



Le 23-11-2023 V1

Description de la carte principale

Table des matières

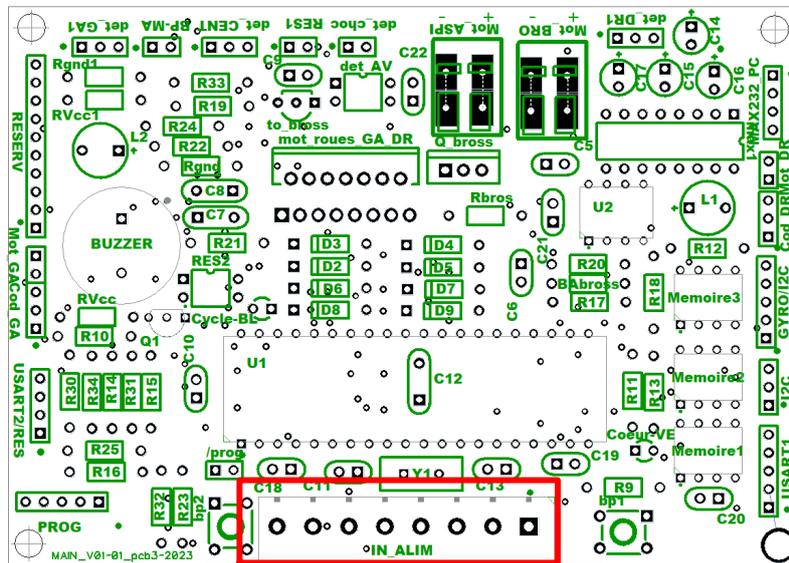
Description de la carte principale	1
Description de la carte :.....	2
Les connexions avec la carte alimentation	2
Connexions essentiels	2
Connexions complètes	2
Selfs pour réduire les parasites.....	3
Caractéristiques et Configuration du Pic	4
Blocs explicatifs.....	5
principe et fonctionnement.....	6
les entrées Analogiques.....	6
Entrées TOR (tout ou rien).....	7
réserves	9
Bouton poussoirs	10
Les sorties	11
Moteurs et leds	11
Commande du moteurs brosses et aspiration.....	13
Sortie buzzer	14
La com et les réseaux	15
Gestion des mémoires.....	16
I2C et Gyroscope	17
Communication avec Max 232 ou boitier additionnel	18
Option directement sur le pc en passant par un db9.....	19
Option 2 en passant par un boitier RS232 intermédiaire.....	20
Schéma des deux options	20
Connexion sur usart2	21



Le 23-11-2023 V1

Description de la carte :

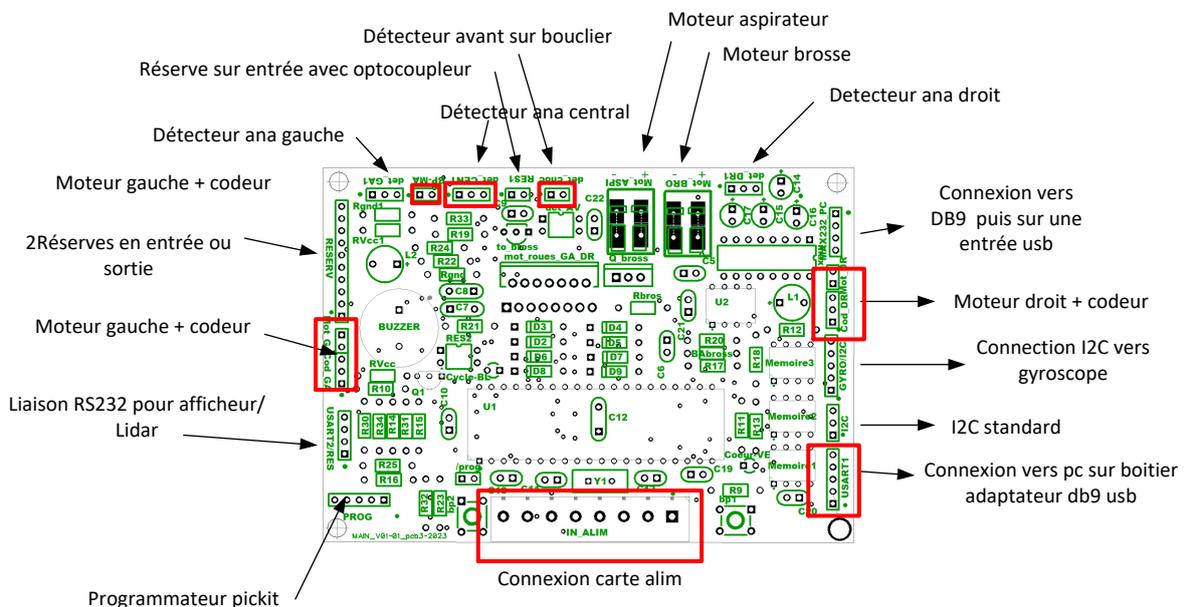
Les connexions avec la carte alimentation



Bat Vcc 3.3v 0v 5v 0v 12v 0v

Connexions avec alimentation

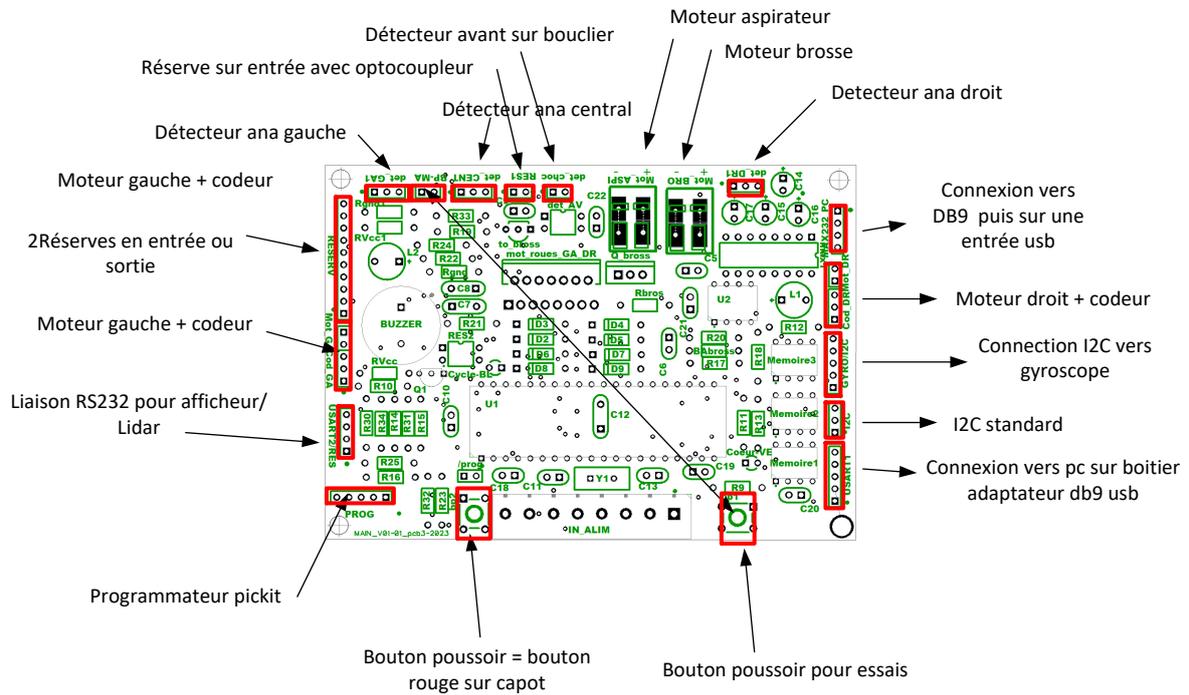
Connexions essentielles



Connexions complètes

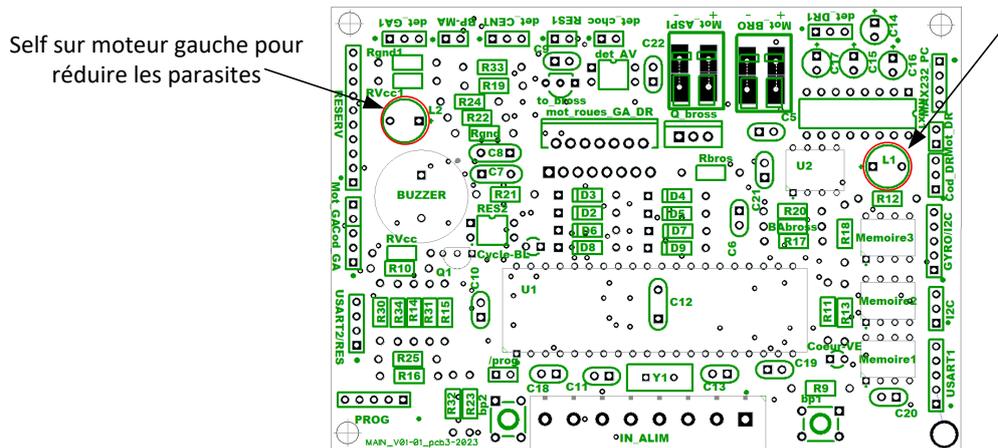


Le 23-11-2023 V1



Selfs pour réduire les parasites

Self sur moteur droit pour réduire les parasites

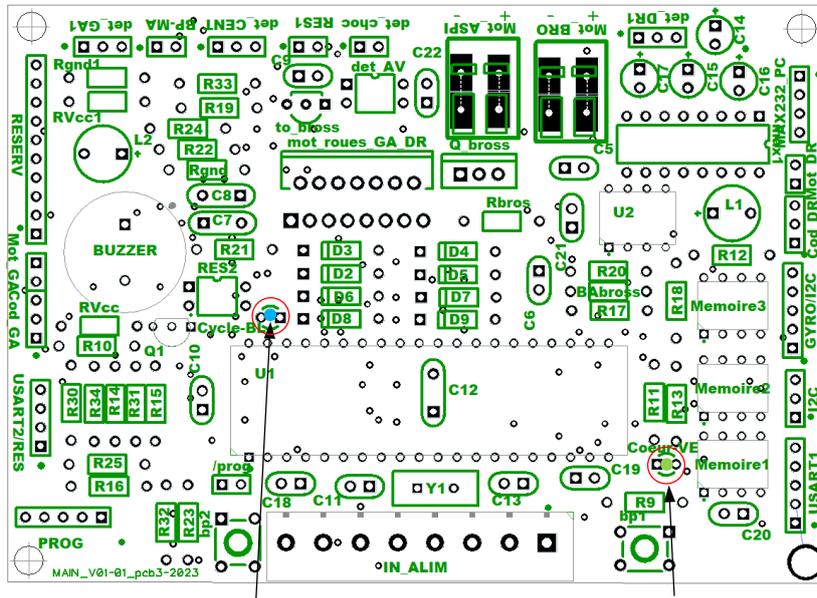


Connexion avec le pickit pour programmation



Le 23-11-2023 V1

Les boutons poussoirs



Led bleue indication de fonctionnement

LED verte cœur du robot

Caractéristiques et Configuration du Pic

- 2 usart
- 4 entrées analogiques (10 bits)
- 2 interruptions sur les entrées int1 et int0 (codeur roue)
- 1 réseau I²C master
- 1 réseau SPI master
- 2 PWM (hacheur moteur)
- 5 entrées
- 12 sorties
- 2 réserves
- Un oscillateur 20mhz

128K Bytes Program Flash Memory

3728 Bytes Data SRAM Memory

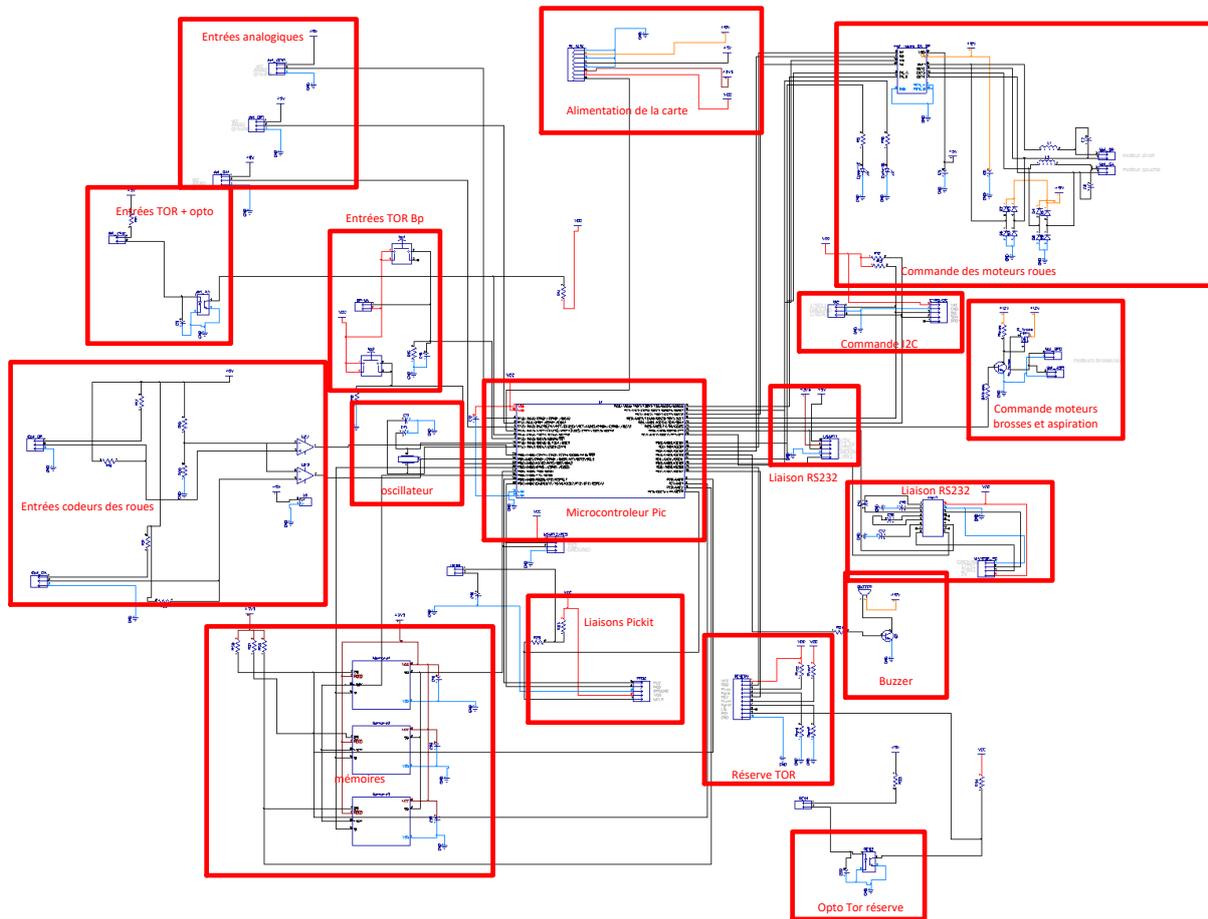
1024 Bytes Data EEPROM

Tension d'utilisation 2.3V to 5.5V (PIC18F27/47K40)



Le 23-11-2023 V1

Blocs explicatifs

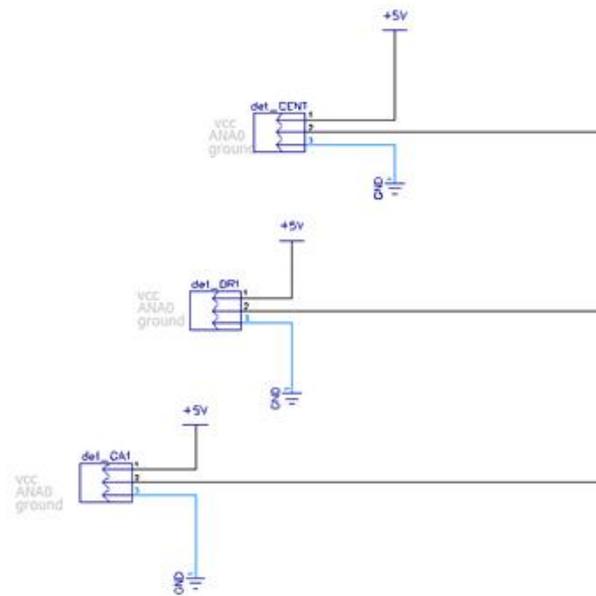




Le 23-11-2023 V1

principe et fonctionnement

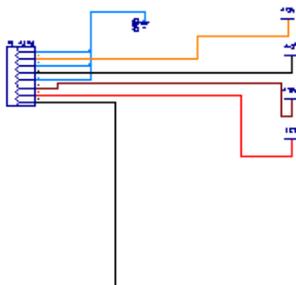
les entrées Analogiques



Entrée analogique 1

Entrée analogique 2

Entrée analogique 3



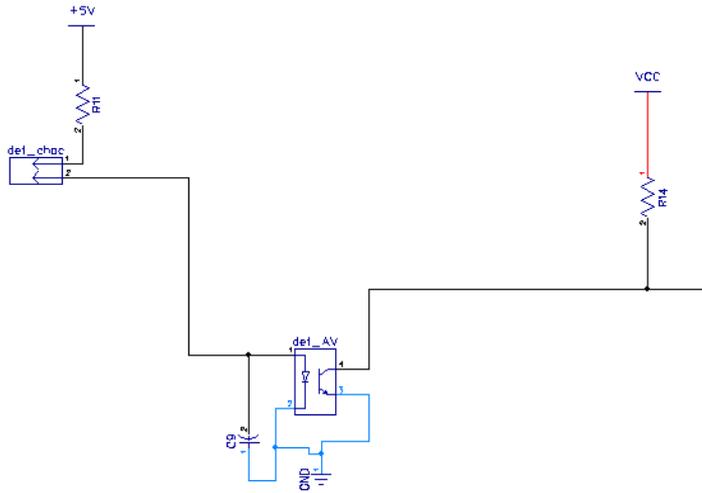
Entrée analogique 3
Tension de la batterie

```
#define det_cent_ANA      PORTAbits.RA0 // ;RA0 P2 ANA entrée sur détecteur optique central
#define det_DR_ANA      PORTAbits.RA1 // ;RA1 P3 ANA
#define det_GA_ANA      PORTAbits.RA2 // ;RA2 P4 ANA
#define ANA_BATT        PORTAbits.RA3 // ;RA3 P5 tension batterie
```

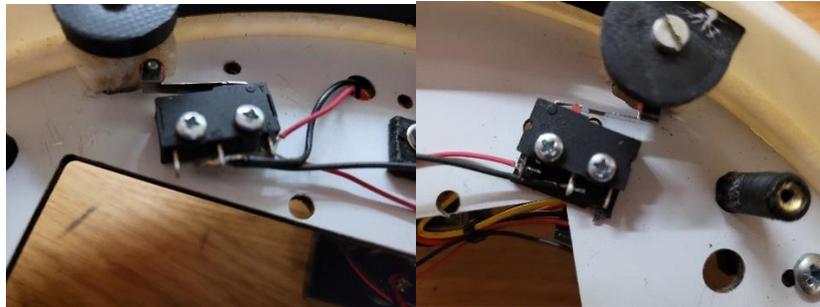


Le 23-11-2023 V1

Entrées TOR (tout ou rien)



RA4 entrée des deux détecteurs sur le
pare choc avant

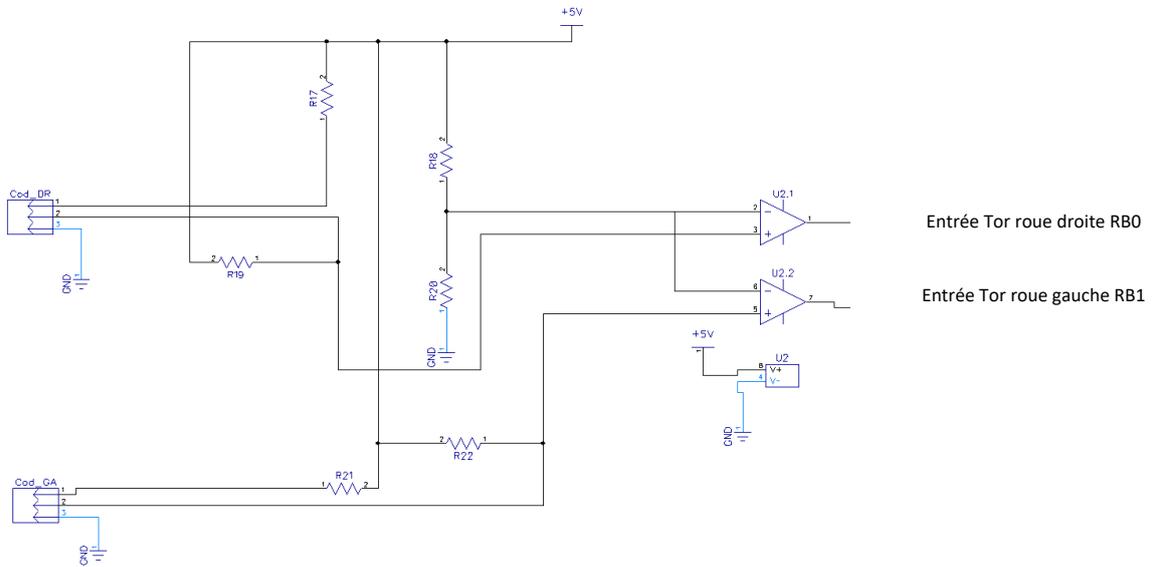


```
#define entree_det_AV_ID
```

```
PORTAbits.RA4 // ;RA4 P6
```

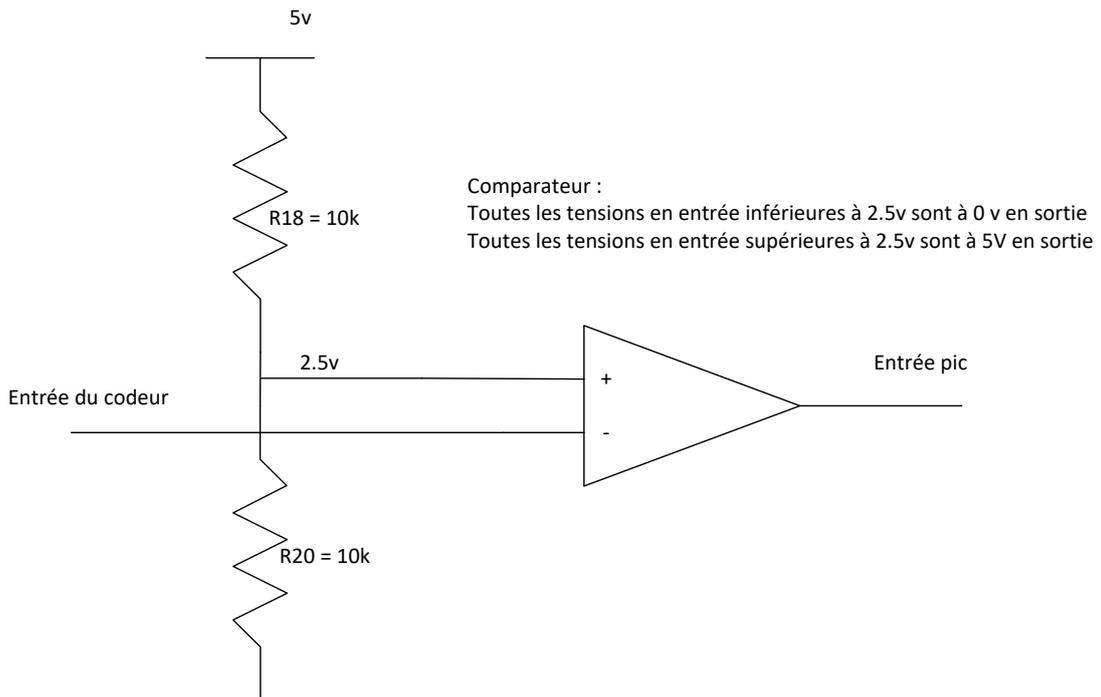


Le 23-11-2023 V1



Entrée Tor roue droite RB0

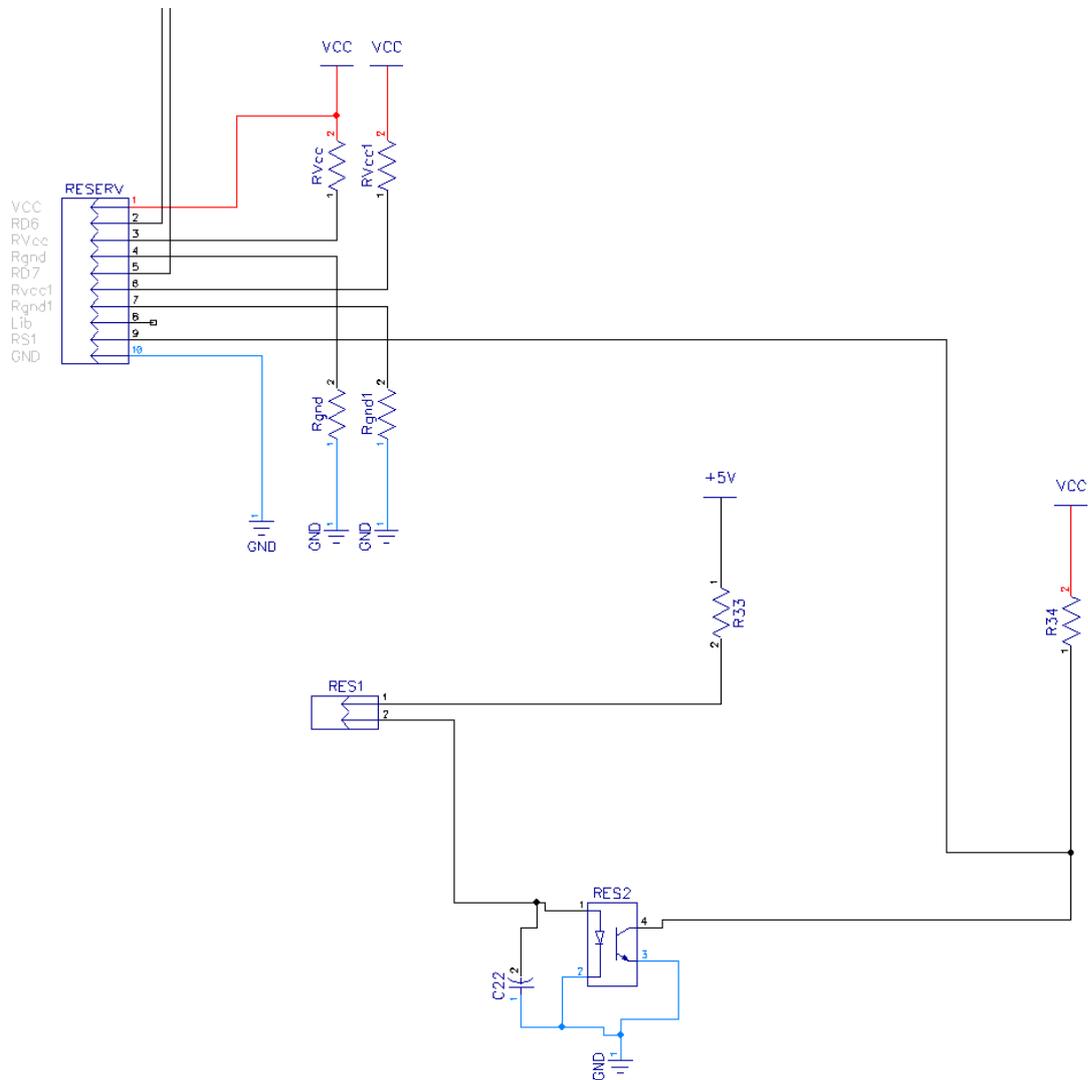
Entrée Tor roue gauche RB1





Le 23-11-2023 V1

réserves



Réserves sur RD6 et RD7, il y a des ponts possibles pour mettre l'optocoupleur sur une des entrées.

```
#define reserve_D6
```

```
PORTDbits.RD6 // patte 29
```

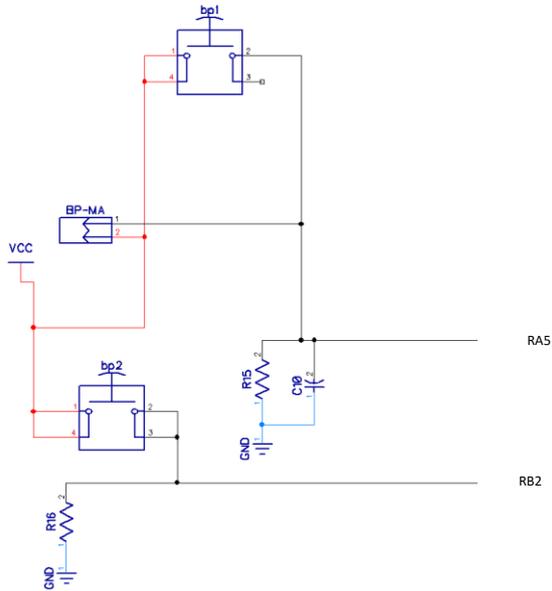
```
#define reserve_D7
```

```
LATDbits.LATD7 // patte 30
```



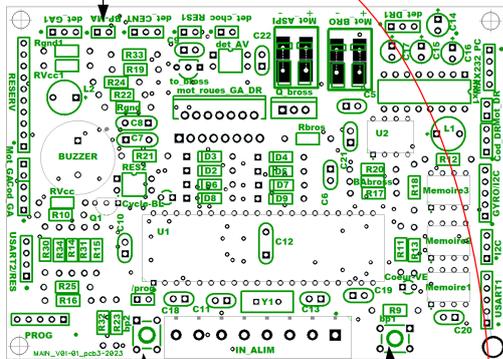
Le 23-11-2023 V1

Bouton poussoirs



Bouton poussoir pour essais

Sur la même entrée



Bouton poussoir = bouton rouge sur capot

Bouton poussoir pour essais

```
#define entree_bp1          PORTAbits.RA5 // ;RA5 P7
```

```
#define entree_bp2          PORTBbits.RB2 // ;RB2 P35
```

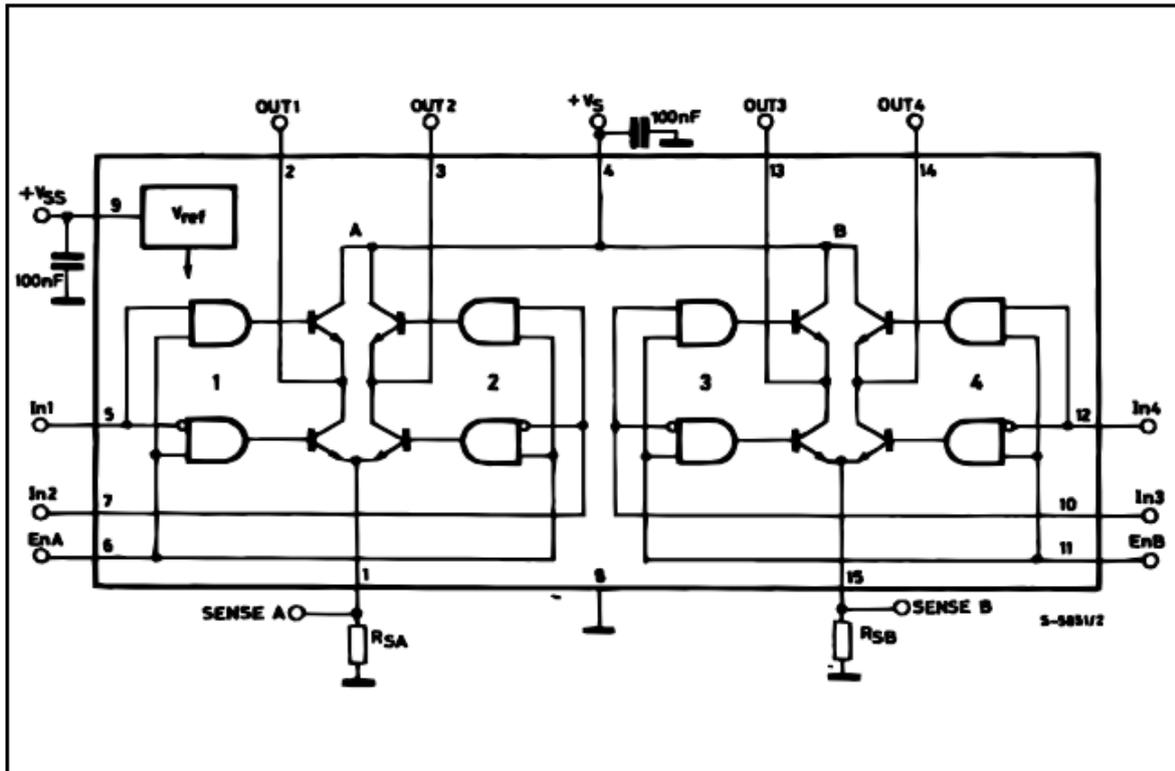


Le 23-11-2023 V1

Les sorties

Moteurs et leds

BLOCK DIAGRAM



Sur le block diagram on reconnaît les ponts en H il y en a deux A et B donc moteur Droit et gauche .

On notera qu'il y a un système de protection pour que les transistors qui sont sur le même côté ne puissent pas faire de courts circuits.

Il a été prévu un « sense » qui permet avec la résistance de mesurer le courant dans le moteur nous n'utilisons pas cette fonction.

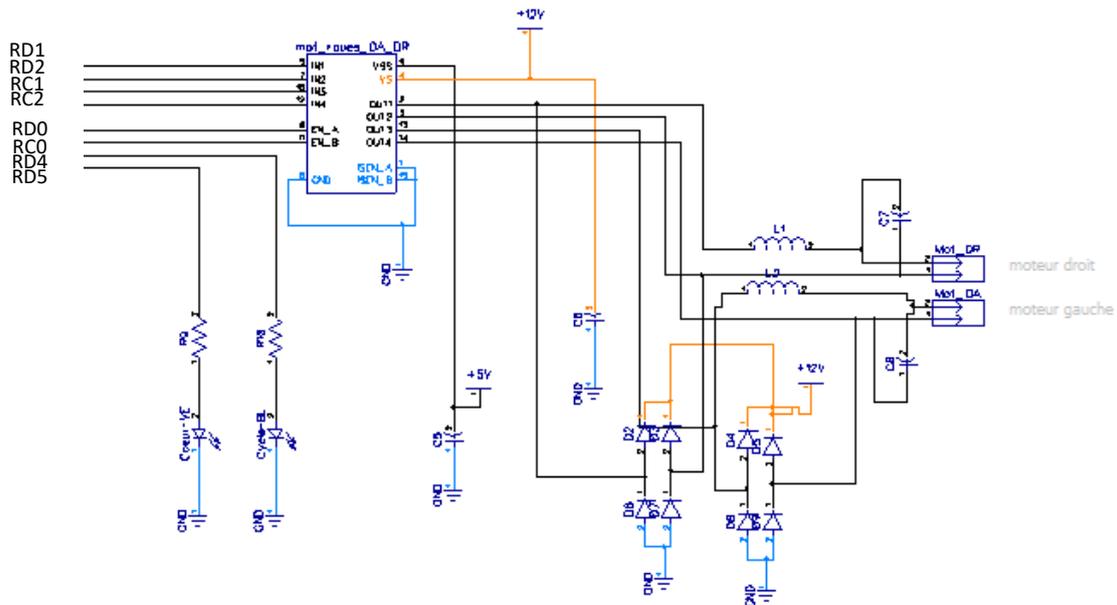
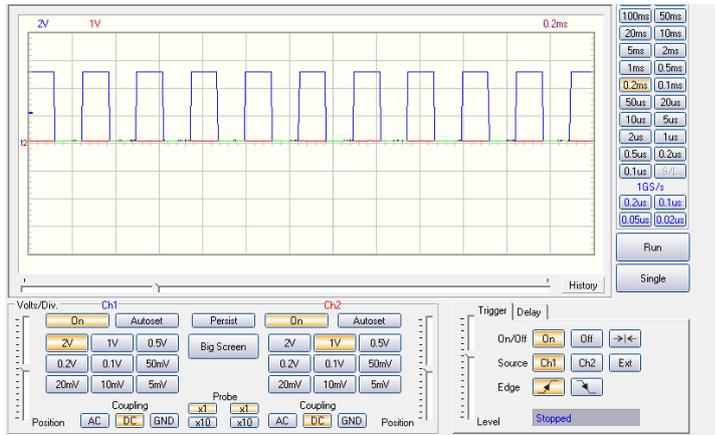
Les entrées pour le moteur de droite sont IN1 et IN2, la commande en PWM sur ENA .

Les entrées pour le moteur de gauche sont IN3 et IN4, la commande en PWM sur ENB .

Les commandes moteurs IN1,2,3,4 sont en TOR, les commandes des PWM introduisent un rapport cyclique voici un exemple (50%) ce qui signifie que le moteur sera alimenté en 6v.



Le 23-11-2023 V1



```

#define sortie_mot_enable_B_PW4 LATbits.LATC0 // patte 15 commande du pwm gauche
#define sortie_mot_IN3_recul_GA LATbits.LATC1 // patte 16
#define sortie_mot_IN4_avance_GA LATbits.LATC2 // patte 17

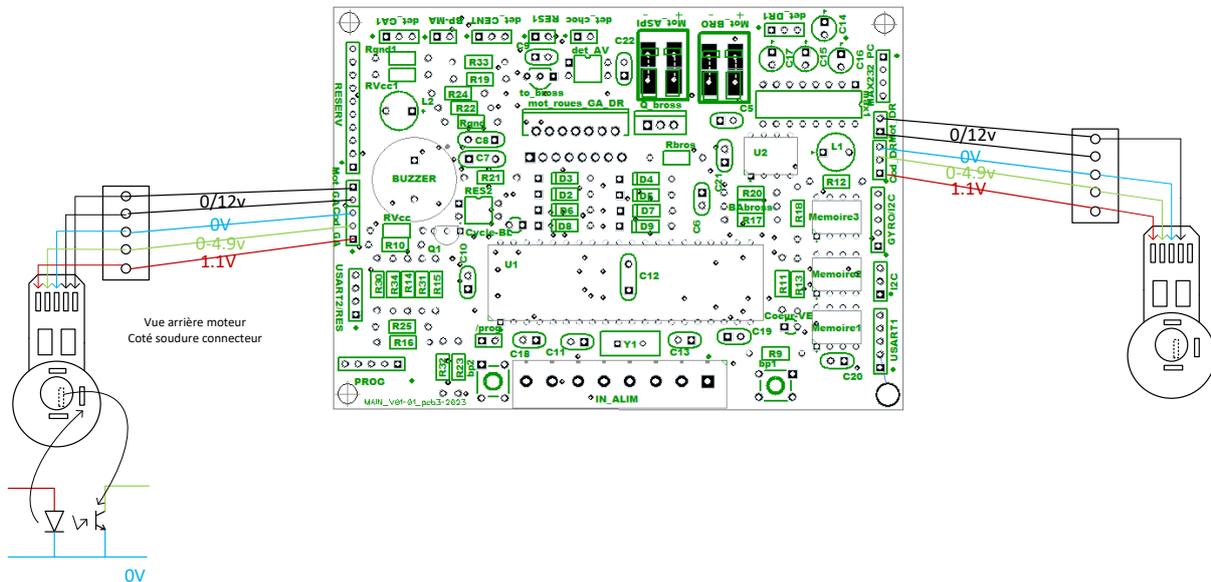
#define sortie_mot_enable_PW3 LATDbits.LATD0 /// patte 19 commande du pwm dr
#define sortie_mot_IN1_recul_DR LATDbits.LATD1 /// patte 20
#define sortie_mot_IN2_avance_DR LATDbits.LATD2 /// patte 21

#define sortie_led_coeur_ve LATDbits.LATD4 // patte 27
#define sortie_led_bleue LATDbits.LATD5 // patte 28

```

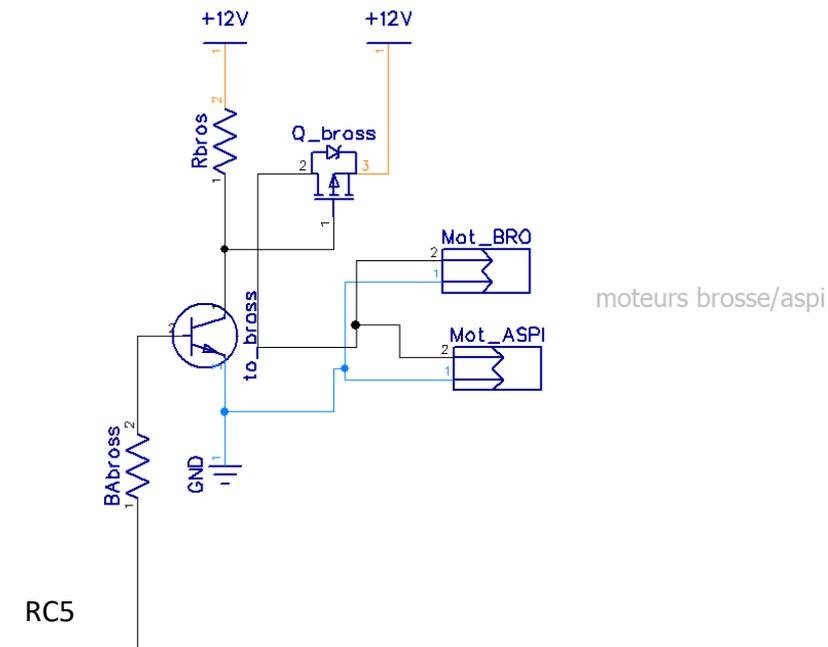


Le 23-11-2023 V1



ATTENTION ne pas mettre du 5V en direct sur la diode

Commande du moteurs brosses et aspiration



#define sortie_mot_brosse

LATCbits.LATC5

// patte 24



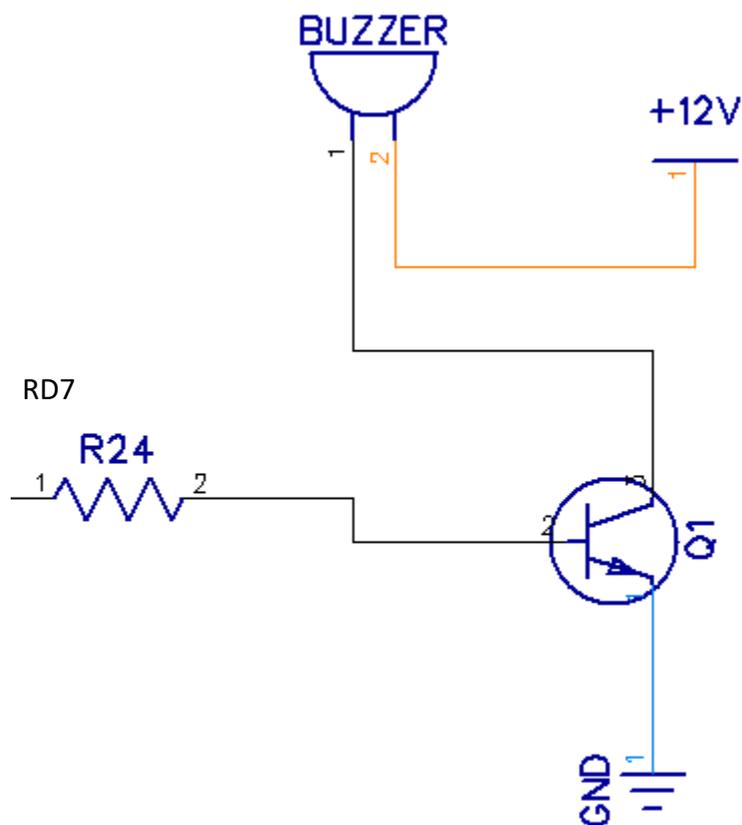
Le 23-11-2023 V1

Une tension appliquée sur RC5 met en conduction le transistor, (2N3904) un courant passe dans sa base. La tension Vce est de 6v rbross est de 1.2k, le mosfet est piloté en tension par VGS(grille source), le courant sur la grille est nulle, sur la sortie (P2) du mosfet on a 12V.

IRF 9530 canal P Vgs pour les mosfets, qui est la tension appliquée à la grille, vis-à-vis de la source.

Id pour les mosfets (courant traversant le drain) maximum est de 12A (ce que nous n'atteindrons jamais).

Sortie buzzer



#define sortie_buzzer

LATDbits.LATD3

// patte 22



Le 23-11-2023 V1

```
#define sortie_led_coeur_ve    LATDbits.LATD4    // patte 27
```

Transistor passant : Une tension de 5v est appliquée sur R24 le transistor se met à conduire car la tension de base est de 0.7v la tension 5-0.7v est sur la résistance R24, la tension vce est proche de 0 soit 0.03v dans ce cas le buzzer sonne.

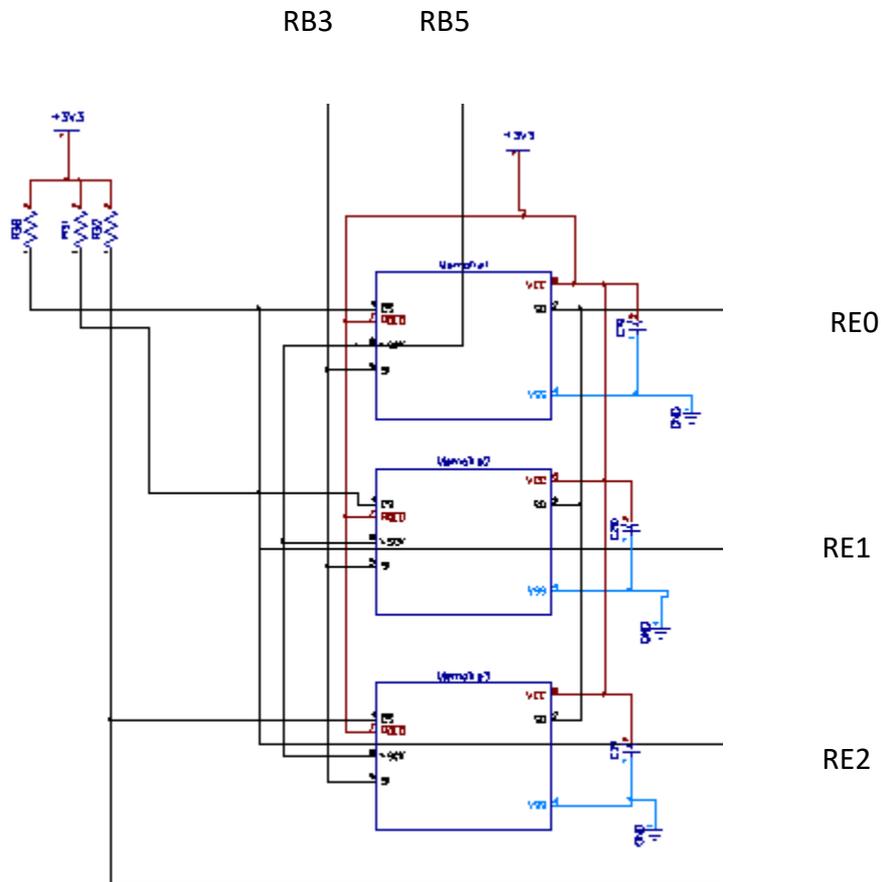
Transistor bloqué : pas de tension sur la base du transistor, le transistor se comporte comme un isolant, la tension sur son collecteur (3) est de 12v le buzzer n'émet aucun son.

La com et les réseaux



Le 23-11-2023 V1

Gestion des mémoires



▼ Hardware Settings	
Serial Protocol:	SPI
Mode:	Master
SPI Mode:	SPI Mode 0
Input Data Sampled At:	Middle
Clock Source Selection:	FOSC/4
Actual Clock Frequency(Hz):	1250000.00

Chaque boîtier est sélectionné par une sortie TOR,

```
#define sortie_pasCS_mem1      LATEbits.LATE0      //patte 8
```



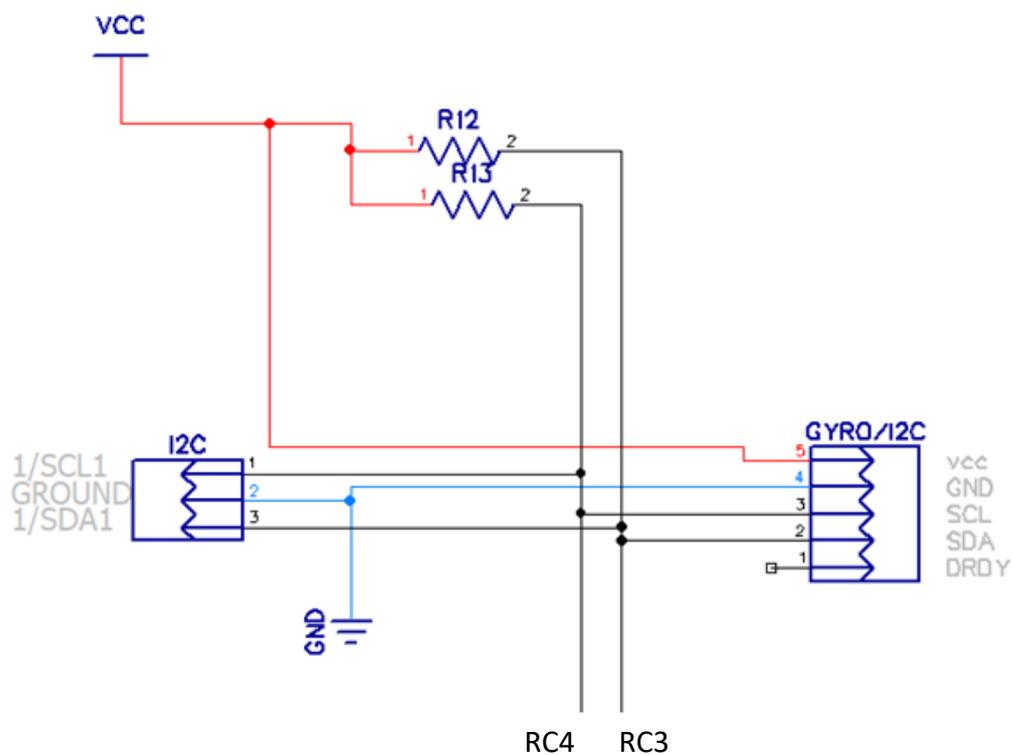
Le 23-11-2023 V1

```
#define sortie_pasCS_mem2    LATEbits.LATE1    //patte 9
#define sortie_pasCS_mem3    LATEbits.LATE2    //patte 10
```

le réseau est un SPI avec données et clock sur

```
#define SO_mem_SDI           LATBbits.LATB4 //    ;RB4 P37
#define sck_mem_SCK2        PORTBbits.RB5 //    ;RB5 P38
```

I2C et Gyroscope

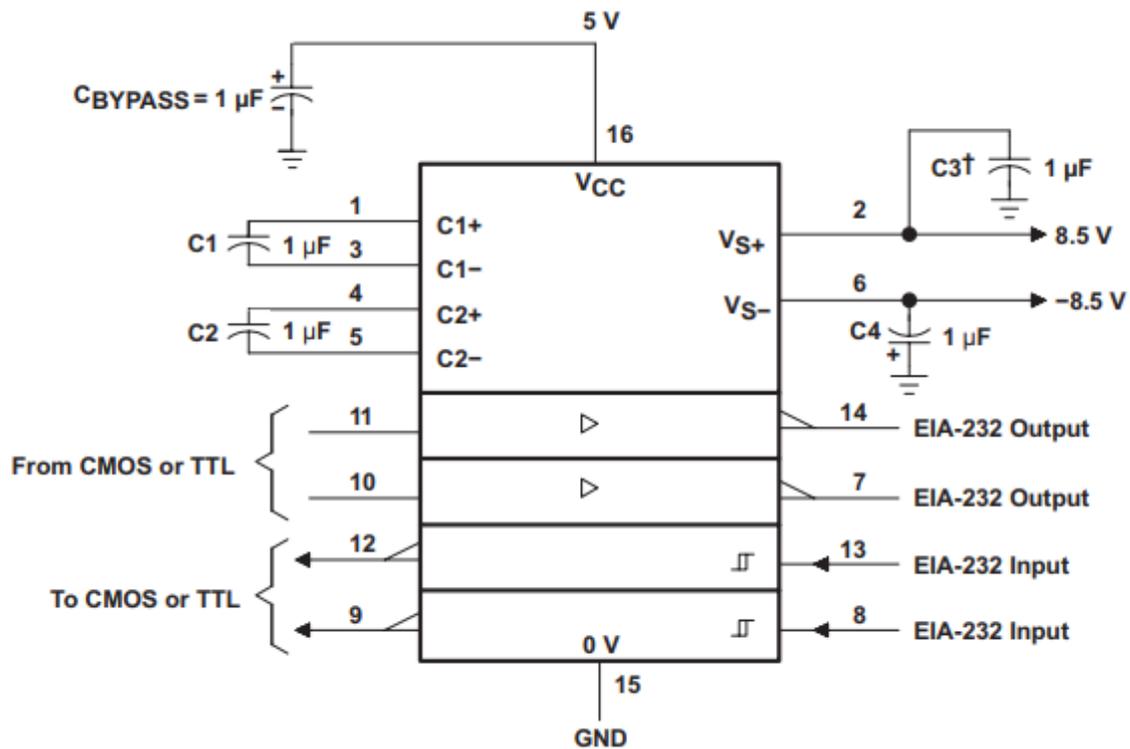


▼ Software Settings	
Interrupt Driven:	<input checked="" type="checkbox"/>
▼ Hardware Settings	
Serial Protocol:	I2C
Mode:	Master
I2C Clock Frequency(Hz):	4883 ≤ 100000 ≤ 312500
Actual Clock Frequency(Hz):	104166.66



Le 23-11-2023 V1

Communication avec Max 232 ou boitier additionnel



Vcc est de 4.5v et un maximum de 5.5V.

Les condensateurs sont céramique ou tentale.

Vitesse :

Flux maximum est de 120 kbps (kilo bit par seconde).

La valeur la plus courante est celle que nous utiliserons

9600 bauds	9600 bits/s	104.167 μs	1200 octets/s	960 octets/s	1.042 ms
------------	-------------	------------	---------------	--------------	----------

Néanmoins nous pourrions aussi aller sur l'autre vitesse commune qui est de :

115200 bauds	115200 bits/s	8.681 μs	14400 octets/s	11520 octets/s	86.806 μs
--------------	---------------	----------	----------------	----------------	-----------

Dans ce cas ne pas oublier de changer dans l'émetteur et le récepteur.(le Pic et le pc par exemple).

Deux options pour communiquer en RS232 (envoi de caractères) au pc ou un autre périphérique par exemple un Bluetooth, lidar

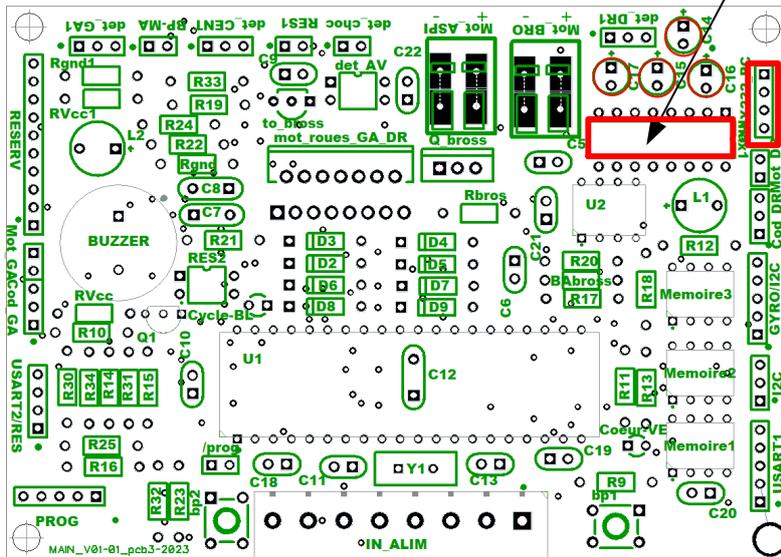
Soit en passant par le rs232 directement donc sur usart1 ou en passant par le boitier (option Max232) associé au condensateur pour relever la tension et envoyé sur un connecteur Max323_pc.



Le 23-11-2023 V1

Option directement sur le pc en passant par un db9

Max 232

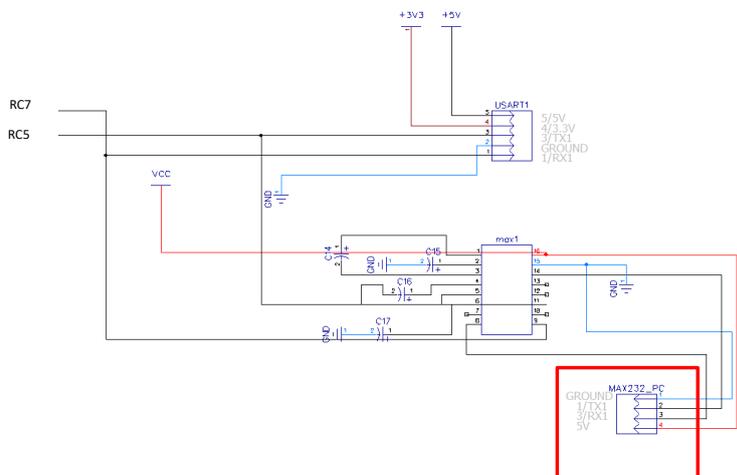


Connecteur de sortie allant directement sur PC en RS232 DB9

```
#define com_pc_mess_TX1          PORTCbits.RC6          // patte 25
```

```
#define com_pc_mess_RX1          PORTCbits.RC7          // patte 26
```

Connexion pour envoyer en RS232 les trames de caractères au PC





Le 23-11-2023 V1

Option 2 en passant par un boîtier RS232 intermédiaire

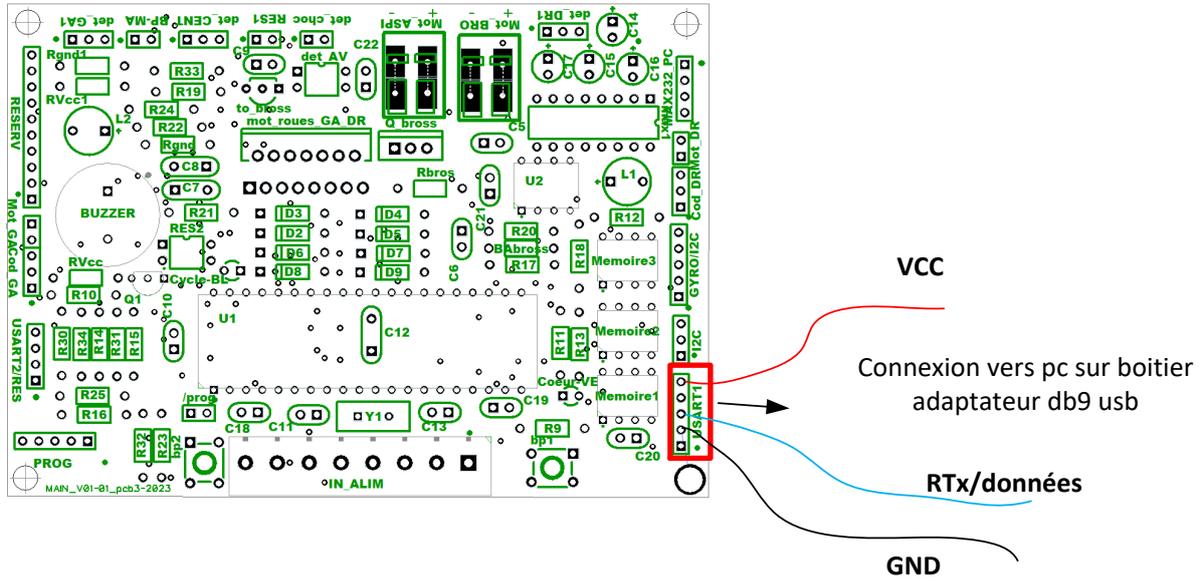
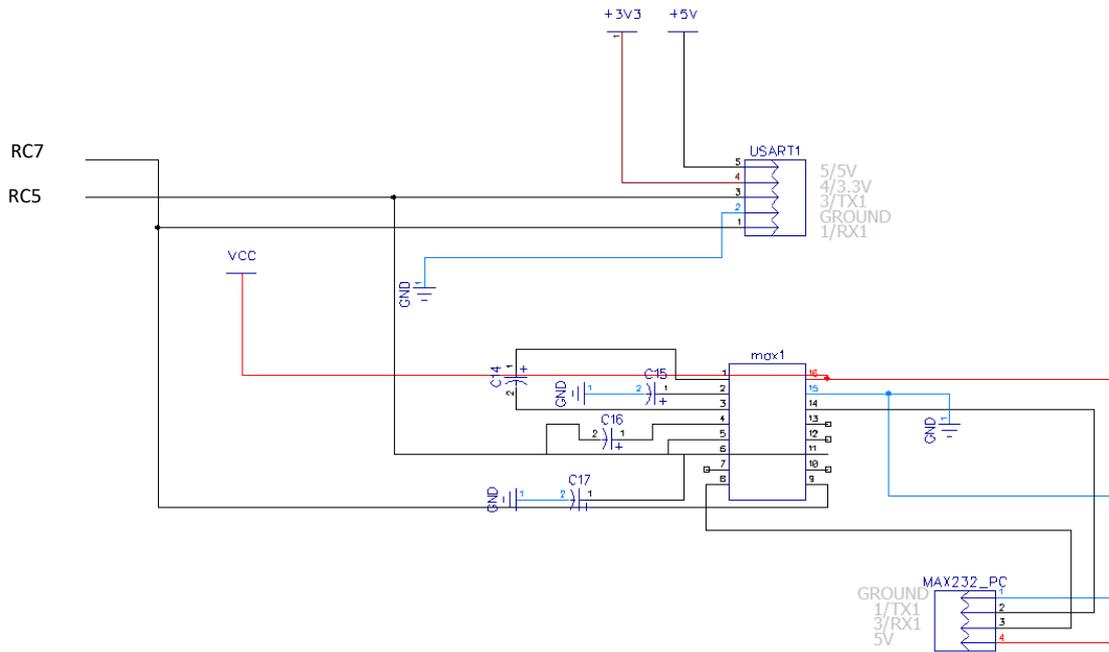


Schéma des deux options



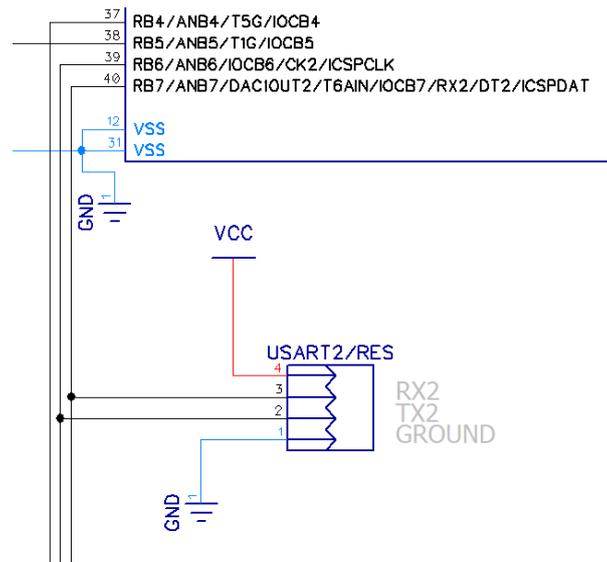


Le 23-11-2023 V1

Connexion sur usart2

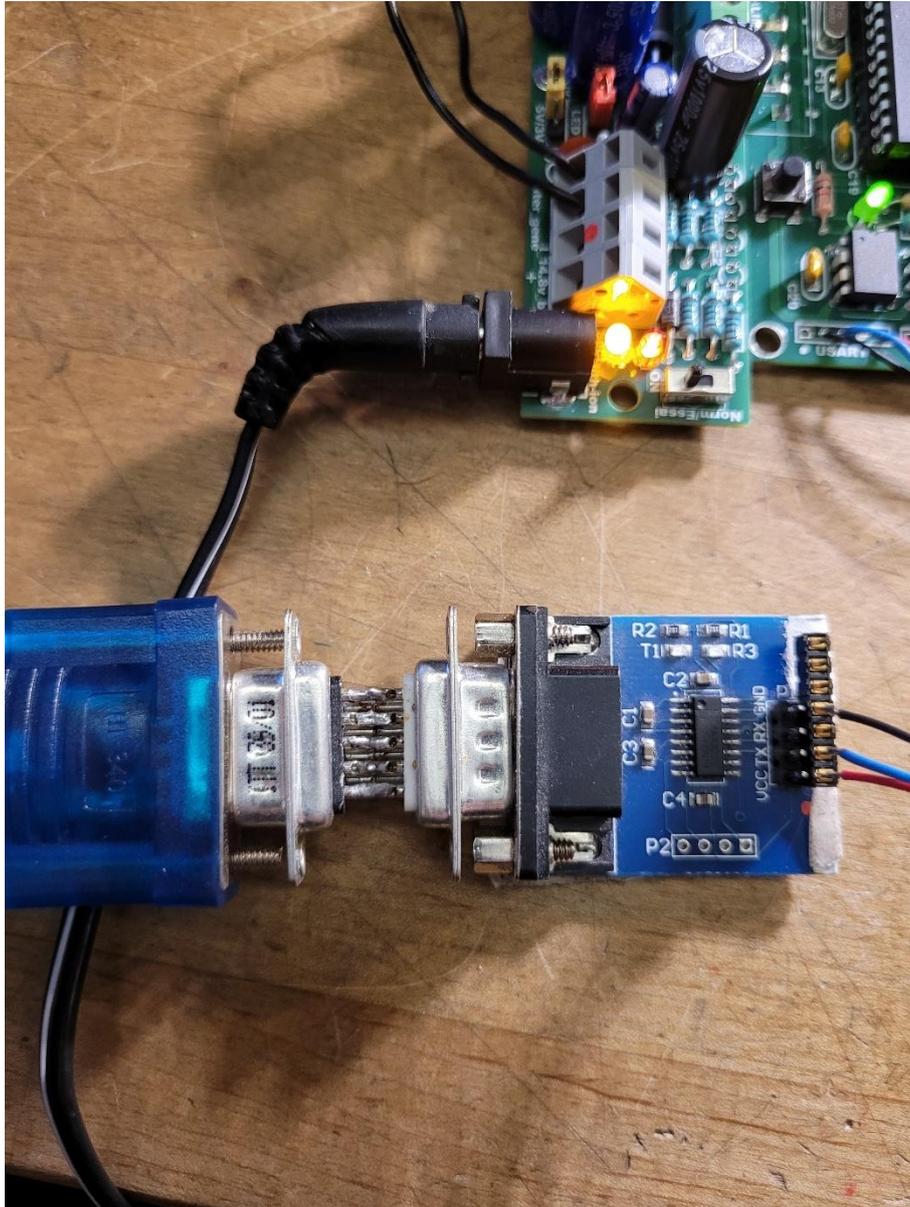
Nous disposons d'un deuxième usart ce qui peut être très intéressant quand on veut lire des informations sur la transmissions qui s'adresse a un autre périphérique

exemple :développement du lidar avec une connexion sur pc pour voir les résultats envoyés par le lidar.





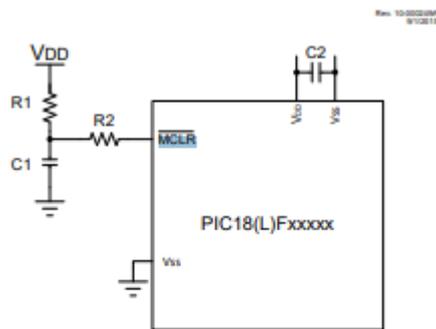
Le 23-11-2023 V1



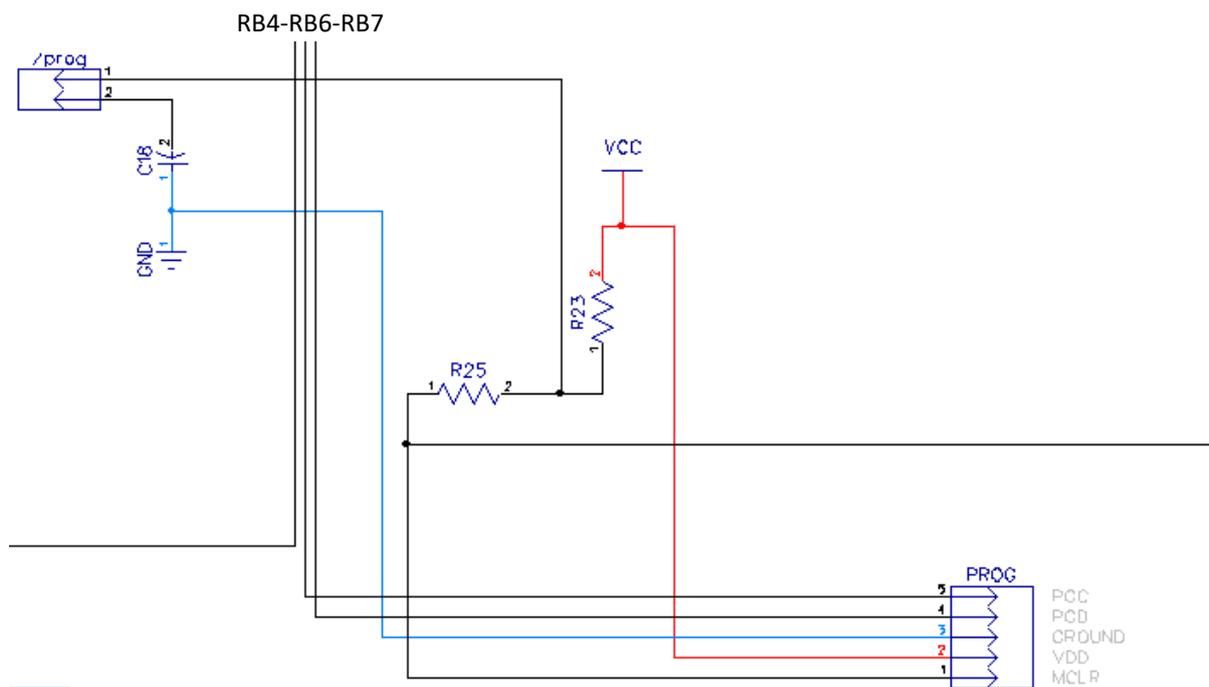
Recommandation de microchip



Le 23-11-2023 V1

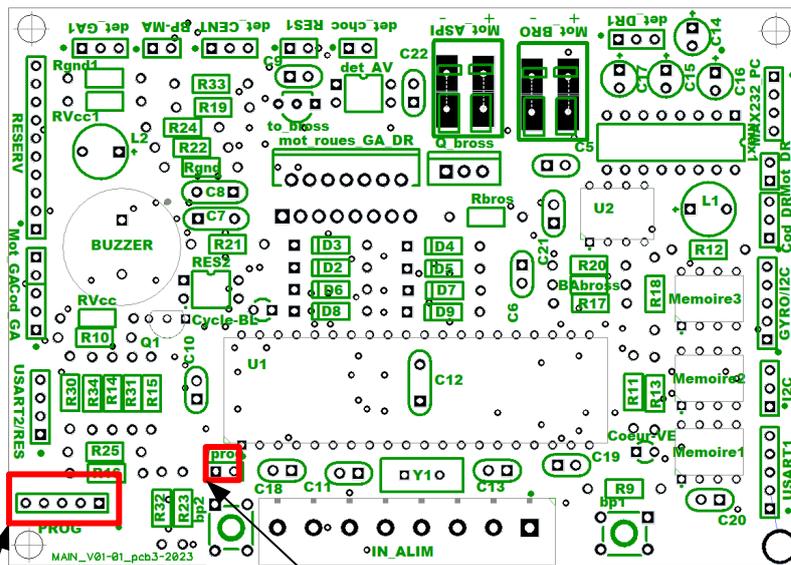


Key (all values are recommendations):
C1 and C2 : 0.1 μ F, 20V ceramic
R1: 10 k Ω
R2: 100 Ω to 470 Ω





Le 23-11-2023 V1



Programmeur pickit

/prog