

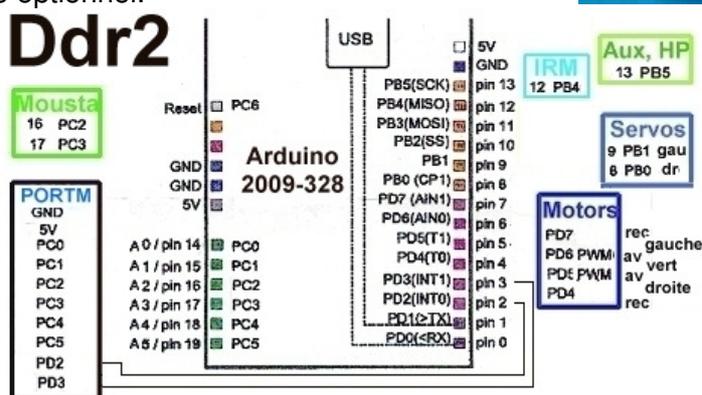


## Exploiter la carte Diduino-Robot Ddr2

La carte Diduino-Robot est compatible Arduino-Freduino, mais plusieurs "shields" sont répartis sur la carte, en utilisant les pins à disposition. Détaillons les possibilités de la carte, en s'adressant à un lecteur qui sait programmer une carte Arduino. Une note d'application provisoire (nos tests sous [www.didel.com/diduo/DiduinoRobotTestsHard.pdf](http://www.didel.com/diduo/DiduinoRobotTestsHard.pdf)) et des documents pédagogiques sont en préparation.



Le schéma ci-dessous résume ce qui a été ajouté à la base Arduino. Par rapport au Ddr1, les capteurs de distances et de suivi de ligne ont été reporté dans un module optionnel.

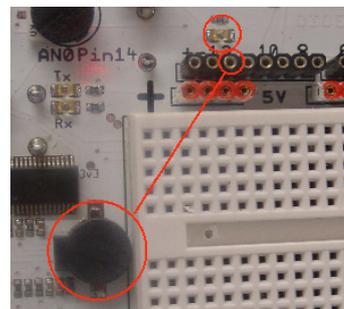


### Quelques caractéristiques

Dimensions 120 x 130mm. Diamètre du disque 120mm.  
Poids sans accus 120g. Avec accus 210 à 240g  
Consommation Moteurs arrêtés 5V 30 mA Avance 150mA Moteurs bloqués 400mA

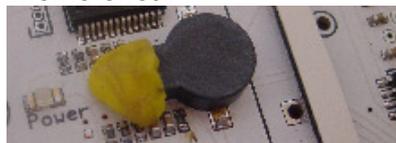
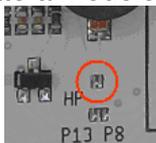
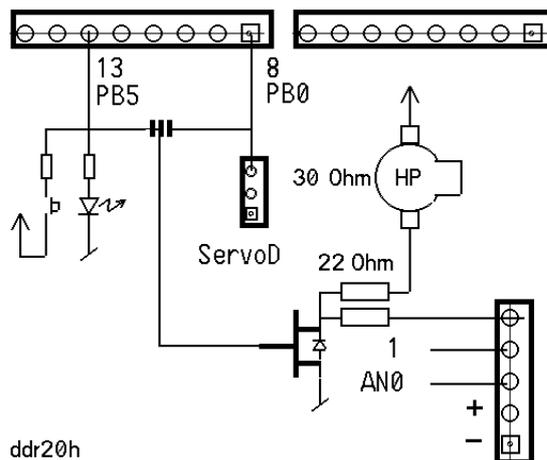
### Haut parleur

Aussi appelé buzzer (un buzzer en principe a un ampli incorporé et fait un son continu dès qu'il y a assez de tension), le petit haut-parleur est sur la même pin 13 que la LED utilisée dans les premiers exercices. Quand la pin 13 est activée, la LED s'allume, le courant passe dans la bobine du haut-parleur et la membrane est attirée. Elle est relâchée si la pin 13 est désactivée. On entend un clic à chaque activation et désactivation. Si on active/désactive avec une demi-période de 5 à 0.05 millisecondes (50000 à 50 microsecondes), on entend un son de 100 Hz à 10 kHz.



Un poussoir a été ajouté en parallèle. Le logiciel peut lire ce poussoir, qui évidemment allume la Led, et perturbe le HP.

Un transistor type N (IRLML2502) commande le haut-parleur, à travers une résistance de 22 Ohm qui limite le courant (et réduit un peu le son). Un connecteur à l'arrière du robot est prévu pour allumer une ampoule (voir l'expérience "Les robots se suivent" dans xxx) et pourrait aussi commander un électro-aimant à faible courant. On peut déconnecter le HP en coupant 0.5mm de conducteur entre deux zones de PCB, et rétablir la connexion avec une goutte de soudure. Un peu de pâte à modeler a le même effet !

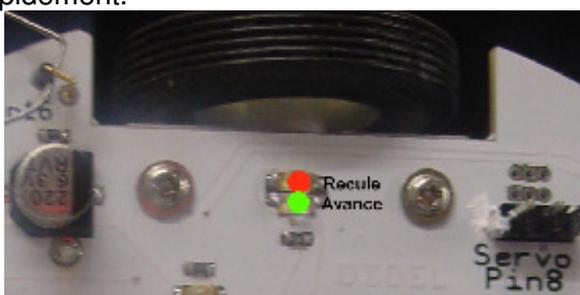
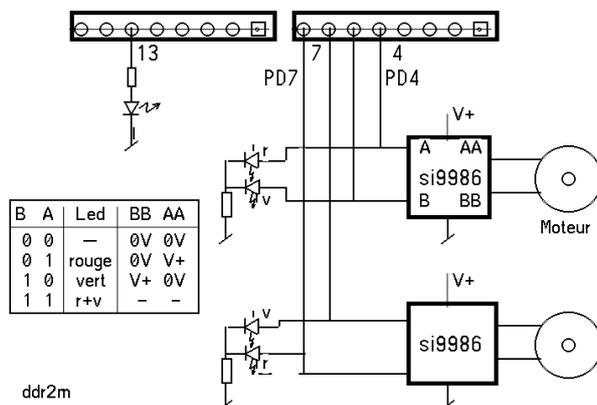


## Amplis moteur

Les moteurs ne sont pas alimentés par USB, mais par les accus seulement. Ce que l'on veut, c'est mettre au point le comportement du robot et on n'a pas besoin de le voir se déplacer à chaque itération de programmation, Des Leds bicolores donnent une information très complète sur la direction et la vitesse. La Led en vert signifie que le moteur fait avancer.

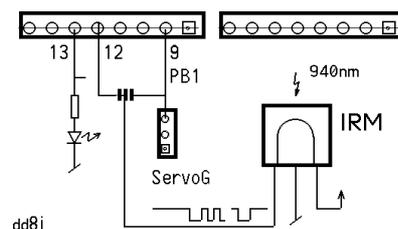
Les pins 5 et 6 actives font avancer (sorties PWM ). Plusieurs shields moteur utilisent les mêmes pins 4 à 7.

A noter que le circuit utilisé, le Si9986 a un mode "roue libre" lorsque les deux entrées sont à l'état "1". Dans l'état avec les deux entrées à zéro, le moteur est en "court-circuit" et se freine rapidement.



## Module IRM

Le capteur IRM contient une électronique de filtrage de signaux de 1-2ms modulés à 38 kHz. Ce n'est pas un phototransistor. Il peut s'utiliser en tout-ou-rien, ou décoder une télécommande. Le cavalier qui choisit la pin est en-dessous du buzzer.

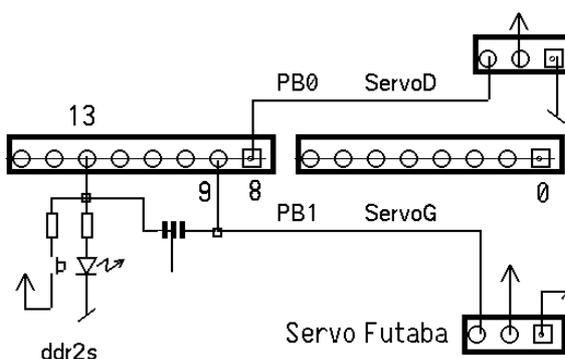


## Servos

Deux prises servos existent sur le robot, reliées aux pins 8 (PB0) et 9 (PB1).

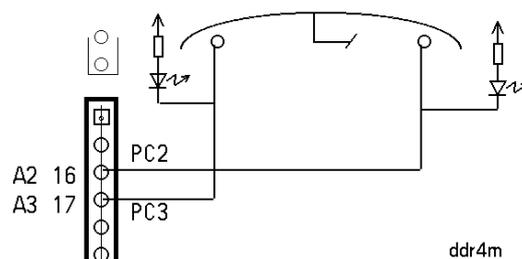
Le problème des servos est la multiplicité des prises, et le fait que certaines prises ont le Gnd au milieu.

Les prises sont au pas de 2.5mm avec le Gnd (pin carrée) selon la norme Futaba et Graupner.



## Moustaches

Les moustaches en circuit imprimé sont fiables et ont l'avantage d'être simples à comprendre et à gérer. Elles remplacent deux poussoirs pour les exercices, pour initialiser par exemple le programme dans l'un de 4 modes.



## Port Microduals

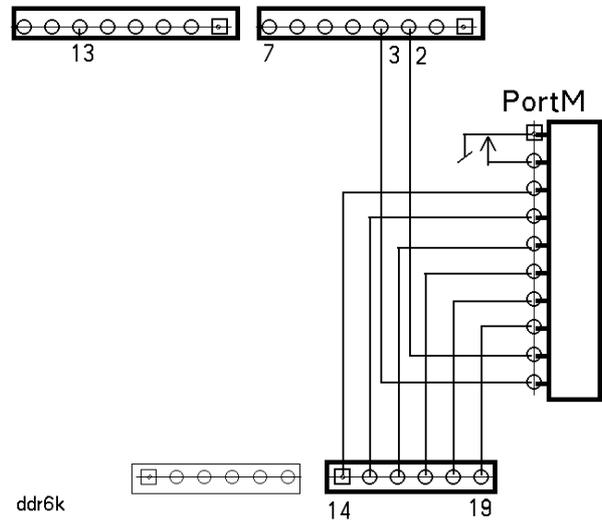
Pour afficher un compteur 8 bits sur le PortM on utilise la procédure `WritePortM`, basée sur le C et pas sur les simplifications d'Arduino..

```
//CountOnPortM.ino
void setup() {
  DDRC = DDRC | 0b00111111 ;
  DDRD = DDRD | 0b00001100 ;
}

void WritePortM (int vx) {
  PORTC = vx;
  PORTD = vx<<4;}

int cnt;

void loop() {
  cnt++; WritePortM (cnt);
  delay(1000);
}
```

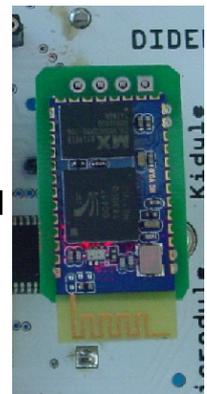
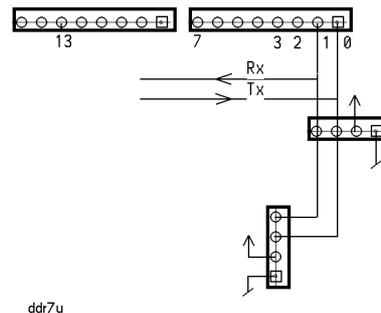


## Noms proposés

Haut-parleur	Potentiomètres	Moteurs	Servos	Distance	
13 HP	17 LdrGauche 16 LdrDroite	6 MotGAv 5 MotDAv vert	7 MotGRec 4 MotDRec rouge	9 ServoG 8 ServoD	19 DistG 18 DistD 10 Eclaire

## Communications série UART

Les signaux Rx, Tx, dont on voit l'activité sur 2 Leds liées au FTDI, sont utilisés pour le chargement de programmes et pour l'interface PC. Les modules Bluetooth se contentent de ces deux lignes et l'idée est de ne pas utiliser un shield existant, mais un module miniature avec un adaptateur. Deux connecteurs ont été prévus pour cet adaptateur à définir. Ces connecteurs sont utiles pour brancher un crayon logique, qui visualise si les pins sont en entrées ou actives, et montre les impulsions brèves.

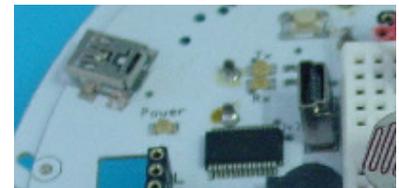


## Communications série SPI ou I2C

Les pins 10 à 13 sont utilisées par des modules SPI ou I2C et se trouvent en concurrence avec des fonctions de la carte que l'on ne veut pas nécessairement perdre. Des cavaliers ont été prévus pour libérer les pins 13 et 12. La pin 11 n'est pas utilisée. La pin 10 (SS) est utilisée par le capteur de distance.

## USB

Un connecteur vertical a été rajouté, pour permettre des tests à mobilité réduite à côté du PC, sans accus. L'interrupteur permet de décider si les moteurs tournent.



## Batteries

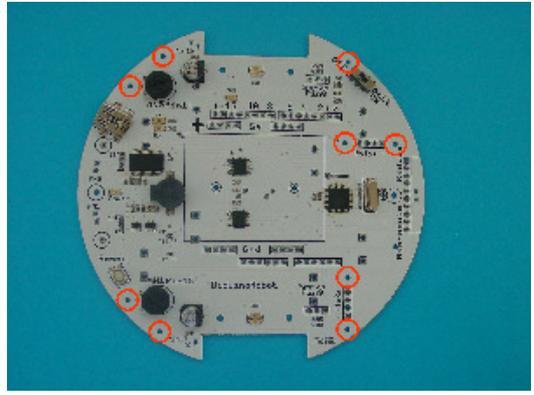
Le Diduino-Robot fonctionne sur batteries AA rechargeables. L'utilisation de piles n'est pas encouragée écologiquement. Le chargeur local recharge les accus quand USB est sous tensions.

Un connecteur vertical a été rajouté, pour permettre des tests à mobilité réduite à côté du PC, sans accus. L'interrupteur permet de décider si les moteurs tournent.



## Fixation des accessoires

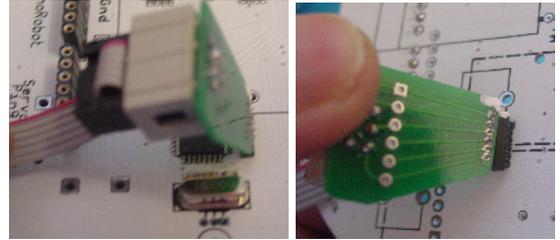
Des trous de 1.4mm sont répartis sur la périphérie. Les équerres peuvent être vissées avec des vis autotaraudeuses dans le PCB. On peut utiliser des vis M1.6 (tarudage préalable) et s'il n'y a pas trop d'effort, éviter les écrous.



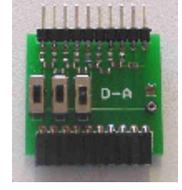
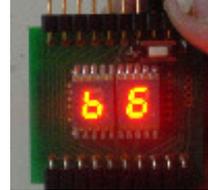
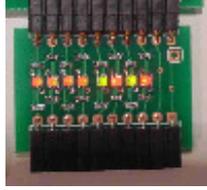
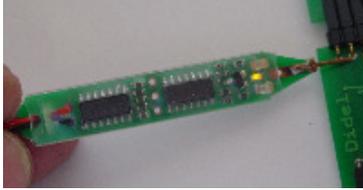
## Programmation du processeur AtMega 328

Le connecteur de programmation traditionnel des AVR (2x3pins), utilisé pour charger le bootloader est remplacé par un connecteur en ligne au pas de 1.27mm. L'adaptateur AVR permet de se relier à un programmeur AVR en passant par le connecteur à 6 pins.

On trouve aussi toutes les pins de programmation sur les connecteurs Arduino.

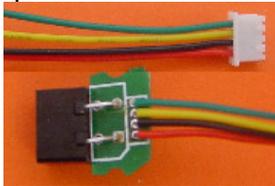


## Crayon lumineux, Microdules



## Qu'est-ce qui vous manque ?

Le robot est petit, c'est tout son intérêt. Ne demandez pas des actionneurs et capteurs disproportionnés



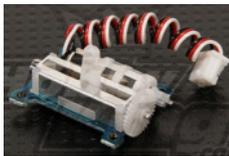
IrDist Didel 5-40cm



Sharp GP2D12



SRF06 2cm – 5m



Servo 1.7g



Servo 6g



Suivi de piste

Suivi de lumière

Sonar intelligent