



XBotMicro pour Arduino/Diduino

Le XBotMicro développé en Suisse, est une base de robot à faible consommation compatible entre autres avec Arduino. XBotMicro est le complément idéal pour comprendre les capteurs, moteurs, interruptions et apprendre à programmer en C des applications intéressantes. Lors du développement, XBotMicro est alimenté par la carte microcontrôleur 5V, 50mA. Lors de déplacements autonomes, il est alimenté par son accumulateur 3.2 à 3.7V, 30 mAh plus 30-50mA pour la carte AVR328. La plupart des cartes microcontrôleurs tournent dans cette gamme de tension.

Ce que vous obtenez

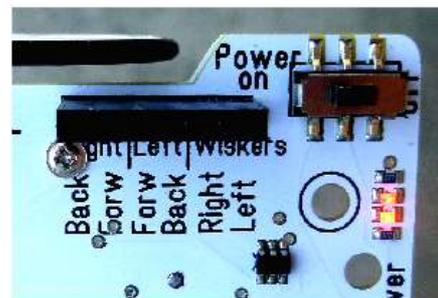
La boîte contient la base de robot testée avec un accumulateur, des moustaches, des potelets et des fils de câblage.

Le XBot est alimenté par un accumulateur LiPo de 250mAh, qui permet 4 heures d'autonomie si on n'utilise pas des capteurs gourmands en énergie. L'interrupteur, situé sur le côté droit, alimente le robot à 3.7V. Il peut rester enclenché si le +5V est connecté.

Sur le côté gauche, c'est la puissance pour les moteurs.

Commutez sur Power on et observez les 2 LEDs proches de l'interrupteur.

Si la tension est supérieure à 3.6V, les 2 LEDs sont allumées. Si elle est inférieure à 3.2V, une seule LED est allumée et il est temps de recharger l'accu.

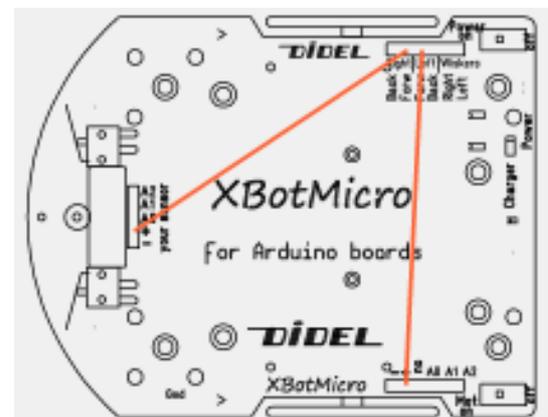


Pressez sur les moustaches, les LEDs s'allument lorsque les microswitch correspondants sont actifs.

Vous trouvez l'état des moustaches Droite et Gauche sur les deux pins arrière. Etat "HIGH" lorsque la moustache n'est pas pressée, "LOW" dans le cas contraire.

XBotMicro peut se déplacer seul

L'interrupteur à l'arrière gauche permet d'alimenter les deux moteurs. Vous pouvez immédiatement tester les fonctions des 4 lignes qui commandent les moteurs. Prendre un fil pour connecter le + (HIGH) à l'une des quatre entrées des moteurs. Il y a un second + sur le connecteur frontal; relier deux fils et observer la vitesse de déplacement du XBot. Stoppez les moteurs à l'aide de l'interrupteur et vous pouvez encore voir sur les LEDs bicolores dans quel sens les moteurs tourneraient s'ils étaient alimentés - vert signifie avance et rouge recule. Remarquez encore qu'il n'y a pas besoin de processeur pour faire plusieurs expériences amusantes avec juste 2 LDRs ou un 555. Posez un breadboard sur la surface du Xbot pour tester vos schémas.



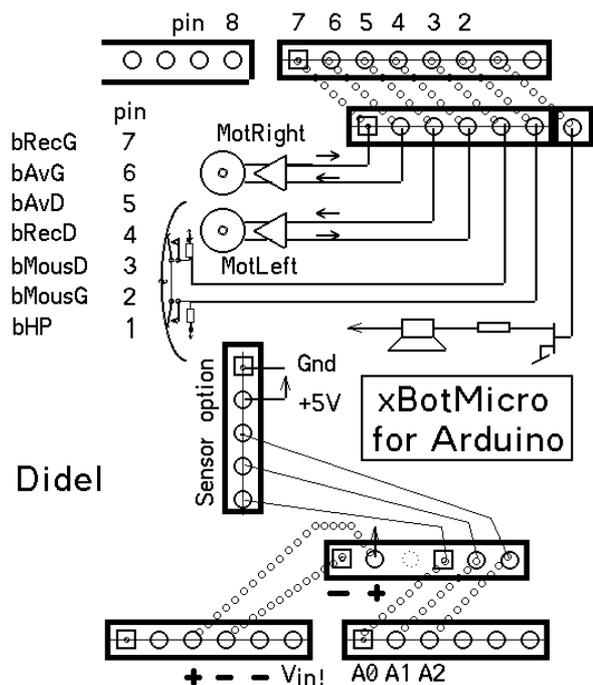
Autres caractéristiques

Les engrenages des moteurs ont un rapport de 1:96. Lors des déplacements normaux le courant est de 20 à 30mA par moteur. Avec les roues Didel de 32mm de diamètre, la vitesse maximale est de ~50cm/s. Il est impossible de garantir une trajectoire parfaitement rectiligne. La capacité de l'accumulateur LiPo est de 250mAh. Il se coupe s'il y a court-circuit ou décharge complète. Comme pour tous les accus LiPo, la tension à pleine charge est de 4.1V, mais elle descend rapidement à 3.8V, reste longtemps à 3.7-3.5V. L'accu LiPo se recharge quand la base reçoit du 5V via la carte microcontrôleur. Les temps de programmation étant toujours très supérieurs au temps passé à tester des comportements autonomes, la durée de l'accu est importante seulement dans une exposition: le Xbot roulera plus de 4 heures à éviter les obstacles.

Signaux de commande

Six lignes commandent les moteurs et reçoivent l'information depuis les moustaches. Une ligne supplémentaire commande le haut-parleur en utilisant la pin1, transmissin série. Il en résulte un son bizarre quand le programme se charge (très utile) et quand le Serial.print envoie de l'information. Soulevez le fil si cela vous gêne.

Les deux lignes d'alimentation sont sur le connecteur arrière gauche. Ces lignes sont alimentées en +5V lorsque le XBot est connecté à USB. Si l'accu LiPo n'est pas entièrement chargé, ce +5V est utilisé pour le chargeur (environ 50mA pendant la charge, LED allumée). Lorsque le robot est autonome, le 3.7V de l'accu LiPo est envoyé à la carte processeur.



Comment connecter une carte compatible Arduino ?

Placez les potelets selon votre carte. Vis en dessous pour que la carte tienne sur le filetage.

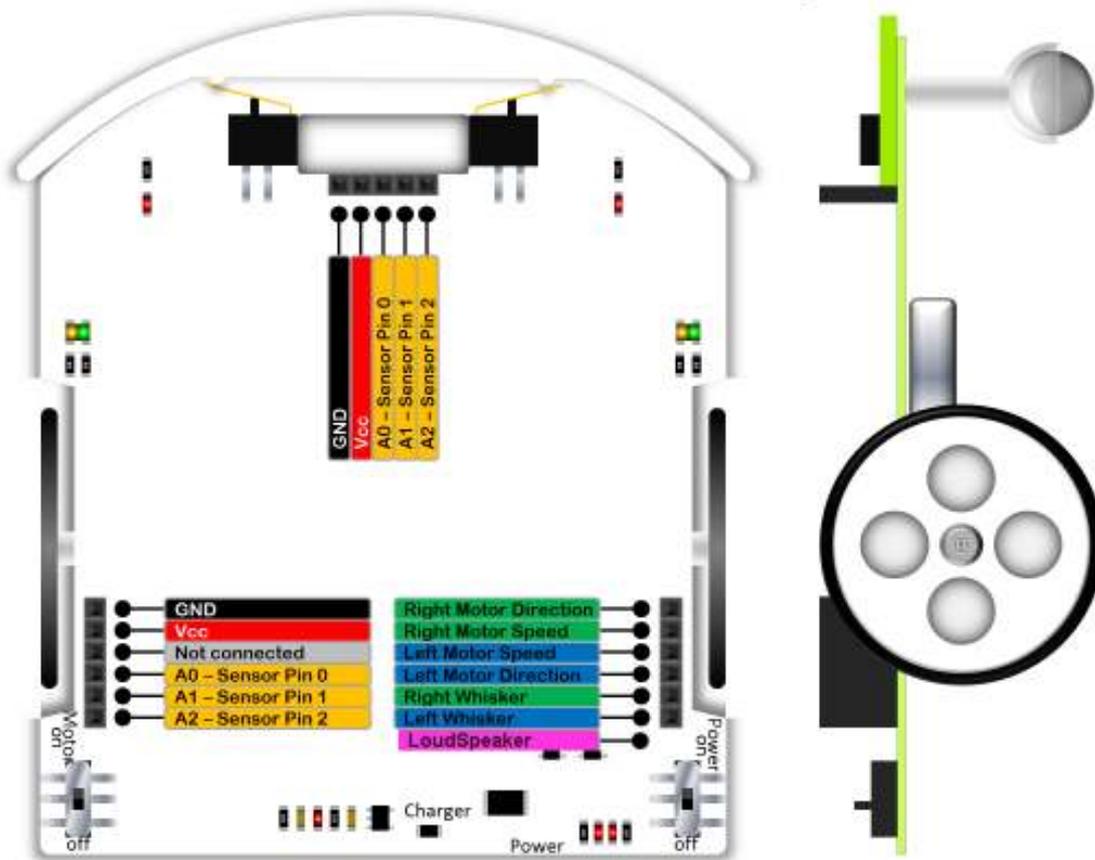
Un potelet comme écrou suffit à maintenir la carte.

Des fils courts ont été spécialement fabriqués pour que les connexions soient propres. Il est conseillé d'utiliser des fils noir et rouge pour l'alimentation et de faire attention au câblage, car de fausses connexions peuvent détruire des composants.



Generously provided by
<http://mathias-wilhelm.de/>

A TRIBUTE TO DIDEL **XBotMicro**



Câblage et définitions (compatible LearnCbot www.didel.com/coursera/LC.pdf)

<pre>#define bRecG 4 // PORTD actifs à 1 #define bAVG 5 #define bAVD 6 #define bRecD 7 // pour les exercice simples #define Avance PORTD = 0b01100000; PORTD &= 0b01101111 #define Recule PORTD = 0b10010000; PORTD &= 0b10011111 #define TourneD PORTD = 0b10100000; PORTD &= 0b10101111 #define TourneG PORTD = 0b01010000; PORTD &= 0b01011111 #define Stop PORTD &= ~0b11110000 #define bMousD 3 // PORTD, 3) #define bMousG 2 #define ObsG !(PIND&1<<bMousD) #define ObsD !(PIND&1<<bMousG)</pre>	<pre>void setup () { DDRD &= 0b11110011 ; // bits 3 2 in DDRD = 0b11110000 ; // bits 7..4 out PORTD &= 0b00001111 ; // motors stopped } // principe pour éviter les obstacles: void loop () { ... avancer if (ObsD) { ... recule, tourner à gauche } else if (ObsG) { ... } }</pre>
---	--

Pour une vitesse proportionnelle, le PWM existe sur les pins 5 et 6. Il faut astucier pour la marche arrière, voir <http://www.didel.com/diduo/CommandeMoteurs.pdf>

Programmes exemples

Effectuer des figures géométriques est facile à programmer. Le glissement des roues ne permet pas une bonne répétitivité, qui dépend de plus de la tension de l'accu.

Avec seulement les moustaches, on peut faire des exercices plus variés et intéressants que cela en a l'air:

- ajouter un mouvement aléatoire pour le robot ne se piège pas dans un coin
- suivre un paroi avec des coins rentrant
- suivre une paroi avec des coins sortants
- sortir d'un labyrinthe
- balayer l'espace en zig-zag entre 2 murs parallèles.

Capteurs

Les capteurs permettent une grande variété d'applications. Ils ont tous des limitations intéressantes à étudier, mais qui parfois ne facilitent pas une programmation fiable de l'application. Voir www.didel.com/xbot/RobotiqueMotivante.pdf

Librairie

La documentation sous www.didel.com/didduino/Liens.pdf s'est développée sur plusieurs années, les habitudes de programmation et le matériel a évolué. Le MOOC "Microcontrôleur" a apporté une approche qui permet de mieux programmer, mais il faut perdre certaines pratiques Arduino.

Une librairie performante est en développement, elle est écrite dans une optique C portable en dehors de l'environnement Arduino, donc apparemment indigeste pour ceux qui ont été biberonnés par Arduino. Si vous avez l'âge de raison: www.didel.com/xbot/Librairies.pdf

Câbler un autre microcontrôleur

Tout microcontrôleur qui a 7 entrées-sorties peut piloter la base XbotMicro. Dans le cas du Raspberry A+, un adaptateur facilite le câblage, et l'alimentation 5V 1A doit être fournie par un accu 5V de mobile ou tablette. Un AtTiny-24 tourne la librairie ci-dessus aussi bien que le Diduino; il est programmé avec le AtTinyProg. Un Pinguino nécessite d'adapter la routine de l'interrupt timer.

