



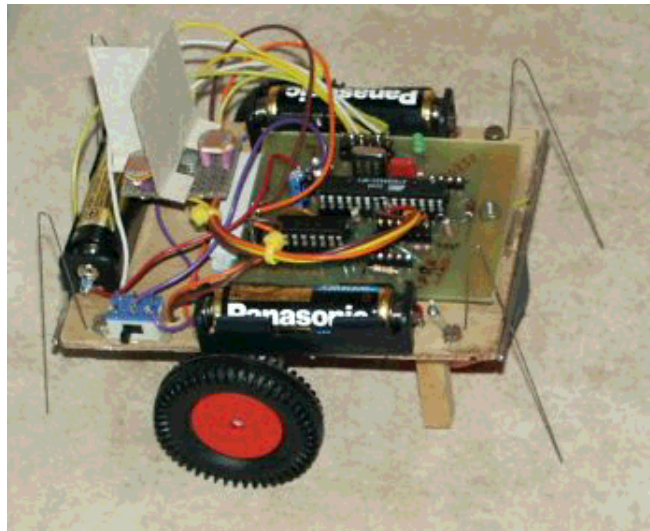
par Katja et Guido Socher
<katja@linuxfocus.org
guido@linuxfocus.org>

L'auteur:

Katja est l'éditrice allemande de LinuxFocus. Elle aime Tux, l'infographie, les films & la photographie et la mer. Sa page personnelle est là.

Guido est fan de Linux depuis la première heure parce que ce système vous donne le choix et la liberté. Cela permet de choisir et développer des solutions en accord avec ses besoins.

Construire un robot chercheur de Lumière



Résumé:

Dans cet article, nous décrivons comment construire un robot autonome, avec un microcontrôleur, qui essaie de se déplacer vers le point le plus lumineux.

Traduit en Français par:
John Perr
<johnperr@linuxfocus.org>

Introduction

Il y deux ans, nous avons décrit dans LinuxFocus un robot marcheur contrôlé par Linux. Il avait une conception très spéciale car il marchait sur ses pattes sans l'aide de moteurs conventionnels. Ce fut un aspect très intéressant du robot. Toutefois, il était très lent, demandait de fort courants et nécessitait des composants spécifiques ainsi qu'un certain doigté pour sa construction.

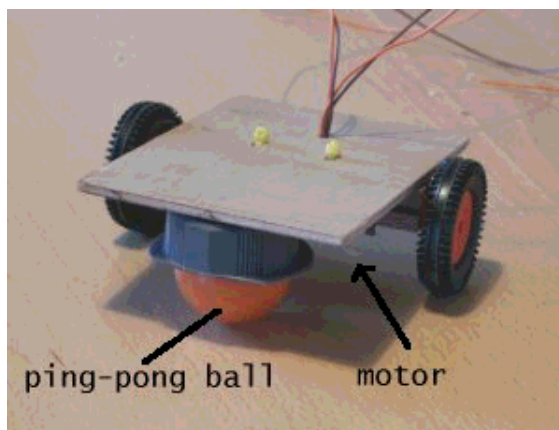
La conception de notre nouveau robot est très différente. Il est bon marché et vous pourrez le construire à partir de composants facilement disponibles partout dans le monde. C'est un robot contrôlé par un microcontrôleur AVR. Comme il est autonome (c'est à dire non contrôlé par une personne) nous l'avons programmé pour qu'il se dirige vers la source de lumière la plus brillante de la pièce.

La mécanique

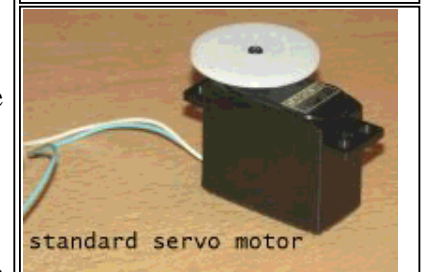
Le robot n'a que deux roues entraînées par deux moteurs indépendants. La troisième roue est une balle de ping-pong. Cette configuration permet au robot de tourner sur place. Les roues en caoutchouc proviennent d'un jouet mais vous n'avez pas à casser les vôtres. Le couvercle d'un pot à confiture, pourvu d'un élastique constitue une roue très acceptable.

Il est évident qu'un robot autonome doit pouvoir fonctionner à partir de piles. Étant donné que le microcontrôleur fonctionne avec une tension de 4,5 à 5 Volts, il doit en être de même avec les moteurs. Ils doivent aussi avoir une consommation qui permette d'avoir des piles et un circuit de contrôle de taille raisonnable. Nous utilisons pour cela un circuit intégré dédié au contrôle des moteurs, le l293d. Il peut supporter des charges jusqu'à 0,5A. Les moteurs devront par conséquent avoir besoin de moins de 0,5A dans le pire des cas.

Nous avons utilisé deux petits moto-réducteurs de chez Conrad (www.conrad.fr, référence 242802) mais vous pouvez utiliser n'importe quel petit moto-réducteur. Nous pensons que la meilleure solution aurait été d'utiliser des servomoteurs de modélisme comme ceux montés sur les bateaux, les voitures ou les avions. Normalement, ces servomoteurs sont limités angulairement mais cela peut être corrigé en ouvrant la boîte et en enlevant la butée, le potentiomètre et l'électronique. C'est un parfait petit moteur puissant et facile à trouver.

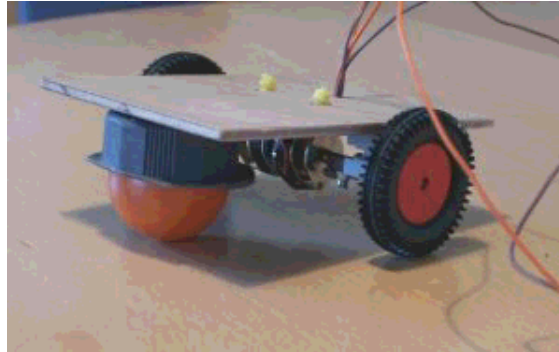


Le petit moto-réducteur de chez Conrad



Un servomoteur standard modifié pour fonctionner comme un moteur. Il s'agit sûrement de la meilleure solution, mais l'idée ne nous est venue qu'après la construction du robot

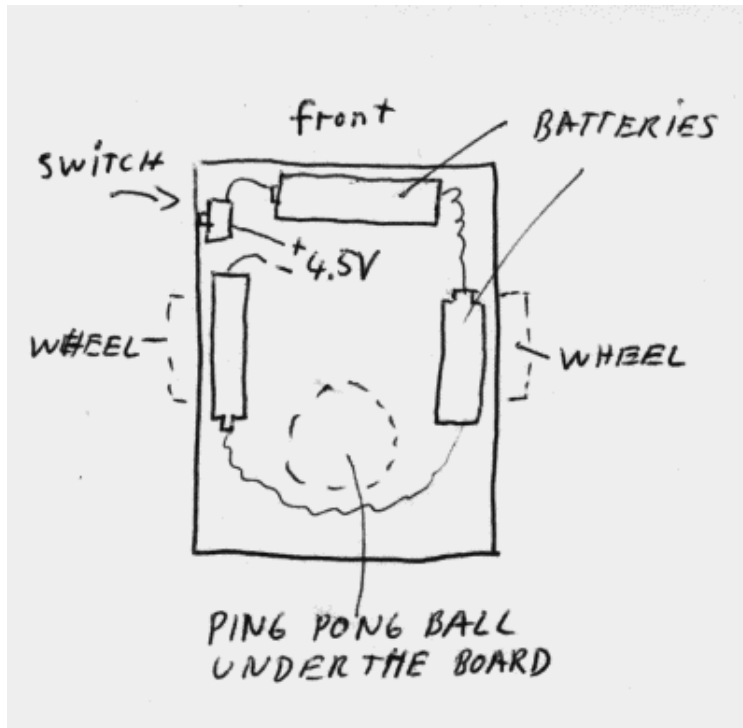
Pour construire le robot, monter les moteurs sous une planchette de bois (12cm x 9cm) et placer les quasiment au milieu afin que la charge repose sur les deux axes. La troisième roue, la balle de ping-pong, ne devra supporter qu'un faible partie du poids du robot afin qu'elle glisse facilement dans son logement (voir les dessins).



Le logement de la balle de ping-pong est réalisé à partir du couvercle d'une petite bouteille en plastique qui se trouve avoir la bonne taille.



L'alimentation est assurée par trois piles de taille AAA. Les positions des trois supports sont indiquées ci-dessous. Les piles sont plutôt lourdes, c'est pourquoi il faut faire attention à placer la majorité du poids sur les roues et un peu sur la balle de ping-pong. Un interrupteur marche-arrêt peut être placé sur le coté.



Les capteurs

Notre robot a deux types de capteurs:

- contact: le robot sait s'il a rencontré un objet
- lumière: le robot peut chercher le point le plus lumineux de la pièce

Les capteurs de contact sont de simple interrupteurs faits de tige d'acier. Elles sont pliées comme indiqué ci-dessous:

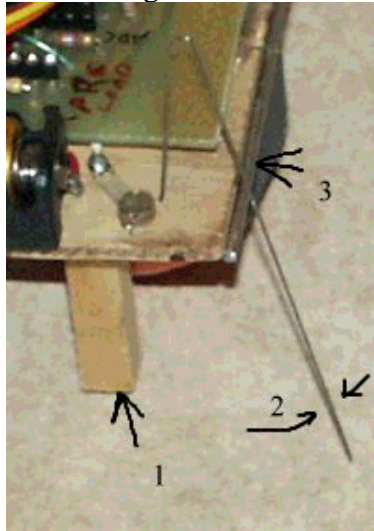


Il y a quatre capteurs de contact, montés avec des vis sur les coins de la planchette.

Quand le robot rencontre un objet, la tige d'acier (2, figure ci-dessous) touche le second conducteur sur la planche (3) et cela ferme un circuit.

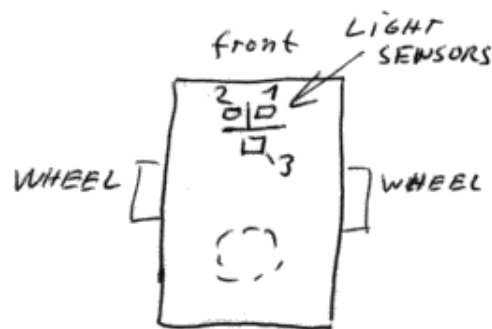
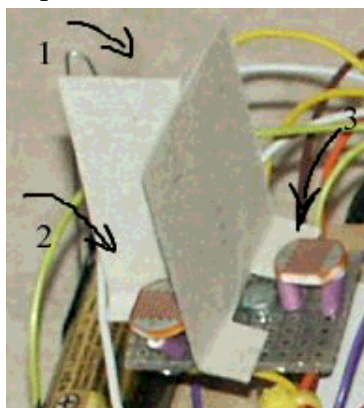
Afin d'éviter de tordre les fils d'acier quand la balle de ping-pong n'est pas dans son logement, nous avons ajouté un support en bois (1) sous le châssis. Il doit être 5 mm au dessus du sol quand la balle de ping pong est dans son logement.

Les tiges d'acier des contacts doivent avoir une garde de 5-7mm au dessus du sol.



Les capteurs de lumière sont 3 photo résistances entre lesquelles nous avons placé des masques en carton comme le montre la figure ci-dessous. Ces masques créent des ombres sur les résistances quand la lumière vient de coté. Au cas où la lumière vient exactement du dessus, les trois capteurs recevront la même quantité de lumière. La comparaison des valeurs des trois résistances permettra de déterminer la direction à prendre.

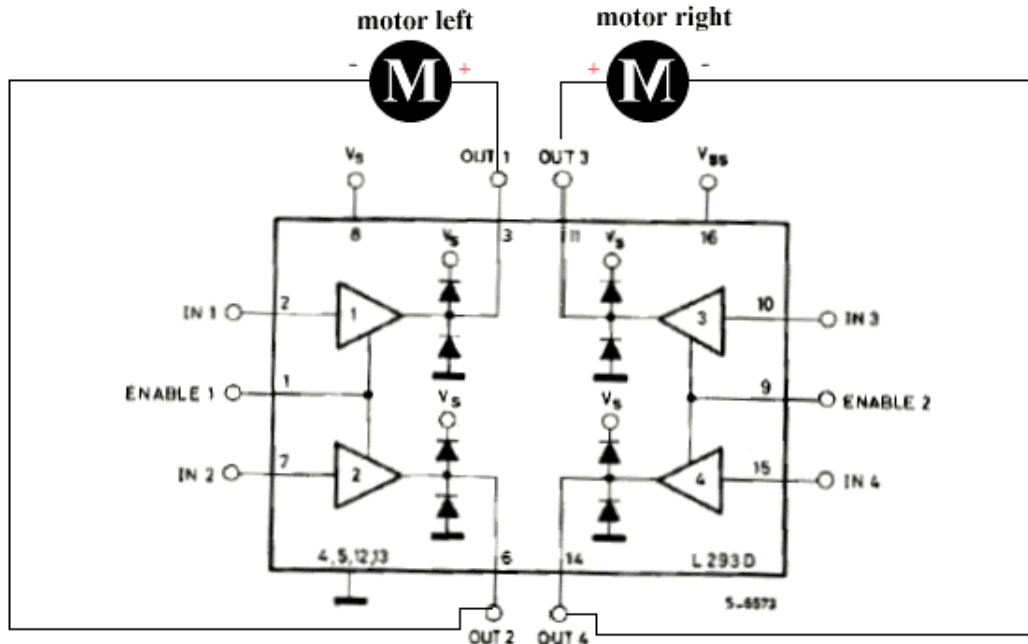
Vous pouvez souder les trois résistances sur un petit morceau de circuit imprimé d'expérimentation (ceux avec beaucoup de trous) et fixer l'ensemble au robot avec une unique vis.



La façon de connecter les capteurs et les moteurs au circuit imprimé du microcontrôleur sera expliquée plus tard. Maintenant que la partie mécanique est réalisée, jetons un oeil au cerveau du robot.

Le Circuit

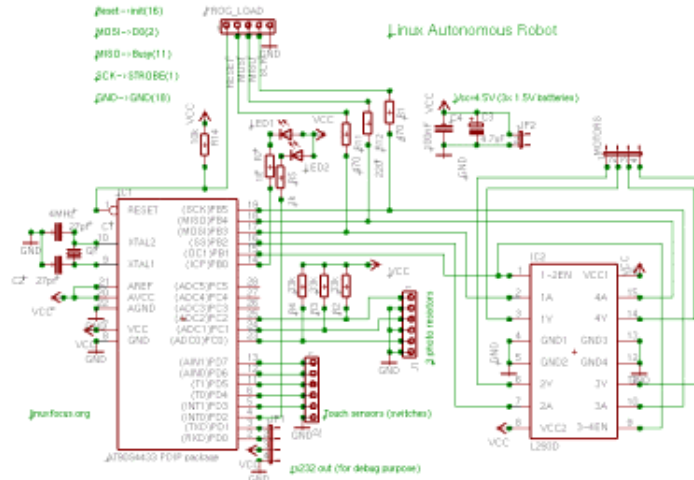
Nous utilisons un microcontrôleur AT90S4433 comme cerveau de notre robot mais ce cerveau ne peut délivrer suffisamment de puissance pour faire marcher les moteurs. C'est là que le circuit de puissance L293D intervient. Il contient quatre amplificateurs digitaux munis de diodes de protection contre les surtensions dues aux bobinages des moteurs. Cela permet aussi de piloter le moteur dans un sens ou dans l'autre.



Un des moteur est relié entre les sorties 1 et 2 et l'autre entre les sorties 3 et 4. La broche "enable" permet de contrôler la vitesse des moteurs en changeant la longueur des impulsions envoyées à cette broche.

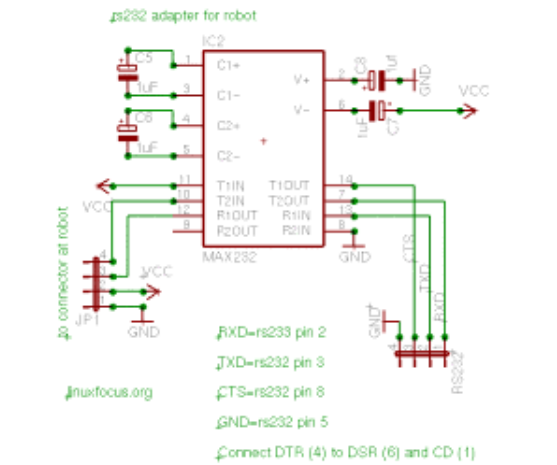
Le reste du circuit est très simple: Nous utilisons encore le microcontrôleur Amtel AT90S4433. Nous avons déjà utilisé ce microcontrôleur dans des articles précédents de LinuxFocus. Ses entrées analogiques peuvent être utilisées pour mesurer la quantité de lumière perçue par les photo résistances et les entrées digitales permettront de connecter les détecteurs de contact comme montré ci-dessous.

Les informations détaillées sur le microcontrôleur se trouvent dans l'article de Mars 2002 : Programmer le Microcontrôleur AVR avec GCC du même auteur.



Le circuit est alimenté en 4.5V. Trois piles de taille AAA batteries sont par conséquent suffisantes pour le fonctionnement du robot.

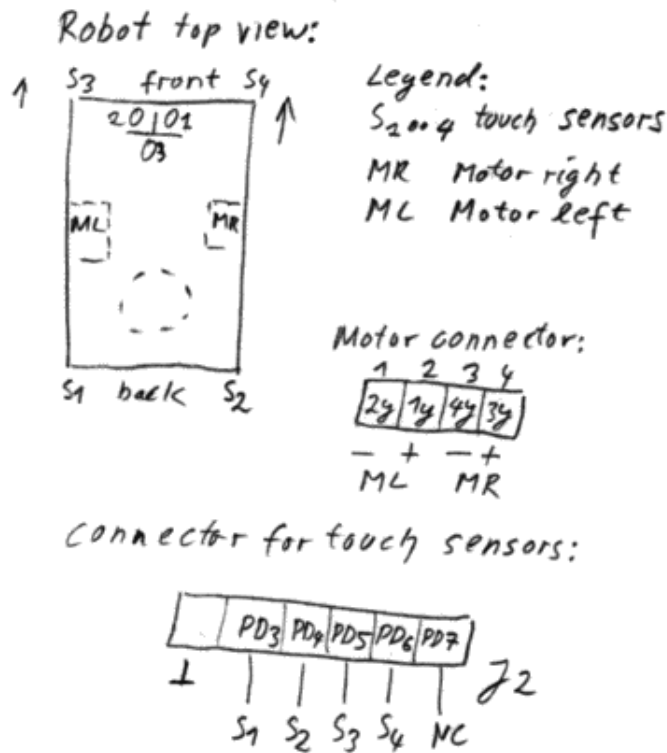
Le circuit de notre robot autonome devrait maintenant être prêt. Toutefois, se pose la question de savoir quoi faire si le robot ne fonctionne pas comme prévu à cause du logiciel. Rien n'est accessible, ni les valeurs des photo-résistances ni les décisions prises par le robot. Il nous faut une sorte d'écran ou d'affichage pour comprendre ce que fait le robot. L'interface série RS232 est bien adaptée à ce besoin car il est possible d'y afficher des contenus de variables et même de communiquer avec le robot. On peut par conséquent implanter le max232 et les autres composants nécessaires à la liaison série sur une platine séparée et la connecter à la demande:



Un lien donné en fin de cet article permet de télé-charger le circuit complet au format Eagle, avec le dessin du circuit imprimé et l'implantation des composants ainsi que le logiciel du robot. Nous ne décrivons pas l'implantation des composants ici, elle est visible avec Eagle. Le circuit est suffisamment petit pour se loger entre les piles.

Le dessin ci-dessous montre le brochage des capteurs de contact ainsi que comment connecter les moteurs. La polarité des moteur est choisie de telle sorte que le robot avance (dans le sens de la flèche) si +3V sont appliqués à la broche "+" et la masse à la broche "-". 1y à 4y sont les noms des broches du

1293d.



Le Logiciel

Nous n'entrerons pas dans les détails ici. Le programme principal se trouve dans le fichier linuxrobot.c (Télé-chargez le à la fin de l'article). Il y a beaucoup de commentaires dans le programme ce qui devrait en rendre la lecture facile pour un programmeur C. La boucle principale mesure d'abord les valeurs des photo résistances en activant 3 fois le convertisseur analogique-digital interne du microcontrôleur en mode "single shot". Les capteurs de contact sont ensuite testés et si l'un d'eux est actif, alors il a priorité sur les capteurs de lumière car le robot a probablement heurté un obstacle. Le moteur opposé au détecteur sera alors activé quelques millisecondes. Si aucun détecteur de contact n'est actif, alors les photo résistances sont comparées entre elles. La comparaison se fait dans la fonction `compare_with_tol()`. Pour éviter les perturbations dues au bruit, deux valeurs sont dites égales si leur différence est inférieure à 5 pour cent.

Selon les résultats de la comparaison des photo résistances, le choix du moteur à faire tourner est fait. Comme il n'y a que deux roues sur notre robot, il peut tourner sur place en faisant tourner une des roues plus vite ou même en faisant tourner les deux en sens inverse. La mesure étant effectuée plusieurs fois par secondes, le mouvement du robot semble continue même si nous arrêtons un moteur une fraction de seconde afin de tourner un peu à droite ou à gauche.

Montage final

En montant l'électronique, c'est toujours une bonne idée de procéder par étapes. Ceci permet de mieux cerner les éventuels défauts.

Il y a trois programmes de test différents inclus dans le packaging logiciel du robot. Le programme ledtest fait clignoter les deux leds. Il se charge avec la commande "make ledtestload". Ceci compilera le programme et le chargera dans le microcontrôleur. Les deux leds doivent commencer à clignoter immédiatement après le chargement du programme. Une fois ce test réussi vous êtes certain que le microcontrôleur avec son oscillateur et les connexions de chargement depuis le PC sont fonctionnels.

Viens ensuite le programme de test du moteur. Il réalise une "balle en caoutchouc électronique" et se charge avec la commande "make motortestload". Le programme motortest vérifie les capteurs de contact en permanence et déplace le robot à l'opposé de tout capteur actif. En touchant le robot avec une main d'un côté il rebondira. En mettant la deuxième main derrière le robot il rebondira d'avant en arrière entre vos deux mains comme une balle de caoutchouc. Si le robot réussit cet essai alors tout aura été vérifié excepté les capteurs de lumière et la liaison RS232.

Viens ensuite le programme d'essai final appelé adctest (compiler et charger le avec make adctestload). Le programme teste la connexion RS232 qui sert à déverminer le robot et à tester les 3 photo résistances. Chargez le programme dans le microcontrôleur et connectez l'adaptateur RS232 à votre PC. Lancer ensuite les 3 commandes suivantes dans un terminal:

```
make ttydevinit
./ttydevinit /dev/ttyS0
cat /dev/ttyS0
```

Le robot devrait périodiquement afficher les valeurs des intensités lumineuses qu'il a mesuré avec les capteurs photo sensibles.

Une fois tous les essais réussis, vous pouvez charger le programme final dans le robot avec la commande "make load". La meilleure aire pour les premiers essais est une pièce avec juste une lampe en son centre. Le robot devrait aller droit dans la direction de la lampe et s'arrêter là.

Il est amusant de voir comment il se retourne si vous le posez sur le sol le dos à la lumière ou comment il évite les ombres.

Problèmes et améliorations

Nous avons commencé ce robot à titre d'essai. ce fut amusant de construire un robot autonome, capable de prendre seul des décisions et qui n'a pas de connexion au PC. Le programme inclus dans le packaging que vous pouvez télé-charger à la fin de cet article est encore petit et simple mais fait ce que nous lui avons demandé: Le robot court vers le point le plus lumineux.

Voici quelques idées de développements complémentaires:

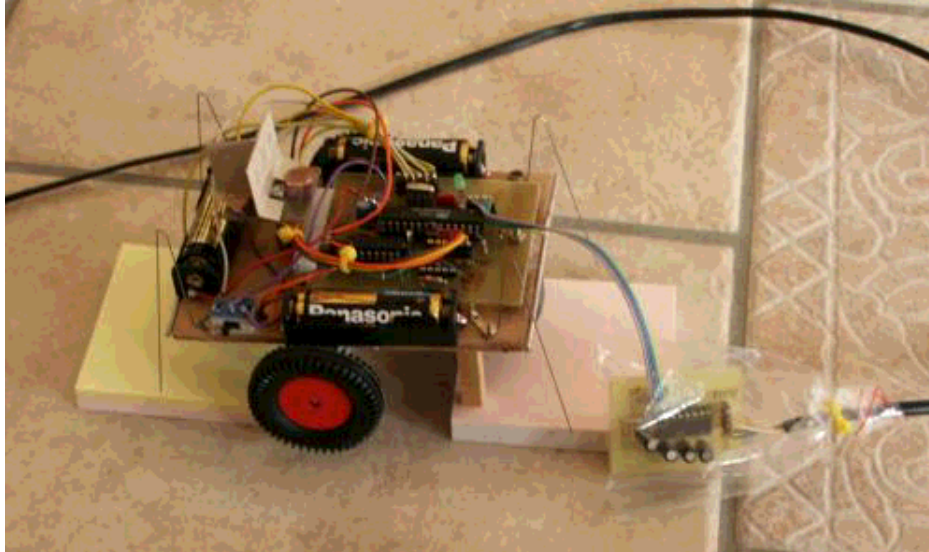
- Les capteurs de contact ne sont vérifiés qu'à de larges intervalles (quelques millisecondes) ce qui limite la capacité de réponse du robot. Ils devraient être vérifiés plus souvent.

- Si un des capteurs de contact est actif, alors il devient prioritaire sur tout le reste et le robot se déplace dans la direction opposée pendant quelques millisecondes. Si un capteur différent est activé pendant ce temps, il est ignoré.
- La sensibilité des photo-résistances décroît avec la luminosité. Ceci peut conduire le programme à croire que tout les capteurs reçoivent la même quantité de lumière car le seuil de 5% est compilé dans le programme. Les valeurs de lumière issues du Convertisseur Analogique Numérique pourraient être ajustées avec un filtre non linéaire pour compenser ces effets.

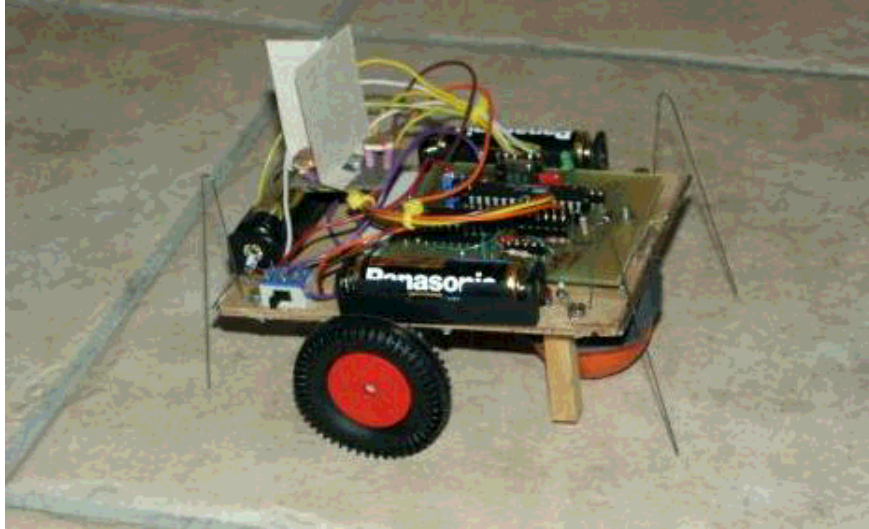
Dans sa version actuelle le programme du robot linux est suffisamment petit et simple pour être compris et étendu. Il n'utilise que 50% de la mémoire disponible du microcontrôleur 4433, ce qui laisse beaucoup de place pour des extensions.

Un point positif de ce robot est son matériel qui est générique: Il est constitué de deux moteurs et quelques capteurs attachés au microcontrôleur. Toute la logique est logicielle. Cela signifie que tout peut être changé à souhait.

Voici une photo du robot en position d'essai. Nous avons juste mis un bloc de post-it dessous afin d'éviter qu'il ne se sauve. La ligne RS232 est connectée pour des besoins de débogage.



... et le robot robot fini cherchant la lumière.....:



Références

- LinuxFocus Mars 2002, article 231: Programmer le microcontrôleur AVR avec GCC
- linuxrobot-0.1.tar.gz: Le logiciel et les schémas au format Eagle
- page de télé-chargement pour cet article: Data sheet du l293d et de possibles futures updates du logiciel linuxrobot peuvent être trouvé ici.

Site Web maintenu par l'équipe d'édition
LinuxFocus

© Katja et Guido Socher

"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

Translation information:

en --> -- : Katja et Guido Socher <katja@linuxfocus.org
<guido@linuxfocus.org>

en --> fr: John Perr <johnperr@linuxfocus.org>