 // Noise Canceler: Off
 // Input Capture on Falling Edge
 // Timer 1 Overflow Interrupt: Off
 // Input Capture Interrupt: Off
 // Compare A Match Interrupt: On
 // Compare B Match Interrupt: Off
 TCCR1A=0x40;
 TCCR1B=0x0A;
 TCNT1H=0x00;
 TCNT1L=0x00;
 ICR1H=0x00;
 ICR1L=0x00;
 OCR1AH=0x4E; // Base de temps = 2 MHz, soit 0,5 us.
 OCR1AL=0x20; // Interruption quand on arrive à 20 000 (0x4E20 soit 10 ms)
 OCR1BH=0x00;
 OCR1BL=0x00;

// Timer/Counter 2 initialization
 // Clock source: System Clock
 // Clock value: Timer 2 Stopped
 // Mode: Normal top=FFh
 // OC2 output: Disconnected
 ASSR=0x00;
 TCCR2=0x00;
 TCNT2=0x00;
 OCR2=0x00;

// External Interrupt(s) initialization

Université François Rabelais
Institut Universitaire de Technologie
Département Génie électrique et Informatique Industrielle

```
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

// LCD module initialization
lcd_init(16);

/* switch to writing in Display RAM */
lcd_gotoxy(0,0);
lcd_putsf("Projet noeud.c");

PORTB=0xFF;
seconde=0;
temps=0;

ENABLE=0;
afficheur1(0,11,0); // Un caractère
afficheur1(1,11,0); // Un caractère
afficheur1(2,11,0); // Un caractère
afficheur1(3,11,0); // Un caractère

// Global enable interrupts
#asm("sei")

RECEPTION=0xF0;
```

```
while (1)
```

```
{
    sprintf(tampon,"Temps = %2d:%2d",seconde,temps);
    lcd_gotoxy(0,2);
    lcd_puts(tampon);

    RECEPTION=USART_Receive();
    sprintf(tampon,"UDR=%2x UCSRA=%2x",RECEPTION,UCSRA);
    lcd_gotoxy(0,1);
    lcd_puts(tampon);

    if(RECEPTION==0xFE) // BP appuyer sur la carte d'émission.
    {
        CHRONO=0;
        lcd_gotoxy(0,3);
        lcd_putsf(" pause  !!");
        delay_ms(1500);
        SERIE=1;
    }
    CHRONO=1;
    lcd_gotoxy(0,3);
    lcd_putsf(" depart  !!");
    dizaine = seconde/10;
    unite = seconde-dizaine*10;
    dixieme = temps/10;
    centieme = temps-dixieme*10;

    // Afficheur d'adresse 0
    if (dizaine == 0) {
        afficheur1(0,0,0);
    }
    else {
        afficheur1(0,dizaine,0); // Caractère des dizaines.
    }
    // Afficheur d'adresse 1
    afficheur1(1,unite,1); // Caractère des unités.
    // Afficheur d'adresse 2
    afficheur1(2,dixieme,0); // Caractère des dixièmes.
    // Afficheur d'adresse 3
    afficheur1(3,centieme,0); // Caractère des centièmes.
    }
}
```