

# CYBERNETIQUE EN NORD

## Description de la carte Electronical\_ Main\_Board\_32



## Table des matières

1 Description de la carte Electronical Main Board.....	3
1.1.1 Présentation.....	3
1.1.2 Cahier des charges.....	3
2 Schéma fonctionnel.....	5
3 Schéma Structurel.....	6
4 PCB.....	7
5 Vues 3D.....	10
6 Liste des commandes, nomenclature.....	11
7 Validation de la carte.....	12
Alimentations.....	12
ICD3 PIC.....	13
RS232 port 2 + RESET.....	14
TEST DES LEDS.....	16
Test des SWITCH.....	17
Test du GO.....	17
AFFICHEUR.....	18
PWM.....	19
INPUT CAN 1-2-3-4.....	20
I2C.....	21
8 Change Logs :.....	22
V1-00 = V0-10 : Version de base.....	22
V1-10.....	22
V1.20.....	23

# 1 Description de la carte Electronical Main Board

## 1.1.1 Présentation

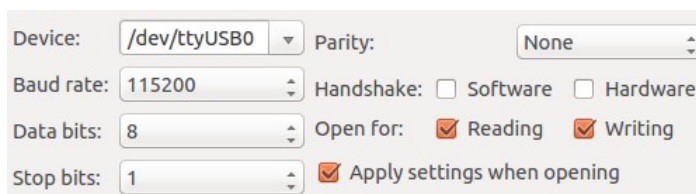
Cette carte a été conçue de le but de pouvoir contrôler un système mobile en mode autonome ou radio commandé. Il est indispensable que cette carte puisse contrôler différents périphériques avec des temps de réaction très courts et maîtrisable.

Cette carte doit permettre de configurer le fonctionnement du robot à l'aide de 14 switches.

Elle aura une interface visuelle permettant d'afficher des informations sur un afficheur LCD 4\*16.

Le Pic de cette carte sera le maître des Bus I2C.

Les ports séries seront configurés ainsi :

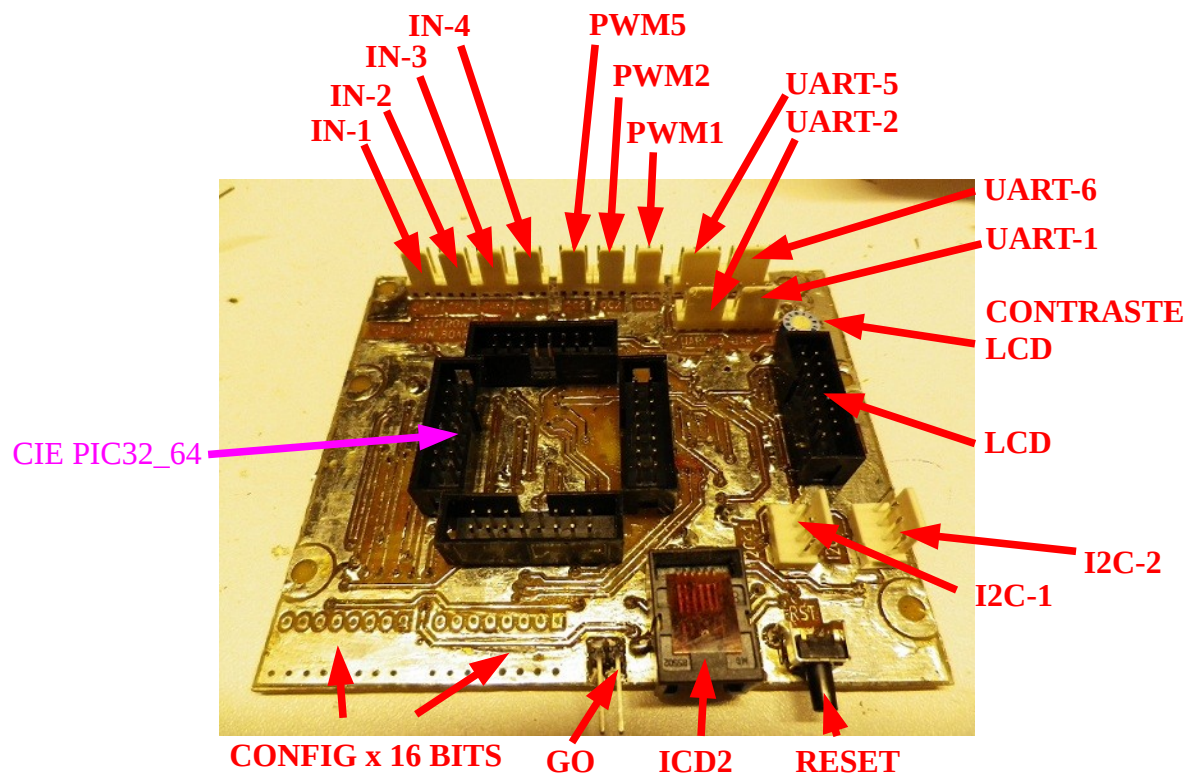


Device:	/dev/ttyUSB0	Parity:	None
Baud rate:	115200	Handshake:	<input type="checkbox"/> Software <input type="checkbox"/> Hardware
Data bits:	8	Open for:	<input checked="" type="checkbox"/> Reading <input checked="" type="checkbox"/> Writing
Stop bits:	1	<input checked="" type="checkbox"/> Apply settings when opening	

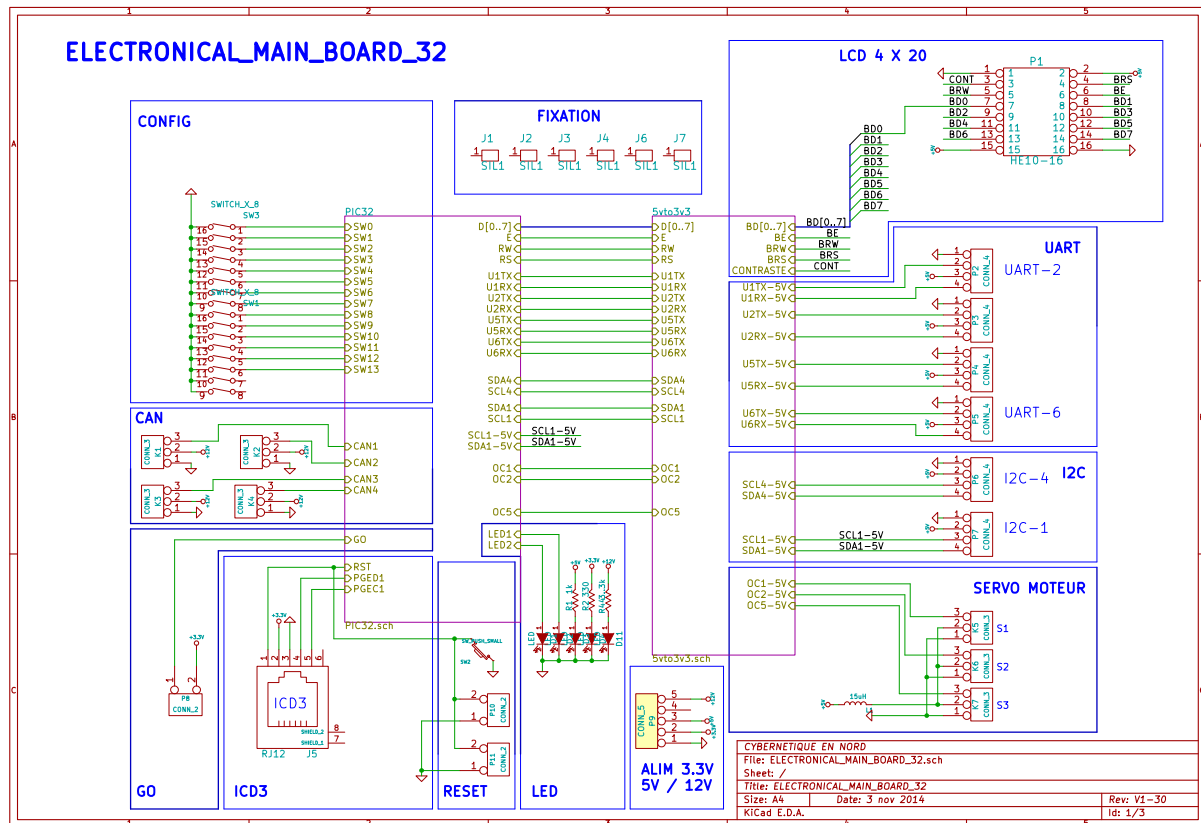
## 1.1.2 Cahier des charges

Cette carte doit :

- Définir une config sur 14bits
- Etre fixer sur 6 colonettes
- Commander un afficheur à texte de 4\*20 caractère avec le réglage du contraste sur la carte par potentiomètre
- Disposer de 2 Bus I2C
- Disposer de 4 ports RS232
- Disposer d'un cavalier pour le «GO »
- Commander 3 ports PWM
- Commander 4 ports entrées
- 1 connecteur ICD3
- 1 bouton poussoir RESET
- Disposer d'une EEPROM
- Disposer d'un capteur de température
- Disposer d'une horloge RTC sauvegardée

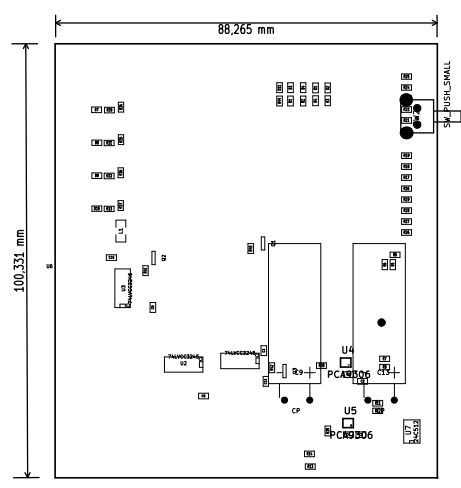
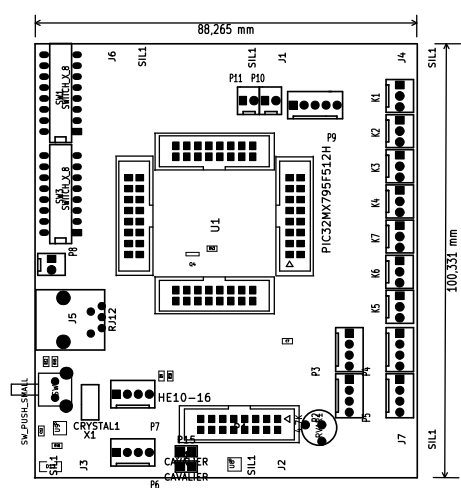
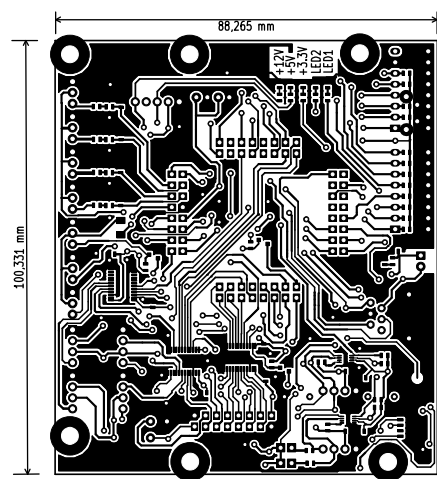
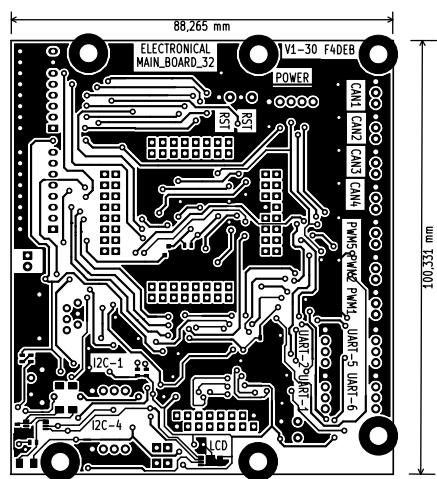


## 2 Schéma fonctionnel





4 PCB

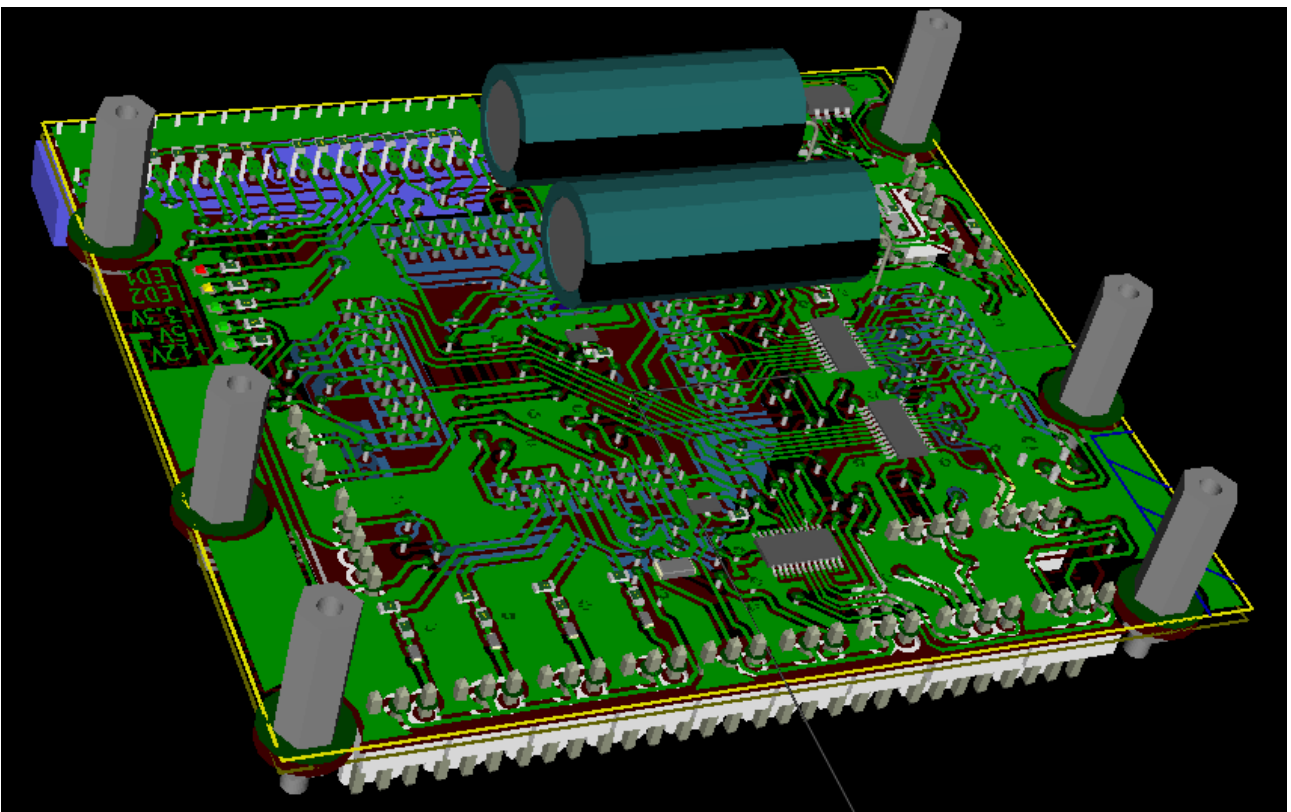
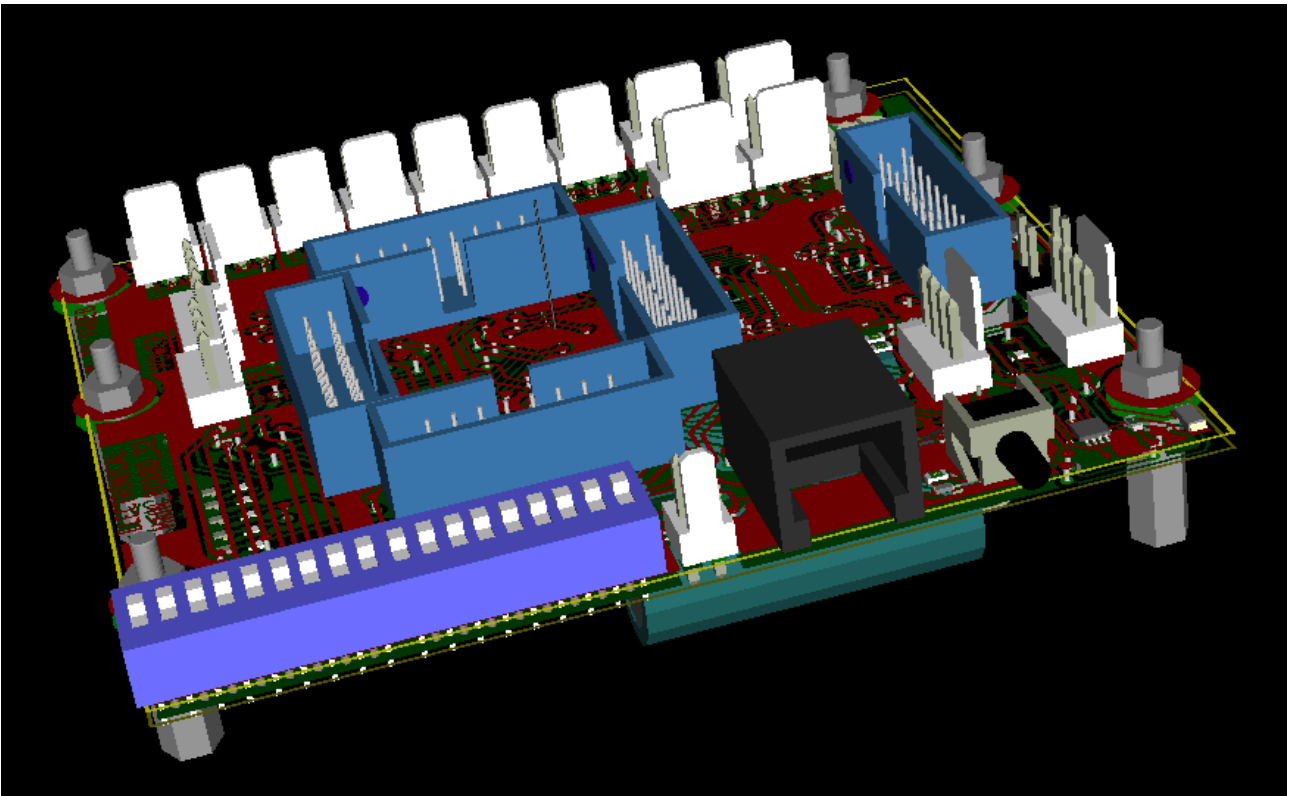








## 5 Vues 3D



## 6 Liste des commandes, nomenclature

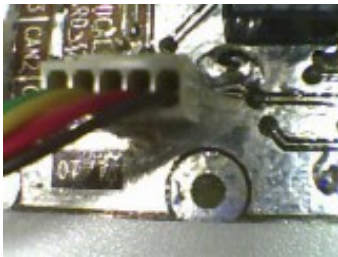
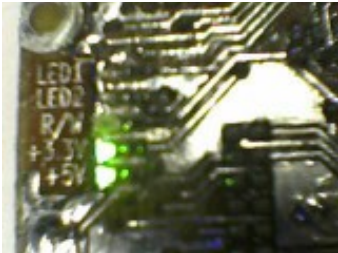

LABEL	Valeur	Code commande	Fournisseur	Prix unitaire	QTY	Prix
C1,C2,C3,C4,C5,C6,C7,C8,C10,C11	100n 0603		Farnell		10	
D3,D4,D7,D8,D9,D10,D11	LED 0603 Vert		Farnell		6	
D2	LED 0603 Rouge		Farnell		1	
D5	LED 0603 Bleu		Farnell		1	
D1	LED 0603 Orange		Farnell		1	
J1,J2,J3,J4,J6,J7	Colonette		Farnell		6	
J5	RJ12		Farnell		1	
K1,K2,K3,K4,K5,K6,K7	KK-3		Farnell		7	
L1	15µH 1206		Farnell		1	
P1	HE10-16-D		Farnell		1	
P2,P3,P4,P5,P6,P7	KK-4		Farnell		6	
P8	KK-2		Farnell		1	
P9	KK-5		Farnell		1	
			Farnell			
R1,R9,R10,R13,R14	1k		Farnell		5	
R2,R3,R4,R6	330		Farnell		4	
R5,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25,R26,R27,R28,R29	10k		Farnell	Farnell	15	
R7,R8,R11,R12,R15	2,1k		Farnell		5	
R30,R31,R32,R33	3,2k		Farnell		4	
R34,R35, R36,R37K	1,5k		Farnell		4	

## 7 Validation de la carte

NE PAS CONNECTER LES ALIMENTATIONS

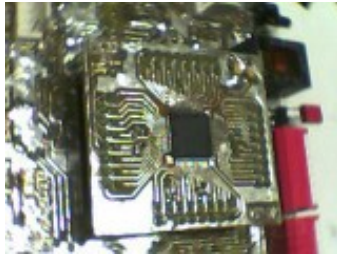
NE PAS METTRE LE CIE PIC32\_64

### Alimentations

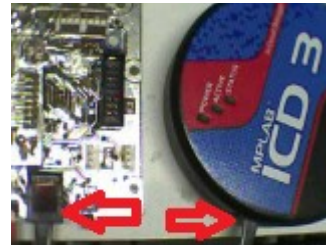
<ul style="list-style-type: none"><li>– Contrôle visuel.</li><li>– Test de continuité , vérifier qu'il n'y ai pas de cour-circuit.</li><li>– Test de la connexion +3,3V et GND.</li><li>– Test de la connexion +5VD et GND</li></ul>	<ul style="list-style-type: none"><li>– Brancher P9</li></ul> 
<ul style="list-style-type: none"><li>– Verifier l'allumage des LEDs 5V et 3,3V</li></ul> 	<ul style="list-style-type: none"><li>– Débrancher P9</li></ul> 

## ICD3 PIC

- Insérer la carte PIC32\_64



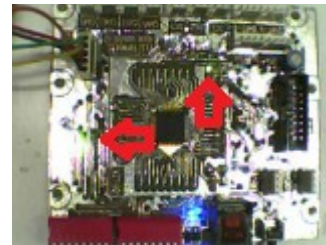
- Connecter le module ICD3 sur J5



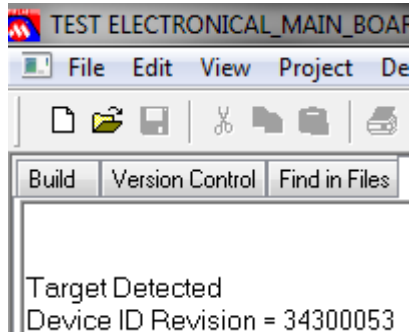
- Brancher P9



- On voit par transparence les leds +5V et +3v3 de la carte ELECTRICAL\_MAIN\_BOARD et la led 3v3 de la carte PIC32\_64

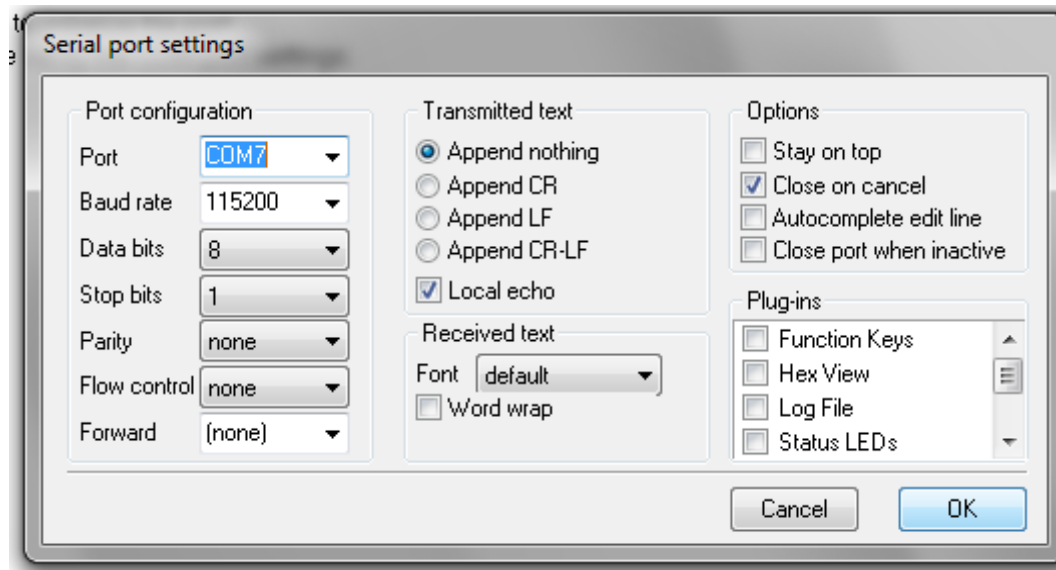


- Le message suivant s'affiche dans la page MPLAB

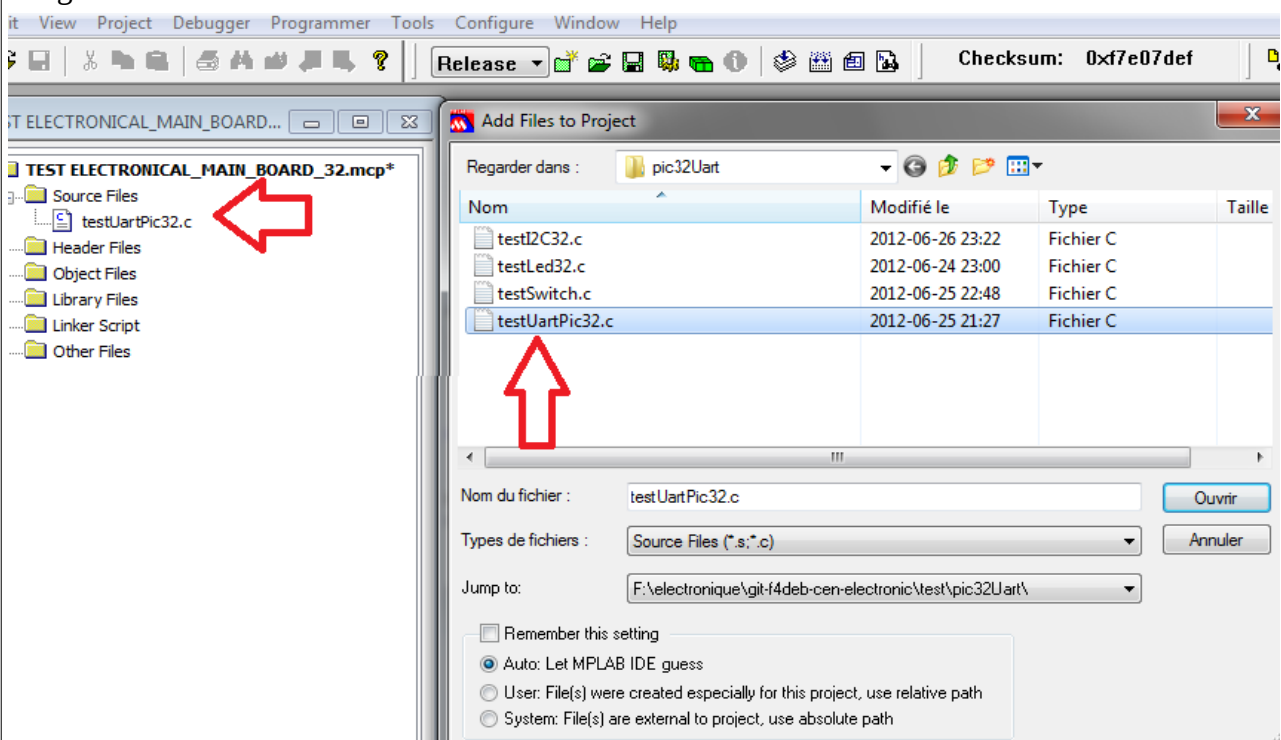


## RS232 port 2 + RESET

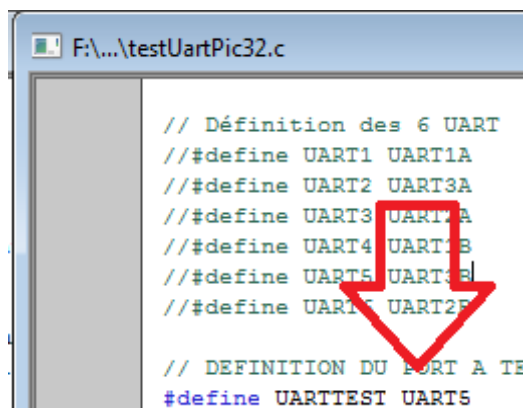
- Configurer le logiciel Termite pour la liaison série comme ci-dessous



- Charger dans le pic le programme : **TEST ELECTRICAL\_MAIN\_BOARD\_32.mcp** et intégrer le fichier **testUartPic32.c** dans Source Files



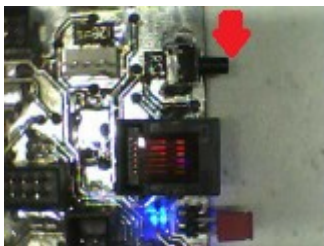
Modifier le numéro de l'UART à tester, compiler et programmer le PIC.



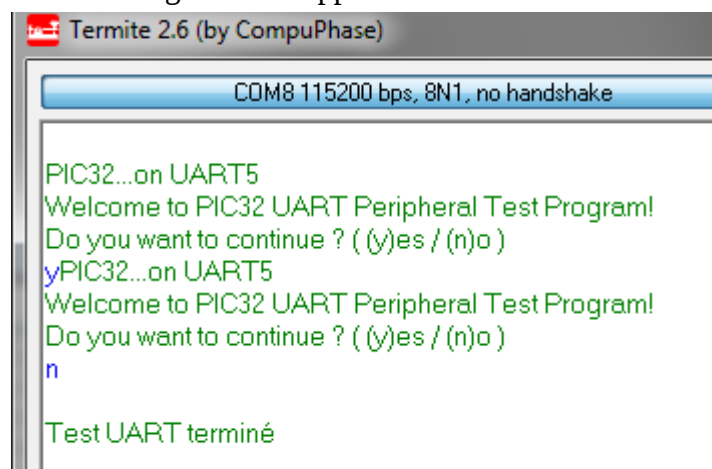
```
// Définition des 6 UART
// #define UART1 UART1A
// #define UART2 UART3A
// #define UART3 UART1A
// #define UART4 UART1B
// #define UART5 UART3B
// #define UART6 UART2B

// DEFINITION DU PORT A TESTER
#define UARTTEST UART5
```

- Appuyer sur le bouton RESET



Le message suivant apparaît alors dans le terminal



```
Termite 2.6 (by CompuPhase)
COM8 115200 bps, 8N1, no handshake

PIC32...on UART5
Welcome to PIC32 UART Peripheral Test Program!
Do you want to continue ? ( y )es / ( n )o
yPIC32...on UART5
Welcome to PIC32 UART Peripheral Test Program!
Do you want to continue ? ( y )es / ( n )o
n
Test UART terminé
```

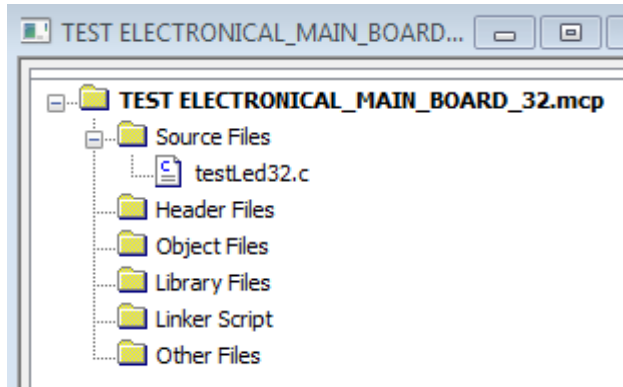
- Effectuer les deux dernières étapes en modifiant le numéro de l'UART afin de tester les quatre UART de la carte.

- UART1
- UART2
- UART5
- UART6

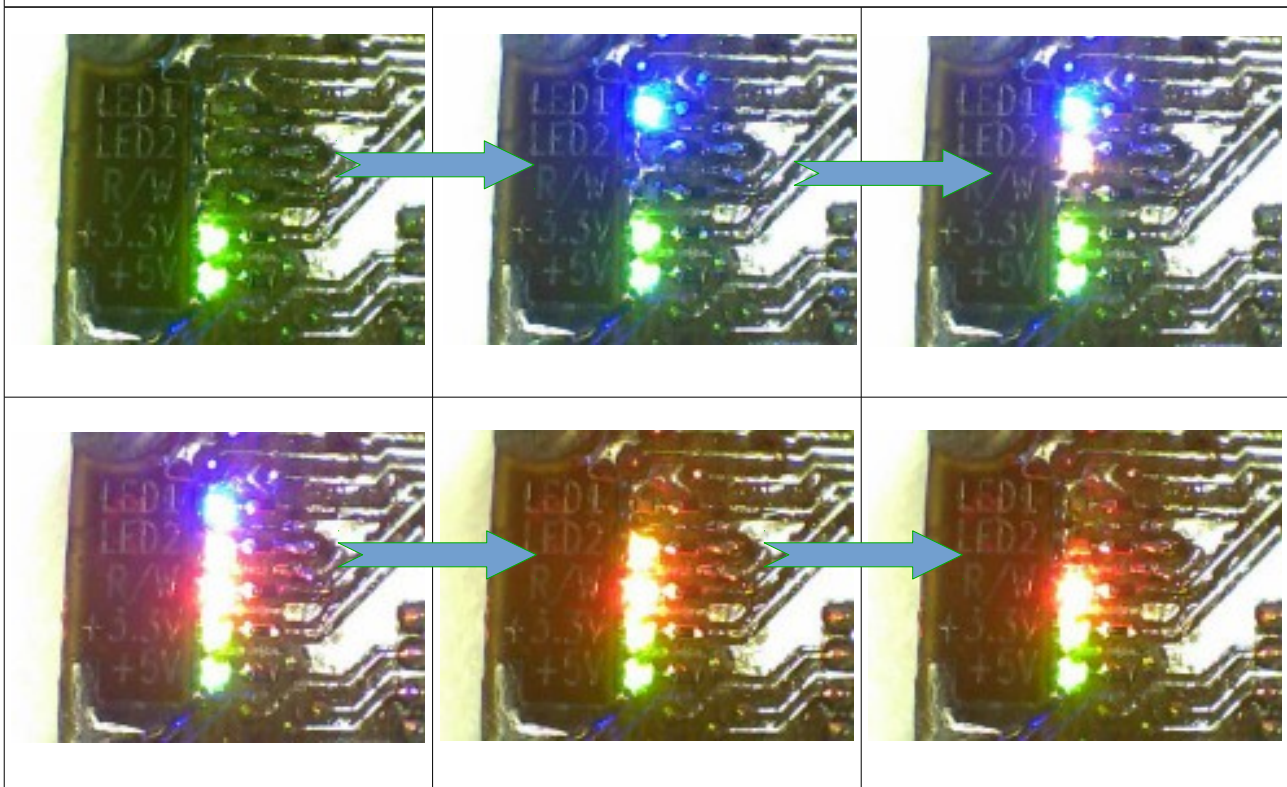


## TEST DES LEDS

- Changer le fichier source, compiler, et programmer le Pic32



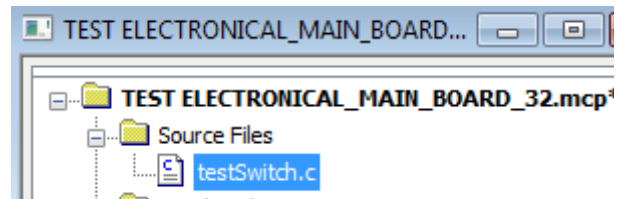
-La séquence des leds est la suivante, elle comporte 6 phases en boucle avec une seconde entre chaque phase, cela permet de contrôler l'activation du quartz à 8Mhz.



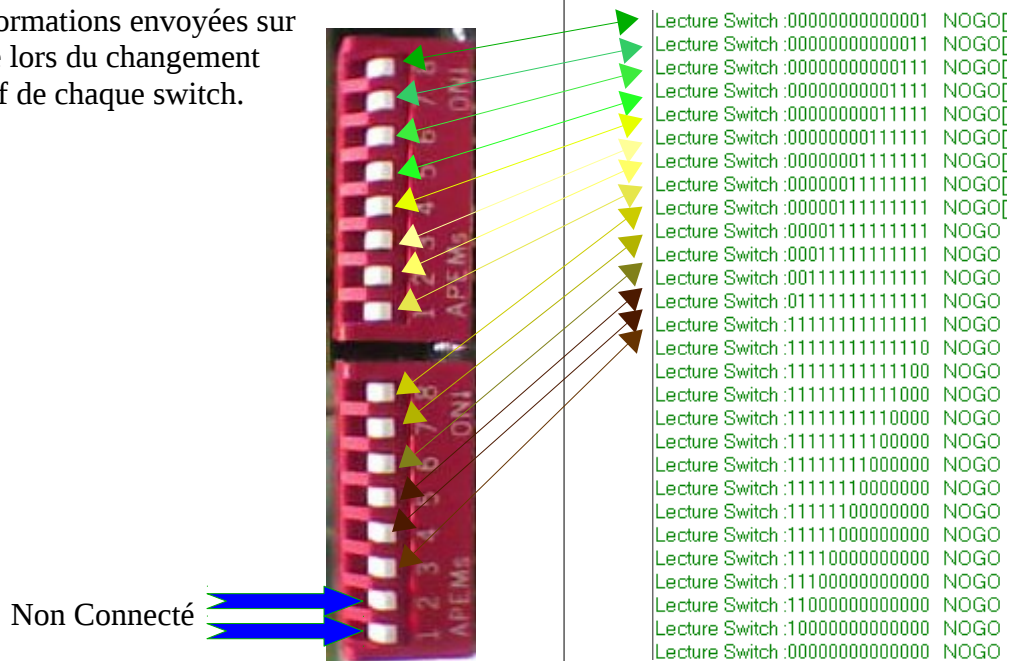


## Test des SWITCH

- Changer le fichier source, compiler, et programmer le Pic32.

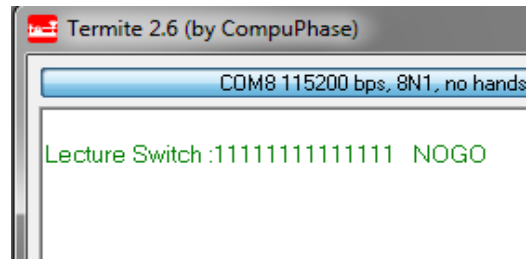
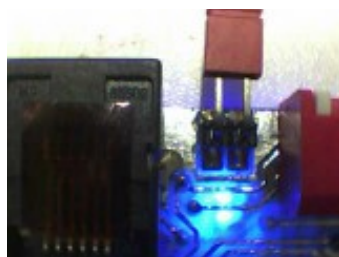


- Liste des informations envoyées sur la liaison série lors du changement d'état successif de chaque switch.

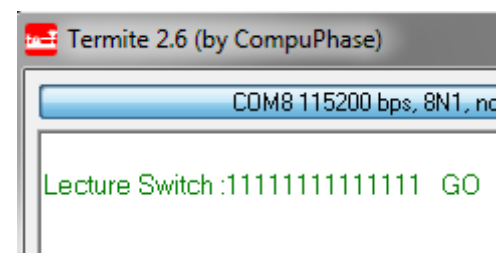


## Test du GO

- NOGO :  
Cavalier en position

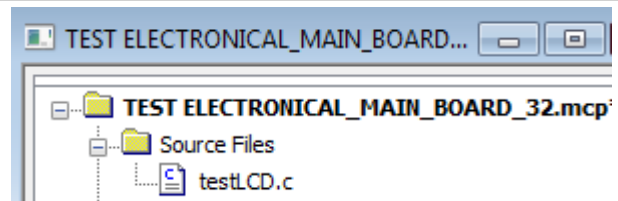


- Go :  
Cavalier retiré



## AFFICHEUR

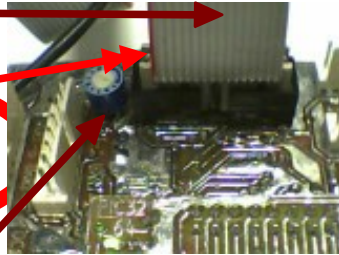
- Changer le fichier source, compiler, et programmer le Pic32.



- Brancher l'afficheur

ATTENTION AU SENS DE LA NAPPE

- Regler RVAR1 pour avoir un bon contraste.

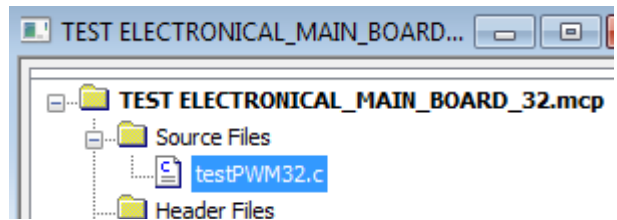


- Voici le résultat sur l'afficheur

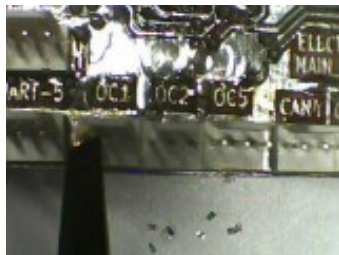


## PWM

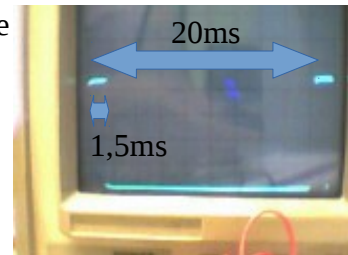
- Changer le fichier source, compiler, et programmer le Pic32



Positionner la sonde scope sur OC1



Voici le résultat sur le scope.  
Amplitude du signal 5V.



Connecter le servomoteur.  
Masse côté du chiffre



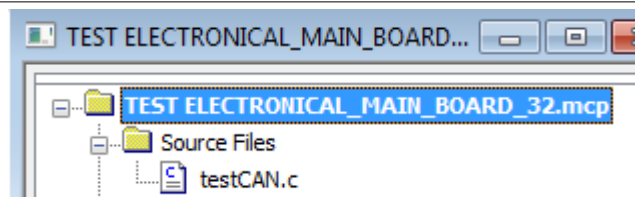
Le servomoteur se trouve en position central



Effectuer la même opération sur les 3 ports PWM (OC1-OC2-OC5)

## INPUT CAN 1-2-3-4

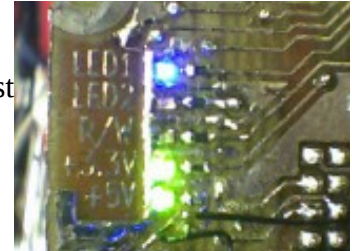
- Changer le fichier source, compiler, et programmer le Pic32



- Insérer le cavalier sur CAN1 comme sur le croquis



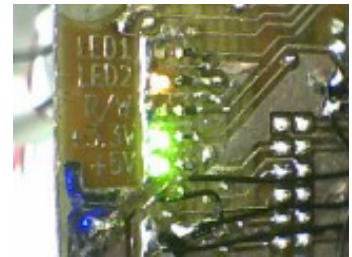
- La led bleu «LED1» s'allume lorsque le cavalier est en place



- Insérer le cavalier sur CAN2 comme sur le croquis



- La led orange «LED2» s'allume lorsque le cavalier est en place



- Insérer le cavalier sur CAN3 comme sur le croquis



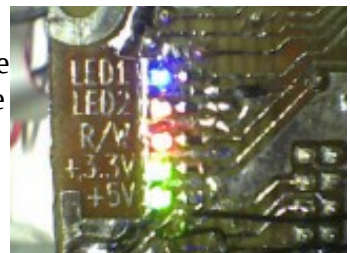
- La led orange «R/W» s'allume lorsque le cavalier est en place



- Insérer le cavalier sur CAN4 comme sur le croquis



- Les 3 leds s'allument lorsque le cavalier est en place



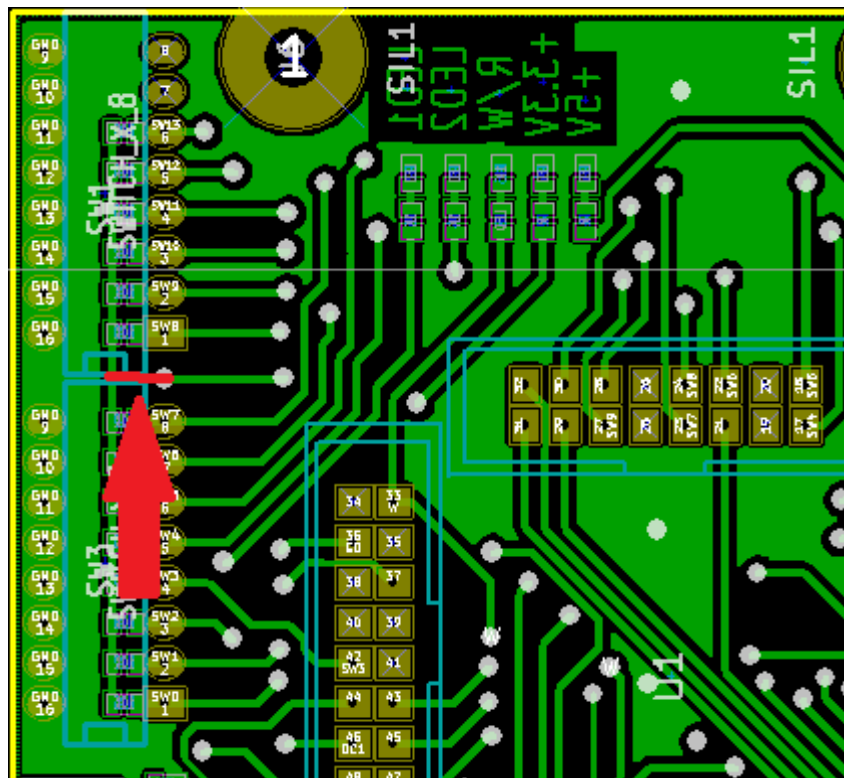


## 8 Change Logs :

**V1-00 = V0-10 : Version de base**

### **V1-10**

- Empreinte RJ12:épaissir les pastilles
- Empreinte HE10 pastille carrée trou de 1mm
- Empreinte KK-4 et -3 trou de 1 mm
- NET switch blindage GND
- NET RJ12 fixation GND
- Connecteur 3,3V et 12V a rajouter
- I2C rajout de résistance de 200k sur EN et Vref2 sur PCA9306
- Déplacer via sous potar LCD
- Changer empreinte potar LCD
- Ajout led + diviseur de tension pour le port entrée en 12V
- LED1 devient RC14
- LED2 devient RD11
- GO devient RG3
- SW13 devient RG2



## V1.20

- Espacer les connecteurs KK
- Ajout Eeprom
- Ajout capteur de température
- Ajout horloge RTC sauvegardée
- Passer résistance de tirage à 5V pour les OC
- Ajout LED 12V