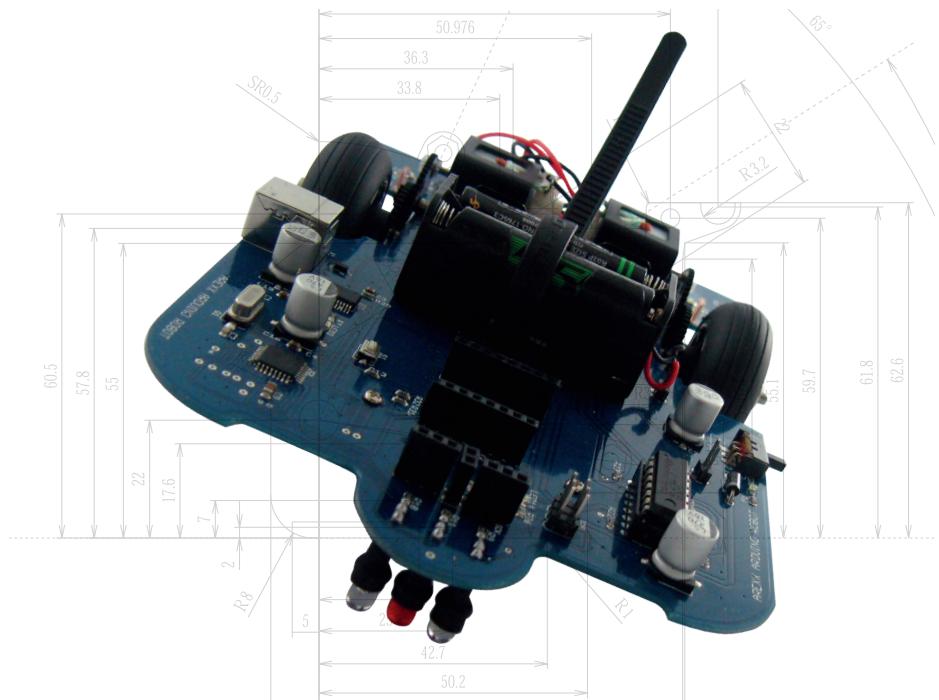




ROBOT ARDUINO D'AREXX

AAR

NOTICE DE MONTAGE: AAR-04



© AREXX - PAYS-BAS V1401

Table des Matières

1. DESCRIPTION DU ROBOT AAR	5
1.1 La famille des robots ARDUINO	5
1.2 Caractéristiques	5
2. INFORMATIONS GENERALES SUR ARDUINO	7
3 Le ROBOT ARDUINO D'AREXX	12
3.1 LED informations du ROBOT ARDUINO	12
3.2 Conception du PCB du robot	13
4. Premiers pas vers l'installation	16
4.1 Téléchargement et installation du logiciel Arduino	16
4.2 Le langage de programmation Arduino	16
4.3 Installation d'un driver USB	16
4.4 Connexion de l'accu	17
4.5 Programmation du robot avec le logiciel Arduino	18
4.6 Sélection d'un programme Arduino	18
4.7 Définition d'une interface COM	19
4.8 Transfert d'un programme dans le robot Arduino	20
4.9 Résolution de problèmes	21
4.10 Autotest de L'AAR ARDUINO	22
5. KITS D'EXTENSION POUR LE ROBOT ARDUINO	23
6. Informations de base concernant le montage en pont	27
7. Systèmes odométriques	27
8. Programmation du Bootloader	33
8. ANNEXE	34
A. Nomenclature	35
B. PCB principal face supérieure	37
C. PCB principal face inférieure	38
D. Schéma technique de l'AAR	39
E. PCB 3D de l'AAR	40

AREXX et AAR sont des marques déposées d'AREXX Engineering - PAYS-BAS.

© Traduction française/French translation (August 2012): AREXX Engineering (NL).
Ce manuel est protégé par les droits d'auteur. Toute reproduction ou copie même partielle est interdite sans l'accord écrit préalable de l'importateur européen:
AREXX Engineering - Zwolle (NL).

Le fabricant et le distributeur ne déclinent toute responsabilité pour les dommages causés par des erreurs de manipulation, d'installation et d'utilisation de ce produit suite au non respect des instructions de montage.

Sous réserve de modifications sans préavis.



Fabricant:
AREXX Engineering
JAMA Oriental



Importateur:
AREXX Engineering
ZWOLLE Pays-Bas

Support technique pour la construction du robot:

WWW.AREXX.COM
WWW.ROBOTERNETZ.DE

© AREXX Pays-Bas et JAMA Taiwan
© Traduction française: AREXX - Pays-Bas

Mentions légales

©2013 AREXX Engineering

Nervistraat 16
8013 RS Zwolle
The Netherlands

Tel.: +31 (0) 38 454 2028
Fax.: +31 (0) 38 452 4482

E-Mail: Info@arexx.nl

Ce mode d'emploi est protégé par les lois du copyright. Il est interdit de copier ou de reprendre tout ou partie du contenu sans l'autorisation préalable de l'éditeur! Sous réserve de modification des spécifications du produit et des accessoires fournis. Le contenu de ce manuel est susceptible de modifications sans préavis.

Vous trouverez des versions plus récentes de ce manuel gratuitement sur <http://www.arexx.com/>

"AREXX et AAR" sont des marques déposées d'AREXX Engineering.

Toutes les autres marques appartiennent à leurs propriétaires respectifs. Nous ne sommes pas responsables du contenu de pages internet externes qui sont citées dans ce manuel!

Notes concernant les limitations de garantie et de responsabilité

La garantie d'AREXX Engineering se limite au remplacement ou à la réparation du robot et de ses accessoires dans le cadre de la durée de garantie légale pour des défauts de fabrication avérés tels que des dommages mécaniques ou l'implantation manquante ou erronée de composants électroniques à l'exception de tous les composants connectés par fiches. Nous déclinons toute responsabilité pour des dommages causés directement ou indirectement par l'utilisation du robot. En sont exceptés les droits qui reposent sur les prescriptions légales inaliénables relatives à la responsabilité du produit. Tout droit à garantie s'éteint dès que des modifications irréversibles (p.ex. soudure d'autres composants, perçage de trous, etc.) ont été effectuées sur le robot ou ses accessoires ou si le robot est endommagé suite au non-respect de ce manuel. Lisez impérativement les consignes de sécurité!

Il n'existe aucune garantie que le logiciel fourni satisfasse des attentes individuelles ou fonctionne parfaitement sans interruption ni erreur.

Par ailleurs, le logiciel est librement modifiable et chargé par l'utilisateur dans l'appareil. C'est pourquoi l'utilisateur assume l'entièvre responsabilité quant à la qualité et la performance de l'appareil et de tous les logiciels.

AREXX Engineering garantit la fonctionnalité des exemples d'application fournis dans la mesure où les conditions spécifiées dans les caractéristiques techniques soient respectées. Si au-delà, l'appareil ou le logiciel pour PC s'avèrent être défaillants ou insuffisants, le client assume tous les frais de service, de réparation et de correction.

Nous vous invitons à prendre également connaissance des accords de licence sur le CD-ROM!

IMPORTANT

Avant la première utilisation du Robot, lisez d'abord attentivement le mode d'emploi! Il explique la bonne utilisation et vous informe des dangers éventuels. Par ailleurs, il contient des informations importantes qui ne sont pas forcément connues de tous.

Symboles

Les symboles suivants sont utilisés dans le manuel:



Le symbole "Attention!" attire l'attention sur des consignes particulièrement importantes qui doivent absolument être respectées. Une erreur risque d'entraîner la destruction du robot ou de ses accessoires ainsi que mettre en danger votre santé ou celle de votre entourage!



Le symbole "Attention!" attire l'attention sur des consignes particulièrement importantes qui doivent absolument être respectées. Une erreur risque d'entraîner la destruction du robot ou de ses accessoires ainsi que mettre en danger votre santé ou celle de votre entourage!

Consignes de sécurité

- Vérifiez la polarité de piles.
- Tenir le robot toujours au sec. Si l'appareil a été mouillé, débranchez immédiatement l'alimentation ou retirez les piles.
- En cas de non-utilisation prolongée, débranchez l'appareil du secteur. ou retirez les piles
- Avant la mise en service du module, contrôlez son état général ainsi que celui des câbles.
- Si vous pensez qu'un fonctionnement en toute sécurité n'est plus garanti, débranchez-le immédiatement du secteur et assurez-vous qu'il ne peut pas être utilisé par inadvertance.
- Demandez conseil à un spécialiste si vous avez des doutes concernant l'utilisation, la sécurité ou le branchement du module.
- Ne jamais utiliser le module dans des conditions défavorables.
- Ce module contient des composants très sensibles, notamment à l'électricité statique. Ne touchez le module que par les bords et évitez tout contact direct avec les composants sur la platine.

Utilisation normale

Cet appareil est une plateforme d'expérimentation destinée aux personnes intéressées par la robotique. L'objectif principal consiste à apprendre la programmation du robot en langage 'C'. Ce produit n'est pas un jouet et ne convient pas aux enfants et adolescents de moins de 14 ans.

Uniquement pour utilisation à l'intérieur. Tenir à l'abri de l'humidité et de l'eau. Si le produit passe d'une pièce froide dans une pièce chaude, attendez qu'il ait atteint la température ambiante avant de le mettre en service. Toute utilisation autre que celle décrite ci-dessus, peut entraîner des dommages et des risques tels que des courts-circuits, incendies, chocs électriques, etc.

1. DESCRIPTION DU ROBOT AAR

1.1 La famille des robots ARDUINO

Arduino est une plateforme „open source“¹ destinée au développement de prototypes électroniques qui met à notre disposition un microcontrôleur ainsi que toutes les interfaces périphériques et logiciels nécessaires.

Le concept Arduino fut développé afin d’appréhender de manière simple l’électronique moderne qui est utilisée dans le monde de la robotique, du pilotage de logiciels et des détecteurs.

En tant que successeur du robot ASURO qui est programmable en langage ‘C’, le robot Arduino ressemble certes fortement à son pré-décesseur mais sa programmation est considérablement plus simple en raison du langage de programmation „open source“ Arduino.

¹ Open Source est une gamme de licences de logiciel dont les sources sont publiquement accessibles afin de promouvoir son développement futur.

1.2. Caractéristiques:

Moteurs	2 moteurs de courant continu (3 Volts)
Processeur	ATmega328P
Langage de programmation	ARDUINO
Alimentation	4 batteries ou piles AAA 4,8 - 5,5 Volts max.
Consommation	Min. 10 mA Max. 600 mA
Communication	Port USB
Extension	les extensions ASURO sont compatibles
Hauteur	40 mm
Largeur	120 mm
Profondeur	180 mm

1.3 Mises en Garde

1. Lisez d'abord ce manuel avant de brancher une source de tension sur l'un des connecteurs! Des connexions erronées peuvent endommager le matériel.
2. Vérifiez attentivement l'affectation des broches! Soyez particulièrement méticuleux lors du câblage du système. Des erreurs de branchement risquent d'endommager des composants. Respectez la polarité des bornes d'alimentation. Une inversion de polarité des bornes d'alimentation risque d'endommager les circuits.
3. N'utilisez pas de systèmes d'alimentation dont les tensions dépassent les valeurs spécifiées!
Utilisez des systèmes d'alimentation stabilisés et filtrés afin d'éviter des crêtes de tension.
4. La platine ne présente aucune protection contre les effets d'eau et d'humidité. Conservez le robot dans un endroit sec.
5. Evitez les courts-circuits avec d'autres objets métalliques ainsi que toute surcharge de la platine ou des connecteurs par pression, traction ou chargement.
6. Evitez les décharges électrostatiques² (voir à ce sujet les précautions, mises en garde et l'article sur Wikipedia "Décharge électrostatique").

1.4 Généralités

- * Le droit de retour s'éteint avec l'ouverture du sachet qui contient les composants et pièces.
- * Lisez attentivement la notice de montage avant de commencer la construction.
- * Manipulez les outils avec précaution.
- * Ne construisez pas le robot en présence d'enfants en bas âge. Les enfants risquent de se blesser avec les outils et/ou avaler de petits composants.
- * Respectez la polarité des piles.
- * Veillez à ce que les piles et le support de piles restent toujours sec. Si le robot est mouillé, retirez les piles et séchez soigneusement tous les composants.
- * Retirez les piles si le robot n'est pas utilisé pendant plus d'une semaine.

² Angl. *electrostatic discharge*, abrégé ESD

2. INFOS GENERALES SUR ARDUINO

2.1. Qu'est-ce qui ou qu'est-ce qu'est ARDUINO?

Arduino est un microcontrôleur mono-platine open-source destiné notamment aux artistes, designers, bricoleurs et autres personnes intéressées afin de leur faciliter l'accès à la programmation et aux microcontrôleurs ainsi qu'au travail sur des projets d'objets interactifs.

La plateforme Arduino est basée sur un microcontrôleur d'Atmel ATmega168 ou ATmega-328. Le système met à la disposition de l'utilisateur aussi bien des entrées//sorties numériques que des entrées analogiques qui permettent au système Arduino de recevoir des signaux de son environnement et d'y réagir.

Différentes plaques Arduino sont commercialisées telles que Arduino Uno, Arduino LilyPad et Arduino Mega 2560. Etant donné que chaque plaque Arduino possède ses propres spécificités, il existe un système Arduino idéal pour pratiquement chaque projet.

Les signaux d'entrées peuvent provenir de commutateurs, détecteurs de lumière, de mouvement, de distance ou de température. Même des commandes provenant d'internet peuvent constituer des signaux d'entrée. Les signaux de sortie peuvent, à leur tour, piloter des moteurs, lampes, pompes et moniteurs.

Pour la programmation, le système possède un compilateur assurant un langage de programmation normalisé et un bootloader. Le langage de programmation est basé sur le langage de programmation Wiring qui correspond à C++.

En 2005, Arduino a démarré comme un simple projet à Ivrea en Italie. L'objectif déclaré était initialement le soutien des étudiants lors de leurs travaux sur des projets. La création du prototype devait alors être bien moins coûteuse qu'avec des méthodes similaires traditionnelles.

Le groupe de développeurs autour de Massimo Banzi et David Cuartielles a nommé le projet d'après un personnage historique du nom de 'Arduino d'Ivrea'. Le mot 'Arduino' signifie 'Ami musclé'.

2.2 Microcontrôleur!

2.2.1 Applications

Un microcontrôleur (parfois appelé également sous sa forme abrégée µC, uC ou MCU) est un petit ordinateur dans un circuit individuel intégré qui contient le noyau du processeur, la mémoire et un jeu de connecteurs d'entrée et de sortie programmables.

Une mémoire de programme et une petite mémoire de données RAM (c'est-à-dire Random Access Memory) font souvent partie du circuit intégré. Les microcontrôleurs sont utilisés dans des installations et systèmes à pilotage automatique tels que dans le pilotage de moteurs, implants, télécommandes, systèmes bureautiques, jouets et outils de haute puissance.

L'économie de poids et de coûts due à l'intégration d'un microprocesseur, d'une mémoire et de connecteurs d'entrée/sortie sur un seul circuit intégré a pour conséquence que la commande digitale devient de plus en plus économique pour un grand nombre d'applications.

Un ménage typique dans un quartier moderne dispose de quatre microprocesseurs et de trois douzaines de microcontrôleurs. Une voiture de classe moyenne utilise 30 ou plus de microcontrôleurs. Ces composants sont également présents dans de nombreux appareils électriques tels que des machines à laver, fours à micro ondes et téléphones.

2.3. Consommation et Vitesse

Certains microcontrôleurs travaillent à une faible fréquence d'impulsions de 4 kHz et présentent une faible consommation dans la zone des milliwatts ou microwatts. Ils sont généralement activables par pression sur un bouton ou par une interruption. La consommation lors de l'attente (générateur d'horloge CPU ainsi que la quasi-totalité des circuits complémentaires à l'arrêt) se situe dans le domaine des nanowatts ce qui garantit la longévité des piles. D'autres microcontrôleurs sont plutôt utilisés dans le domaine des hautes performances où ils sont employés p.ex. comme processeurs de signaux numériques (DSP) avec des taux de fréquence et une consommation plus élevés.

Le système Arduino travaille avec un puissant CI unique Atmel ATmega328P qui est équipé d'un microcontrôleur 8 octets (d'une fréquence de 16MHz) et d'une mémoire flash ISP de 32K bytes. L'alimentation est relativement flexible sur une plage de 7-12Vcc ce qui permet d'assurer des conditions de travail stables et correctement sécurisées pour le circuit intégré ainsi que des conducteurs séparés jusqu'à 2A pour l'alimentation des moteurs.

2.4 Programmes pour Microcontrôleurs

Le logiciel pour le microcontrôleur doit tenir dans l'espace mémoire disponible sur le circuit intégré car une mémoire supplémentaire externe serait trop coûteuse. Le compilateur et l'assembleur sont optimisés pour la traduction du langage de haut niveau et des codes d'assemblage en commandes machine compactes qui sont déposées dans la mémoire du microcontrôleur.

En fonction du type de circuit intégré, le programme peut être sauvegardé dans une mémoire ROM³ permanente qui n'est écrite que pendant la fabrication du circuit, ou bien dans une mémoire Flash ré-inscriptible à dessein ou dans une mémoire ROM ré-inscriptible plusieurs fois.

Initialement, les microcontrôleurs ont seulement été programmés en assembleur mais actuellement il existe d'autres langages de programmation d'un niveau plus élevé. Il s'agit soit de langages spéciaux ou bien de variantes des langages de haut niveau tel que C. Les vendeurs des microcontrôleurs proposent souvent des outils gratuits afin de faciliter la mise en place du matériel. Le système Arduino met à notre disposition des mémoires flash 32 kBytes pour les programmes sketch qui sont programmables en C.

2.5 Architecture des Interfaces

Les microcontrôleurs disposent en général d'une multitude de broches d'entrée/sortie (GPIO) librement programmables. Ces connecteurs GPIO sont programmables en broches d'entrée ou de sortie au moyen de commandes logicielles. Lorsqu'elles sont programmées en broches d'entrée, elles sont souvent utilisées pour la lecture de détecteurs ou de signaux externes. Si elles sont programmées comme broches de sortie, elles sont utilisables pour le pilotage de LED ou de moteurs.

Un grand nombre de systèmes embarqués a besoin de savoir lire les signaux analogiques des détecteurs. A cet effet, on utilise les convertisseurs analogiques/digitaux (en anglais ADC⁴).

Etant donné que les processeurs ont été développés spécialement pour le traitement de données numériques, c'est-à-dire des zéros et des uns, ils ne conviennent pas aux signaux analogiques des détecteurs. C'est le rôle des convertisseurs A/D de transformer les données rentrantes en une forme lisible pour un processeur.

³ ROM = Read Only Memory (mémoire morte)

⁴ Analog-to-digital converter (ADC)

Une forme moins usuelle de la conversion de données dans le microcontrôleur est la conversion digitale en analogique (ou DAC⁵) qui permet au processeur de générer des signaux analogiques ou des niveaux de tension.

En plus des convertisseurs, certains systèmes embarqués disposent de minuteries. Un des types de minuteries les plus courants est l'horloge à interval programmable (PIT⁶) qui décompte à partir d'une valeur de départ vers zéro. Lorsque le zéro est atteint, l'horloge envoie une interruption au processeur pour signaler que le temps réglé s'est écoulé. C'est une fonction utile p.ex. pour des thermostats qui enregistrent régulièrement la température ambiante et qui doivent contrôler s'il est nécessaire d'enclencher la climatisation ou bien le chauffage.

Le module émetteur-récepteur asynchrone universel UART⁷ nous permet d'échanger des données de communication avec d'autres systèmes via une connexion série sans trop solliciter le processeur.

Un matériel spécial embarqué dans le circuit intégré permet également la communication avec d'autres CI dans des formats numériques tels que I2C et Serial Peripheral Interface (SPI).

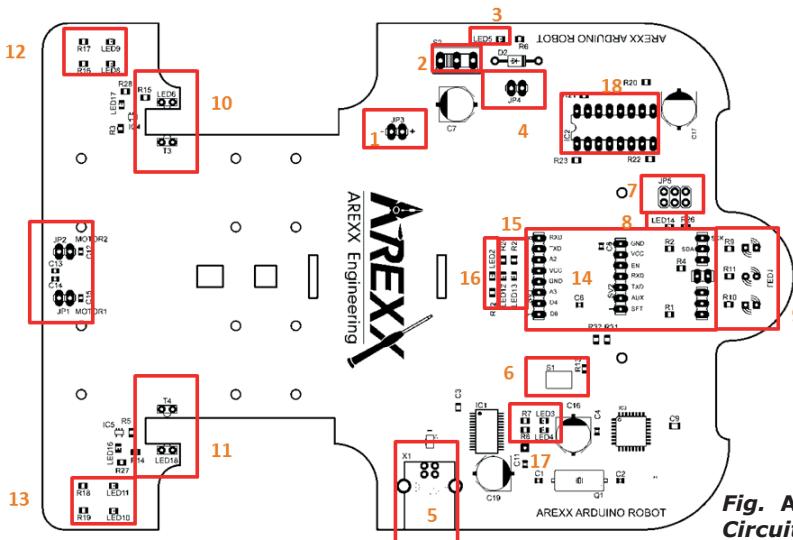
Le système Arduino met à notre disposition 14 connecteurs digitaux I/O et 7 connecteurs analogiques I/O (I/O signifie Entrée/Sortie).

⁵ Digital-to-analog converter (DAC)

⁶ Programmable Interval Timer (PIT)

⁷ Universal Asynchronous Receiver/Transmitter (UART)

3. Le ROBOT ARDUINO d'AREXX



**Fig. AAR PCB:
Circuit imprimé
de l'AAR**

3.1 LED informations du ROBOT ARDUINO

LED 1.	Rouge	Capteur de ligne (fond)
LED 2.	Bleu	Librement programmable
LED 3.	Rouge	DONNEES USB
LED 4.	Vert	DONNEES USB
LED 5.	Bleu	LED d'alimentation
LED 6.	IR	Capteur à LED infrarouge de la roue gauche
LED 7.	NA	NA
LED 8.	Rouge	Indicateur moteur vers l'avant gauche
LED 9.	Rouge	Indicateur moteur vers l'arrière gauche
LED 10.	Rouge	Indicateur moteur vers l'arrière droit
LED 11.	Rouge	Indicateur moteur vers l'avant droit
LED 12.	Vert	Transmission serielle
LED 13.	Rouge	Réception serielle
LED 14.	Bleu	Librement programmable
LED 15.	NA	NA
LED 16.	Rouge	Impulsion du capteur de la roue droite
LED 17.	Rouge	Impulsion du capteur de la roue gauche
LED 18.	IR	Capteur à LED infrarouge de la roue droite

- LED 14 Clignote seulement avec le bootloader
 LED 14 est ETEINTE sans le bootloader
 LED 14 clignote brièvement après la mise sous tension lorsqu'un programme est chargé.

3.2. Conception du PCB du robot ARDUINO

Voir chiffres dans la fig.: AAR PCB

1. Connecteur pour le compartiment à pile
(Respectez bien la polarité!)
2. Interrupteur M/A du robot.
3. LED d'état : indique que le robot est alimenté en tension.
4. Si vous utilisez les batteries rechargeables, vous pouvez brancher cette double fiche qui alimentera le robot avec la bonne tension
5. Connecteur USB : permet de programmer le robot à l'aide du logiciel Arduino
6. Bouton de remise à zéro : pour remettre manuellement le robot à zéro.
7. Connecteur ISP: vous permet d'installer un autre programme bootloader.
8. LED 14: laisse le libre accès à la programmation et clignote lorsque le bootloader est (re)démarré.
9. Suiveur de ligne : Ce module donne libre accès à la programmation et permet au robot de suivre des tracés.
10. Capteur de roue gauche : ce module génère des impulsions proportionnelles à la rotation de la roue gauche.
11. Capteur de roue droite : ce module génère des impulsions proportionnelles à la rotation de la roue droite.
12. LED d'état pour le moteur de gauche : ces LED signalent la rotation vers l'avant et vers l'arrière.
13. LED d'état pour le moteur de droite : ces LED signalent la rotation vers l'avant et vers l'arrière.
14. Connecteur pour la plaque d'extension sur lequel vous pouvez installer p.ex. un module sans fil APC220 ou un module Snake Vision et le connecter sur le système Arduino.
15. LED d'état pour l'interface de communication RS232.
16. LED d'état 2: LED librement accessible pour la programmation.
17. LED d'état pour la communication des données via USB.
18. Contrôleur du moteur

Sur le CD se trouve le logiciel d'auto-test AAR **AAR_SelftestV3** qui vous permet de tester toutes les fonctions de base du robot AAR.

3.2 Informations de fond sur l'AAR

L'AAR est un robot Arduino qui a été spécialement développé pour l'apprentissage du logiciel Arduino. Sur l'avant se trouve une interface USB avec un circuit intégré FT232 qui transforme le signal USB en un signal UART RS232 que le processeur ATMEGA328P (sur l'avant droite) arrive à traiter.

Un commutateur Marche/Arrêt avec l'emplacement JP3 pour l'alimentation et le circuit intégré moteur 2 ont été montés sur le côté opposé. Au dos du circuit imprimé se trouvent les deux moteurs et les capteurs de roue.

Les capteurs de roue contiennent une barrière lumineuse. Les roues dentées comportent quatre trous qui ont été disposés à 90°. Dès que la lumière de la LED touche le capteur, le capteur de roue signale au processeur une impulsion pour la roue concernée. La LED16 ou bien 17 est activée pour donner une indication visuelle. De cette manière la vitesse de rotation des roues arrières individuelles se mesure d'une façon relativement précise.

Sur l'avant se trouvent les emplacements de connexion pour les platines d'extension et en-dessous du circuit imprimé se situent les capteurs pour le suiveur de lignes. Le suiveur de ligne comporte une LED qui éclaire le sol au centre. A côté d'elle sont disposés deux détecteurs à infrarouge qui enregistrent la lumière réfléchie par le sol. Par ailleurs, le circuit imprimé comporte les composants nécessaires (LED, résistances et condensateurs) pour rendre le module de suite fonctionnel.

Le robot fut équipé d'un circuit imprimé Arduino qui est comparable à la platine Arduino Duemilanove. Le cœur du système Arduino est l'ATMEGA328P. Ce microcontrôleur dispose de 14 entrées et sorties numériques dont 6 peuvent servir comme sorties à modulation de largeur d'impulsion (PWM). En outre, le robot possède 6 entrées analogiques, un oscillateur cristallin 16MHz et un port USB pour la programmation et le contrôle. Il offre en plus un connecteur ISP avec lequel les amateurs plus expérimentés peuvent programmer eux-mêmes un bootloader.

Le robot travaille avec un tension embarquée de 5V et se contentera éventuellement d'une source d'alimentation USB ce qui économise quelques étapes lors des essais et de la programmation. Dans ce concept robotique, les emplacements réservés à vos propres pla-tines d'extension ou celles d'AREXX de la série ASURO sont tout particulièrement appréciables.

3.3 INFORMATIONS DE BASE SUR LE LOGICIEL ARDUINO

Le logiciel Arduino fait partie des open source et est par con-séquent accessible à tous. C'est pourquoi les codes source de l'environnement de programmation sont ouverts à tous.

L'environnement Arduino comporte un éditeur de texte, une fenêtre de message et une console de texte. L'environnement de pro-grammation peut communiquer directement avec l'AAR afin de trans-férer des programmes dans le processeur de la manière la plus simple. Les programmes écrits en arduino sont appelés "sketches" ("esquis-ses" en français). Le code source est écrit avec l'éditeur de texte. Le fichier sketch est enregistré avec l'extension ".ino" sur le disque dur du PC.

Dans la fenêtre s'affiche le message que le fichier a été enregistré et éventuellement un message inquant une erreur dans le code source. En bas à gauche de l'écran, le tableau de bord arduino actuel et l'interface série sont représentés.

Arduino dispose de bibliothèques („libraries“) avec des fonctions supplémentaires. Une bibliothèque ("library") est un paquet conte-nant différentes fonctions composées qu'il n'est plus nécessaire de réécrire à chaque fois. Il suffit d'appeler simplement ces fonctions en arduino.

Un programme arduino se divise en trois parties: structure, définiti-ons variables ou constantes et fonctions.

Une structure arduino se compose d'un setup et d'une boucle (la fonction "loop"). Le setup est responsable de l'initialisation des va-riables, des réglages pin ("pin modes") et des bibliothèques ("libra ries").

Le boucle ("loop") est constamment répétée afin que le programme puisse réagir à répétition. Les définitions "variables" sont néces-saires à l'enregistrement des données alors que des constates sont utilisées p.ex. pour définir une broche comme entrée ou sortie et de faire en sorte que la broche reçoive une tension.

4. Premiers pas vers l'Installation

4.1 Téléchargement et Installation du Logiciel Arduino

Installez tout d'abord la première version du logiciel Arduino sur le CD afin que nous soyons sûrs que cette version fonctionne correctement. Plus tard, vous pourrez visiter le site internet Arduino et y télécharger des versions plus récentes.

IMPORTANT:

Il est possible que certaines versions du logiciel ARDUINO créent des problèmes avec certaines versions du logiciel d'application. Il peut donc s'avérer nécessaire de retravailler les programmes d'application après la mise à jour du logiciel ARDUINO (notamment en cas de dysfonctionnements).

4.2 Le langage de programmation Arduino

La syntaxe du langage de programmation Arduino est documentée sur le site internet officiel Arduino. Approfondissez vos connaissances de ce langage de programmation dans la mesure cela s'avère nécessaire.

4.3 Installation d'un driver USB

Lors de la première connexion de la platine Arduino avec le PC, Windows devrait démarrer le processus d'installation du driver. Dans Windows Vista et autres installations Windows plus récentes, le driver est normalement téléchargé automatiquement d'internet et installé.

Sélectionnez l'interface sérielle pour la platine Arduino dans le menu Tools > Serial Port. Généralement, le système propose COM3 ou plus. COM1 et COM2 sont normalement réservés aux ports série définis dans le matériel. Pour le savoir, vous pouvez débrancher le système Arduino et rouvrir le menu en question. La nouvelle entrée est probablement le bon port. Rebranchez la platine et sélectionnez ce port série.

4.4 Le Branchement d'Accumulateurs

Le robot a été dimensionné pour une alimentation d'un conteneur de piles avec quatres piles de 1,5V. Si vous utilisez des batteries rechargeables à la place, vous devez relier le pont JP4.
(Voir fig. 1, no. 4).

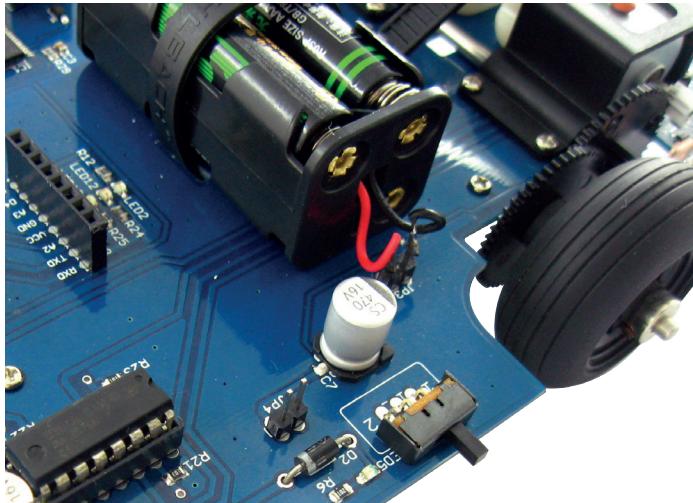


ATTENTION!

Dès la mise en place du pont JP4, la protection contre l'inversion de polarité est désactivée. Une erreur de polarité endommagera irrémédiablement le robot.

Branchez le conteneur à piles tel que représenté dans la figure 2.

Fig. 2:
Branchement
du conteneur
à piles



Mettez le robot sous tension au moyen du commutateur. La LED5 à côté du commutateur s'allume.

4.5 Programmation du robot avec le logiciel Arduino.

Branchez le robot sur le PC au moyen d'un câble USB.

Si le robot est connecté sur le port USB, Arduino ne requiert pas forcément une tension provenant d'une pile. Le connecteur USB du PC remplit alors la fonction d'alimentation.

ATTENTION:

Le robot est systématiquement sous tension lorsqu'il est connecté sur le PC. Le commutateur et la LED5 ne sont actifs qu'en cas d'alimentation par piles.



Ouvrez maintenant le logiciel Arduino (voir fig. 3).

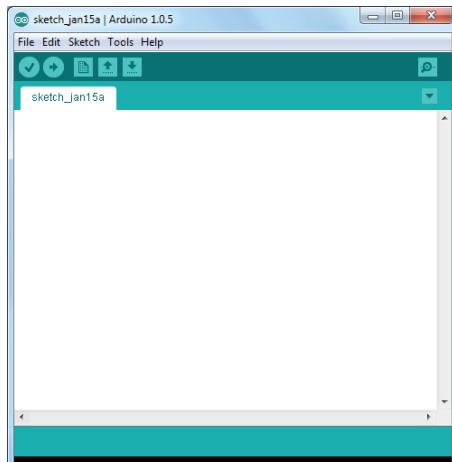


Fig. 3a Logiciel Arduino

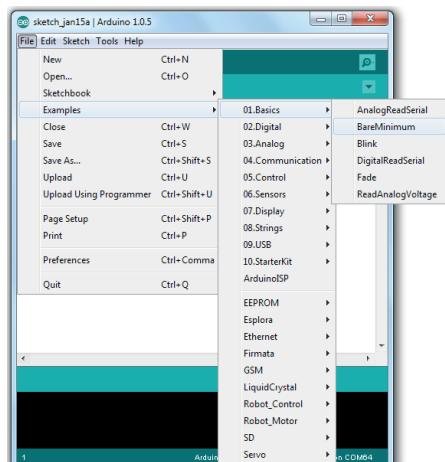


Fig. 3b Ouverture du programme "blink"

4.6 Sélection d'un programme Arduino

A titre d'exemple simple, nous commençons par charger le programme "blink" dans le robot qui déclenche le clignotement de la LED1.

A cet effet, cliquez sur File>Examples>1. Basics>Blink (voir fig. 3b) dans le logiciel Arduino de façon à ce que les informations suivantes s'affichent sur l'écran (fig. 4a).

```

Blink | Arduino 1.0.5
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeating.
This example code is in the public domain.
*/
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
}

```

Fig. 4a Programme Blink

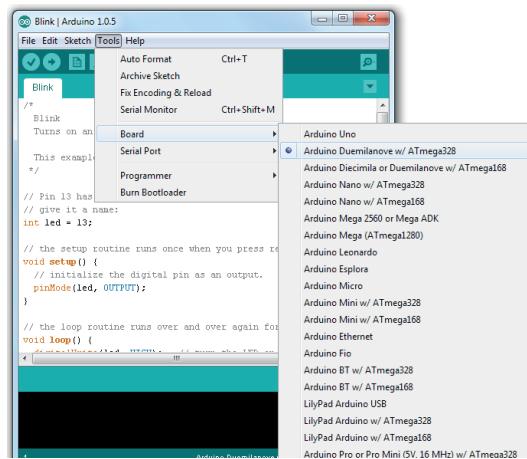


Fig. 4b Définir (sélectionner) Board

Nous devons définir maintenant le bon type de platine Arduino.

Cliquez sur Tools>Board> Arduino Duemilanove or¹¹ Nano w/ Atmega328 (voir fig. 4b)

4.7 Définition de l'interface COM

L'étape suivante définit la bonne interface COM dans le logiciel Arduino. La bonne interface COM (port COM) pour le branchement du robot est le COM 12.

Pour la définition de l'interface COM suivez le menu: Tools>Serial Port>COM 12.
(voir fig. 5)

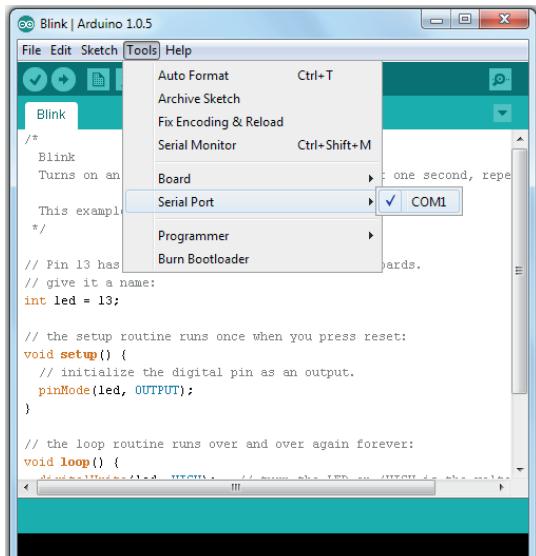


Fig. 5

Définition du port COM

¹¹Terme anglais pour: „ou“

4.8 Transmission d'un Programme vers le robot Arduino

Cliquez ensuite sur la touche marquée par la flèche rouge (ou bien dans le menu „File>Uploading to I/O board“) pour transférer ce programme dans le robot Arduino connecté (voir fig. 6).

Dans la barre d'état le logiciel signale que le système compile et démarre ensuite un upload.

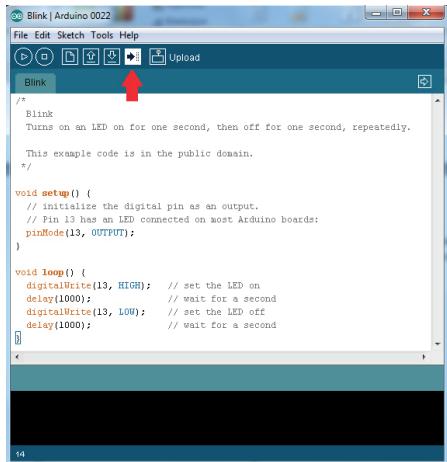


Fig. 6 Transfert d'un programme dans le robot Arduino (processus upload).

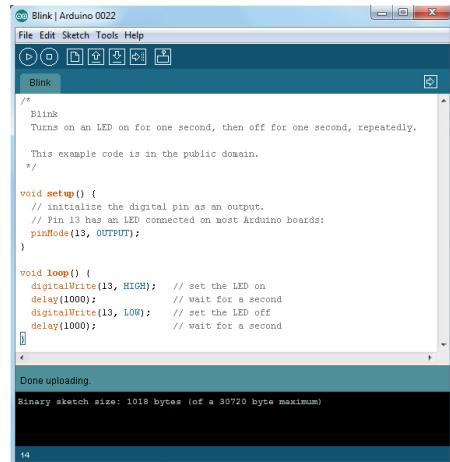


Fig. 7 „Done uploading“ - Message: „Processus de téléchargement terminé“

Vous pouvez maintenant débrancher le robot du PC en retirant le cordon USB, en branchant la tension de batterie et en démarrant le robot.

Pour d'autres informations et téléchargements nous vous invitons à visiter les forums
sur les sites internet:
www.arexx.com --> Forum

4.9 Résolution de problèmes

TOUT D'ABORD vérifiez si le robot est suffisamment alimenté par le port USB. Pour cela, retirez les piles et fermez le jumper JP4. La LED d'alimentation devrait être allumée maintenant.

PROBLEME; Aucune connexion avec le robot AAR.

Vérifiez si les LED de données sur le robot (LED 3 et 4) clignotent brièvement lorsque vous branchez le cordon USB sur le robot.

Vérifiez si le robot est connecté sur votre PC dans votre gestionnaire de matériel.

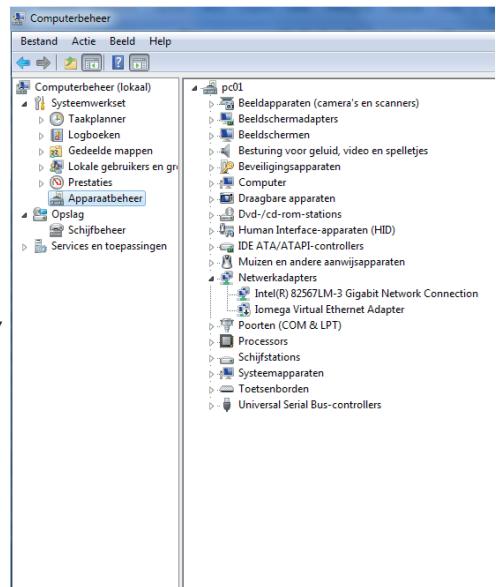
Abb. 6c).

Gestionnaire de matériel



Le port FTDI du robot s'affiche normalement dans la liste des ports comme. **"USB SERIAL PORT"**

Vérifiez si le driver FTDI est installé. C'est probablement la cause du problème. Téléchargez et installez le driver FTDI le plus récent sur votre PC.



PROBLEME; IMPOSSIBLE DE TELECHARGER UN FICHIER .INO DANS LE ROBOT.

Vérifiez si vous utilisez le bon port COM. Il sera différent du COM1 et COM3.

Vérifiez si vous avez sélectionné le bon type de plaque Arduino avec le bon type de processeur (ATmega328).

Regardez dans le menu Sketch si vous avez installé toutes les bibliothèques nécessaires : Voir le point de menu "sketch-->Library --> (page 22).

4.10 AUTOTEST DE L'AAR ARDUINO (SELFTEST)

Ouvrez le programme d'autotest AAR qui se trouve sur le CD et chargez le fichier .ino dans le robot AAR. Cliquez sur le bouton qui est identifié par une flèche rouge (ou bien allez dans la barre de menu « File>Uploading to I/O board ») afin de transférer le programme sélectionné dans le robot Arduino (voir fig. 7).

Chargez les bibliothèques:

Menu "Sketch"-->"Library" --> **SoftPWM** et **TimerOne**
du CD (Examples).

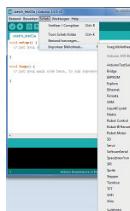


Fig.6d) Téléchargement du logiciel

```
/* Blink | Arduino 1.0.5
File Edit Sketch Tools Help
Blink Upload
Moniteur série! */

/*
Blink
Turns on an LED on for one second, then off for one second, repea
This example code is in the public domain.
*/
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
```

Fig. 7 Téléchargement du logiciel

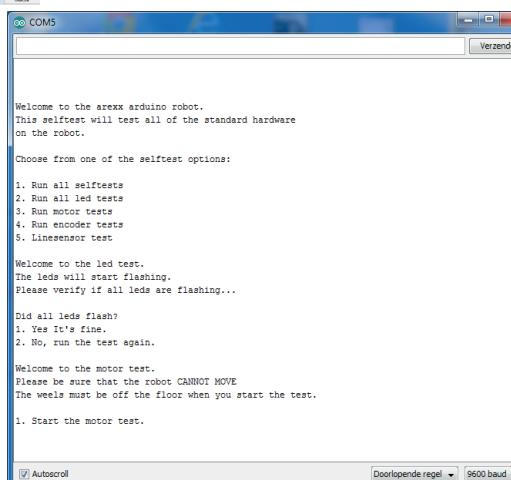


Fig. 8 Moniteur série!

Ouvrez le moniteur série comme indiqué dans la Fig 8.

Laissez le cordon USB branché sur le robot!

Assurez-vous que les batteries sont bien en place et chargées.
Mettez le robot sous tension.

Commencez par le premier test et suivez les étapes sur le moniteur série.

5. KITS D'EXTENSION POUR LE ROBOT ARDUINO

5.1 Kits d'extension ASURO pour le robot Arduino

Le robot Arduino d'AREXX possède différents connecteurs destinés aux modules d'extension. Les connecteurs AAR sont également compatibles avec le robot ASURO d'AREXX (un robot en kit qui est programmé en langage C).

Cela signifie que tous les kits ASURO d'AREXX sont également compatibles avec le robot Arduino. Sur le CD se trouvent quelques programmes d'exemple pour Arduino afin que vous puissiez commencer directement avec les kits d'extension.

Vous trouverez ci-dessous le schéma d'affectation des pins du connecteur d'extension AAR. Il s'agit des connecteurs SV8, SV9 et SV10 sur la platine.

SV1								SV8, SV9 and SV10								SV2						
1.	2.	3.	4.	5.	6.	7.	8.	1.	2.	3.	4.	5.	6.	7.	8.	1.	2.	3.	4.	5.	6.	7.
1.	D8							1.	SCK/PB5							1.	SET					
2.	D4							2.	SCL/ADC5/PC5							2.	AUX					
3.	A3							3.	WS_RIGHT/INT0/PD2							3.	TXD					
4.	GND							4.	RGND							4.	RXD					
5.	VCC							5.	PWM1_L/AIN0/OC0A/PD6							5.	EN					
6.	A2							6.	MOSI/OSC2A/PB3							6.	VCC					
7.	TXD							7.	MISO/PB4							7.	GND					
8.	RXD							8.	SDA/ADC4/PC4													

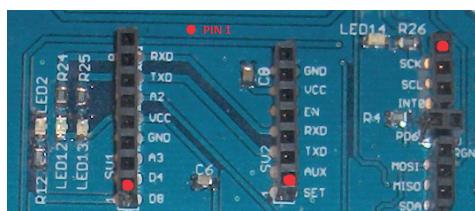
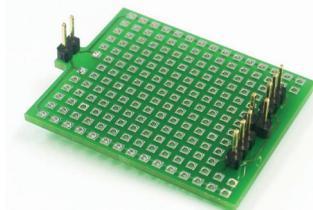


Fig. 9. Configurations des kits d'extension.

Point Rouge 1.



5.2 Configurations des kits d'extension ASURO.

ARX-ULT10

KIT ULTRASONS D'AREXX

Utilisez le programme d'exemple Arduino **AAR_ULTRA** sur le CD.

Avec ce programme, le robot recule lorsqu'il détecte un objet.

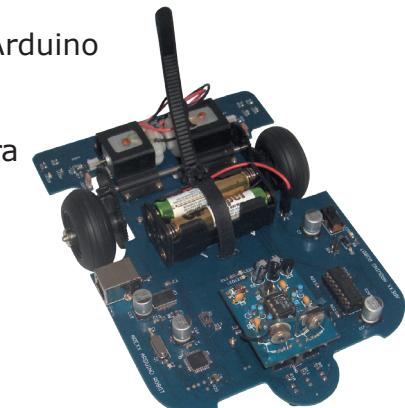


ARX-SNK20

KIT SNAKE VISION (PYRO) D'AREXX

Utilisez le programme d'exemple Arduino **AAR_SNAKE**, sur le CD.

Avec ce programme, le robot suivra un objet chaud.



ARX-DSP30

KIT D'AFFICHAGE AREXX

Utilisez le programme d'exemple Arduino **AAR_DISPLAY** sur le CD.

Avec ce programme, du texte s'affichera sur l'écran.

Les boutons-poussoir sont librement programmables.



ARX-MNSP55

KIT DE DEMINAGE D'AREXX

Utilisez le programme d'exemple Arduino **AAR_MINE** sur le CD.



Avec ce programme, le robot peut détecter du métal.

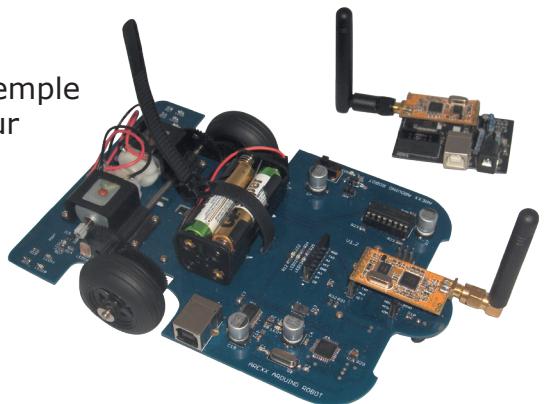
ARX-APC220

KIT SANS FIL D'AREXX

Sur le CD se trouvent quelques programmes d'exemple **AAR_SRC**

Utilisez le programme d'exemple Arduino **AAR_INO.SRC**, sur le CD

Il vous faut également le programme d'application Visual Basic pour le PC (**AAR_VB**) sur le CD.



Le kit ARX-APC-220 vous permet de contrôler l'AAR sans fil.

Ci-dessous nous vous expliquons pas à pas comment ça marche :

- Branchez le RP6v2 USB Programmer et l'APC-220 sur le PC.
- Branchez l'autre APC sur l'AAR (connecteur central).
- Chargez les données AAR_SRC dans le processeur AAR.
- Installez le logiciel AAR_VB sur votre PC.
- Sélectionnez le bon port COM dans le logiciel AAR-VB.

PROGRAMME ANDROID

ARX-BT3

KIT BLUETOOTH D'AREXX

Notre kit ARX-BT03 vous permet de piloter le Robot Arm via Bluetooth à l'aide d'une application ANDROID.

Utilisez le programme d'exemple Arduino **BLUETOOTH_AAR.APK** sur le CD. Installez également l'application **AAR_Android** sur votre appareil.

Suivez les instructions ci-dessous:

- Branchez le module Bluetooth sur le PCB du robot.
- Chargez les données ANDROID HEX dans le processeur du robot.
- Chargez des données ANDROID APK dans votre smartphone ou tablette.



Bluetooth®



* Vous trouverez ces données sur le CD et dans le Google Play Store.



6. Informations de base concernant le montage en pont

Un montage en pont est un circuit électrique permettant p.ex de changer la polarité d'un moteur à courant continu à l'aide de 4 dipôles. Un tel circuit est souvent utilisé en robotique pour piloter des moteurs dans deux directions.

Dans la plupart des cas, il existe des modules intégrés pour ce type de circuits mais afin d'expliquer le fonctionnement et le dimensionnement d'une alimentation, il peut être intéressant d'étudier un circuit plus acien.

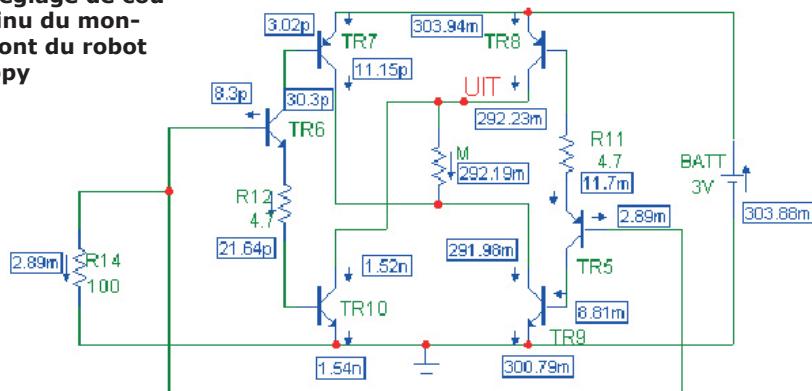
6.1 Montage en pont pour alimentations 3 Volts

Dans l'étage final du robot Hyperpeppy, les quatre circuits du pont ont été conçus comme deux transistors PNP TR7 et TR8 ou bien NPN TR9 et TR10. Dans ce circuit, seuls deux de ces transistors sont autorisés à laisser passer le courant de telle sorte que le courant passe par le moteur M

- via TR7 et TR10 ou bien
- via TR8 et TR9.

A l'aide du simulateur microcap (gratuit) nous pouvons faire calculer très simplement le réglage de courant continu de ce circuit et le lire sur le schéma.

Fig. 8: Réglage de courant continu du montage en pont du robot Hyperpeppy



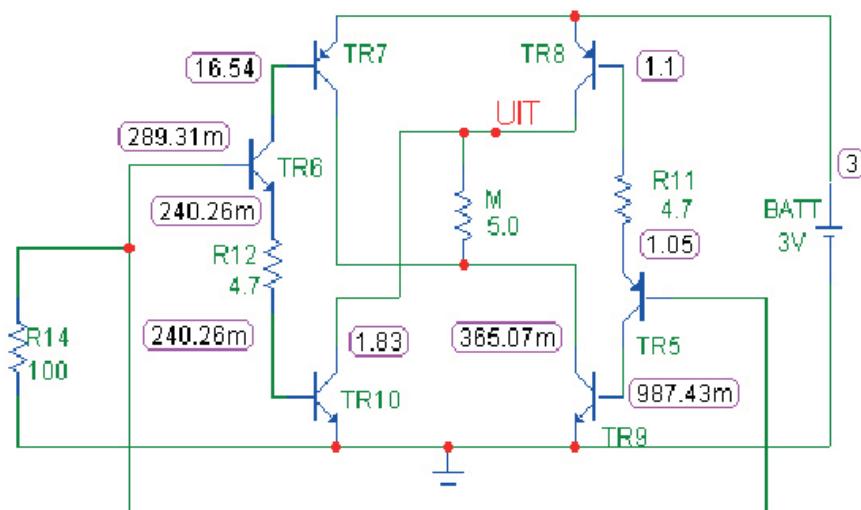
A l'étage final M est le moteur et l'étage de contrôle du pré-amplificateur est simulé par une résistance R14. Cette résistance R14 relie les bornes de base des transistors TR6 et TR7 à la terre. C'est pourquoi seul l'étage final du côté droit conduit un peu de courant.

Les transistors TR8, TR5 et TR9 conduisent le courant et les trois autres sont fermés. Si nous branchons R14 sur une tension positive, nous fermons la branche droite de l'étage final et le courant moteur change de polarité.

Le simulateur microcap nous permet de calculer l'intensité du courant dans chaque composant et de le lire sur le diagramme de connexion à l'écran. La consommation totale de l'étage final s'élève à env. 300mA pour une tension de batterie de 3 Volts.

Ce circuit possède la particularité étonnante que l'étage final travaille à 3 Volts avec des transistors au silicium. Normalement la tension de seuil d'un transistor au silicium s'élève à 0,7V mais le moteur a été monté entre les collecteurs qui n'enregistrent idéalement qu'une chute de tension de 0,3V. Dans la pratique, il reste pour le moteur toujours un résidu confortable de 1,5V. Les tensions calculées par le simulateur microcap sont documentées dans la figure 9.

Fig. 9: Montage en pont avec le circuit L293D



L'alimentation en 3V est le point de départ idéal pour un robot qui n'est alimenté que par 2 piles. Les transistors PNP ne conviennent cependant mal à une intégration dans un CI tel que le L293D. Un CI présente d'autres avantages tels que la fiabilité, la protection contre des erreurs de commutation, un encombrement réduit et moins de poids. Pour ces raisons, nous avons équipé le robot AAR avec un circuit L293D à double montage en pont qui nous permet de piloter deux moteurs à la fois.

6.2 Montage en pont pour 4,5 Volts

Le circuit L293D (voir fig. 10) arrive à piloter un courant de sortie de 600mA par étage de sortie (valeur crête 1,2A). L'alimentation de l'étage final (VCC2) peut varier entre 4,5V et 36V ce qui nous amène à considérer le CI L293D comme la solution idéale pour piloter des moteurs à courant continu.

La tension d'alimentation minimum (VCC2) s'élève cependant à 4,5V ce qui nous oblige à utiliser au moins 4 piles ou batteries. Cet investissement augmente évidemment le poids du robot mais c'est le prix à payer pour l'emploi du CI moderne.

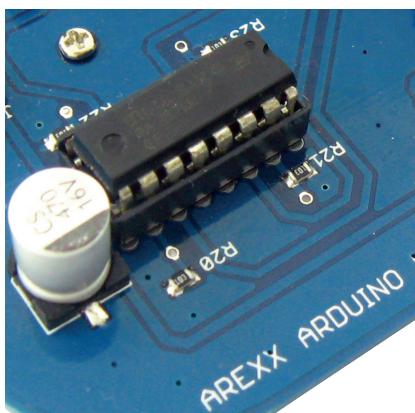


Fig. 10
Montage en pont avec le CI L293D

7. Systèmes odométriques

Ce chapitre décrit quelques concepts intéressants pour l'utilisation du robot AAR. Il s'agit d'idées pour des études et projets artistiques. Que le développement d'un tel logiciel Arduino nous donne des ailes et du génie lors de la programmation.

7.1 Suiveurs de lignes, chercheurs de couleur et fuyards de couleurs

Des capteurs photosensibles permettent de programmer le robot en suiveur de lignes, chercheur de couleur ou fuyard de couleur. Dans le premier cas, le robot suivra fidèlement des lignes continues dans un motif en 8 dans lequel la ligne infinie le fait tourner en bourrique.

Dans le deuxième et troisième cas, le robot évitera par principe la lumière rouge et sera peut-être attiré en même temps par une lumière verte. De tels schémas comportementaux font déjà partie des stratégies pratiques des formes de vie simples.

7.2 Poules mouillées et Mélomanes

Le schéma comportemental qui fait réagir aux bruits environnants est également intéressant. Un robot peureux équipé d'un microphone pourrait éviter une musique présentant de fortes basses tout en montrant une attirance pour les sons aigus d'une flûte. L'amour pour les sons aigus de la flûte pourrait même vaincre la peur des basses lourdes. Ainsi on pourrait obliger le robot de filtrer et de rechercher la source des sons enchanteurs des flûtes à bec en dépit d'une musique rock heavy metal.

Les schémas comportementaux qui réagissent aux basses et aux aigus, à la lumière et aux couleurs, ne nécessitent que très peu de capteurs tels qu'un microphone, deux filtres sonores et quelques capteurs photosensibles équipés de filtres chromatiques.

7.3 Suiveurs de ligne complexes

Les robots qui suivent des lignes ou évitent des motifs de lignes, utilisent généralement une source lumineuse et deux ou plusieurs capteurs photosensibles à l'aide desquels le système de recherche peut identifier une ligne et la suivre. Tout d'abord on peut équiper le robot avec une routine de recherche spécifique dans laquelle le détecteur commence en mode recherche à appliquer une tactique bien particulière p.ex. en tournant selon un schéma hélicoïdal en dessinant des cercles de plus en plus grands jusqu'à ce qu'il découvre un schéma de lignes voyant et commence à suivre la ligne qu'il a trouvé.

L'écriture d'un logiciel qui propose une solution statistiquement satisfaisante pour le processus de recherche de motifs de lignes aléatoires fait déjà partie des tâches de programmation particulièrement difficiles.

7.3.1 Schéma comportemental complexe (comme exercice de programmation)

Le projet peut encore être étendu en organisant la recherche dans un schéma de lignes multicolores où le robot est chassé par des bruits ce qui l'amène à chercher la première ligne rouge qui se présente et qui le conduit dans la sécurité d'un "garage" sombre.

Dès que le fond sonore se calme pendant un certain temps, le robot peut quitter prudemment son "garage" et se mettre à la recherche d'une ligne verte qui l'amène vers un autre "garage" fortement éclairée en vert où le robot se sent "chez lui" même lorsqu'il est exposé à des basses puissantes.

Dès qu'on y mélange les sons aigus d'une flûte, le robot s'agit. Il quitte sa maison verte pour chercher les lignes rouges qui le ramènent vers son sombre cachot.

Il est évident qu'un tel schéma comportemental avec suiveur de lignes, dépendances de couleurs, différentes sources sonores et un schéma complexe de rôle exige une organisation rigoureuse de la structure du logiciel. Seul un concept propre et à structure modulaire permet atteindre cet objectif où le robot se comporte dans toutes les circonstances d'un manière stable et fiable en conservant le comportement spécifié auparavant.

La complexité du logiciel requis forcera l'admiration du programmeur pour les organismes vivants qui combinent un tel comportement avec la recherche régulière de nourriture et une stratégie de reproduction réussie. C'est en effet une performance extraordinaire de la nature de perfectionner et de renouveler chaque jour de tels schémas comportementaux.



8. Programmation du Bootloader

Attention!

Les procédures décrites dans ce chapitre supposent une expérience en programmation !



Il est possible de charger le bootloader Arduino au moyen d'un STK500. Afin de pouvoir transférer les programmes écrits en arduino dans le microcontrôleur Atmega, le processeur Atmega doit être équipé d'un bootloader Arduino spécifique. Le bootloader veille à ce que les signes de code transmis soient écrits au bon endroit de la mémoire Atmega.

Pour appliquer le bootloader, il nous faut les composants suivants:

- * un AVR programmer board (p.ex. le STK500)
- * une alimentation 12V (pour le STK500)
- * un robot AAR avec un connecteur ISP libre sur la platine. (fig. 11)
- * PC avec un port COM physique (de préférence pas de convertisseur USB-RS232 afin d'éviter des risques d'erreurs de timing).



Installez (ou mettez à jour) la version actuelle du logiciel Arduino qui se trouve sur le site internet www.arduino.cc. Le fichier téléchargé sera normalement du type .ZIP ou .RAR sein. Décompressez ces fichiers et placez-les sur votre disque dur.

Utilisez maintenant p.ex. WINAVR pour transférer le bootloader Arduino dans le robot.

Attention!

Le logiciel ARDUINO fait partie de la catégorie freeware et il peut arriver que les différentes versions du logiciel Arduino et les programmes bootloader Arduino ne s'entendent pas très bien!



Fig. 11: Connecteur ISP

En cas de problèmes dans ce domaine, visitez les différents sites internet Arduino ainsi que les forums!

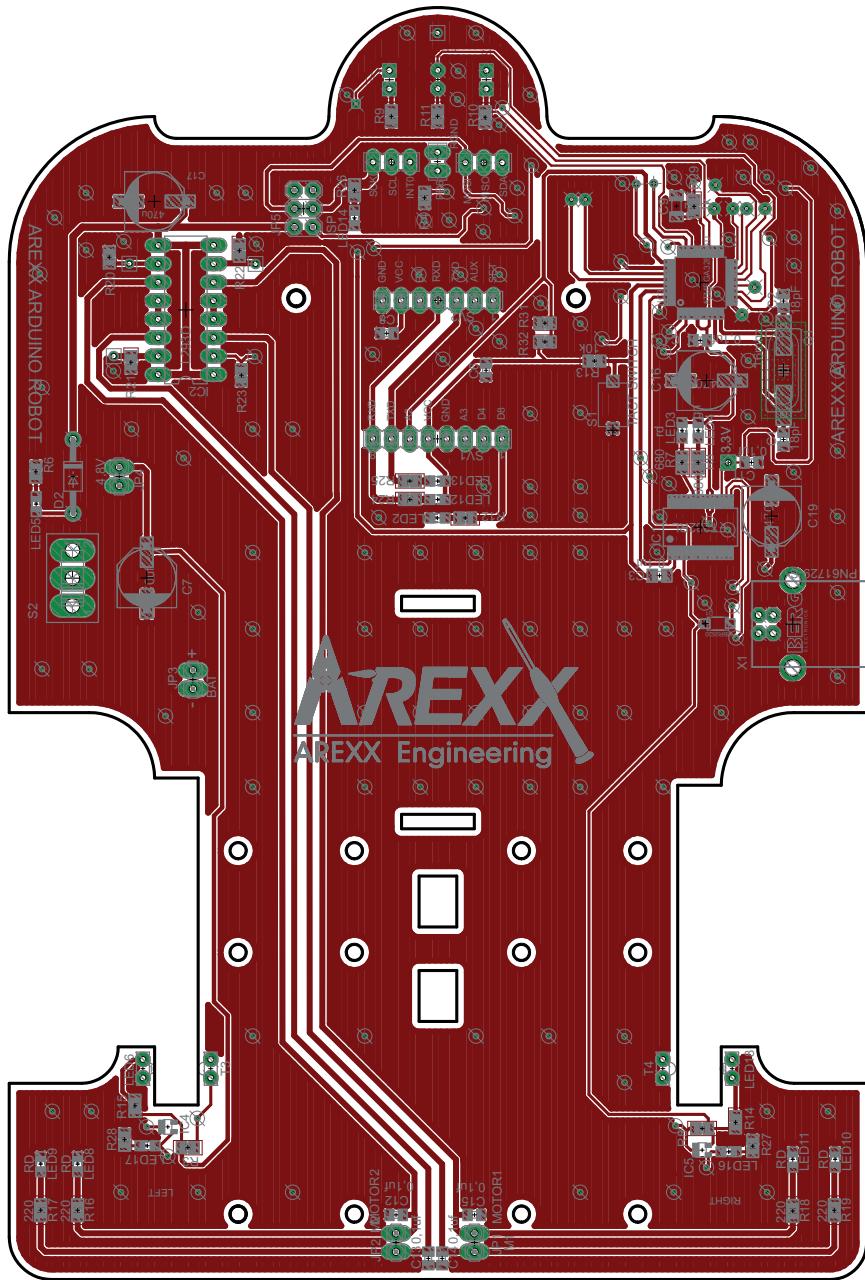
ANNEXE

A. Nomenclature

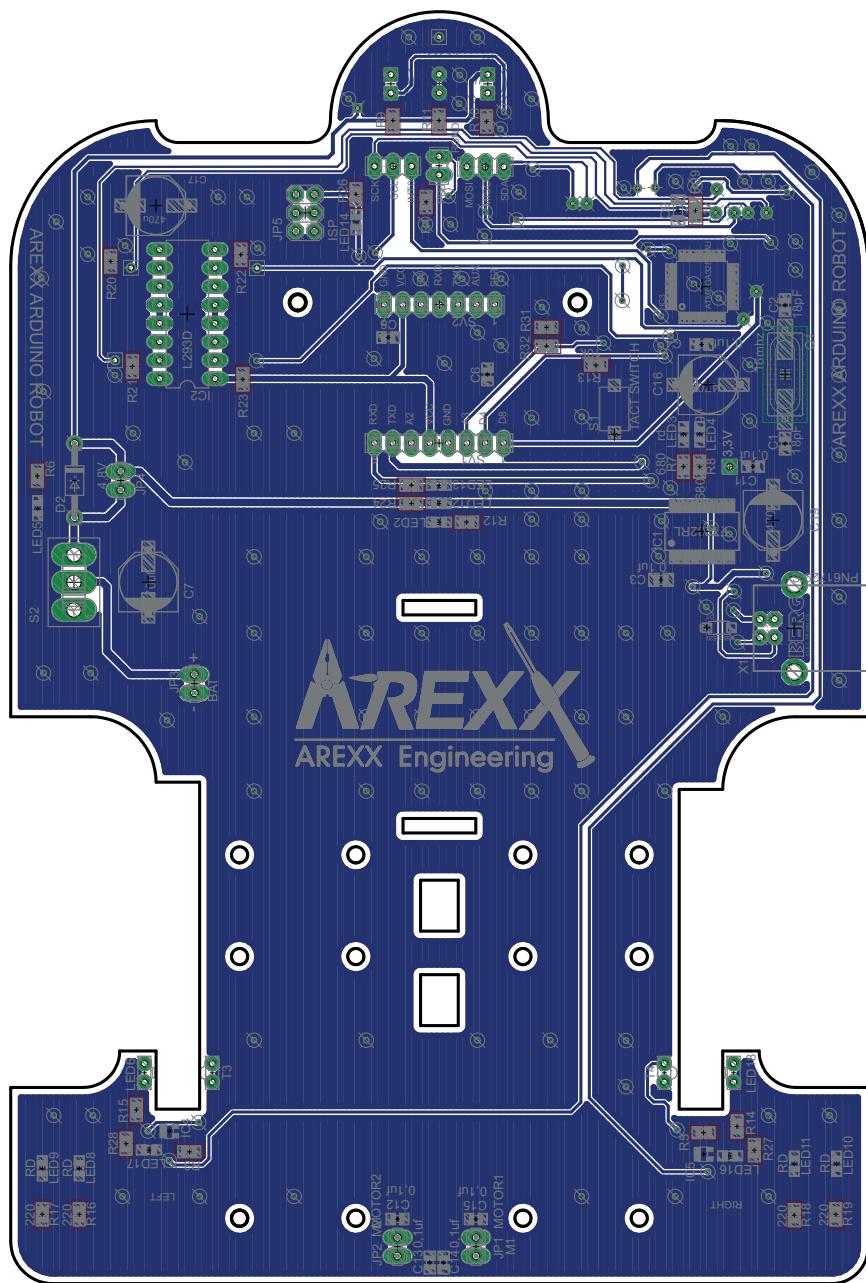
Composant	Valeur	Version
C1	18pF	0805
C2	18pF	0805
C3	0.1uF	C0805K
C4	0,1uF	0805
C6	0,1uF	0805
C7	470uF	CPOL-USF
C8	0,1uF	0805
C9	4,7uF	1206
C11	0,1uF	0805
C12	0,1uF	0805
C13	0,1uF	0805
C14	0,1uF	0805
C15	0,1uF	0805
C16	470uF	CPOL-USF
C17	470uF	CPOL-USF
C19	470uF	CPOL-USF
D1	MBR0520	SOD-123
D2	1N4001	DO41-10
IC1	FT232RL	SSOP28
IC2	L293D	DIL16
IC3	ATMEGA328P-AU	ATMEGA328P-AU
IC4	74AHC1G14DCK	74AHC1G14DCK
IC5	74AHC1G14DCK	74AHC1G14DCK
JP1	M1	1X02
JP2	M2	1X02
JP3	BAT	1X02
JP4	4,8V	1X02
JP5	ISP	2X03
SV2	fem header	FE07-1
T1	SFH300	LED5MM
T2	SFH300	LED5MM
T3	LPT80A	LPT80A
T4	LPT80A	LPT80A
U\$1	3,3V	PIN-T
U\$2	FE03-1	FE03-1
U\$3	FE03-1	FE03-1
U\$4	FE02-1	FE02-1
X1	PN61729-S	PN61729-S
LED1	Rd	LED5MM
LED2	Bl	LEDCHIP-LED0805
LED3	Rd	LEDCHIP-LED0805
LED4	Gn	LEDCHIP-LED0805
LED5	Bl	LEDCHIP-LED0805
LED6	Rd	LEDIRL80A

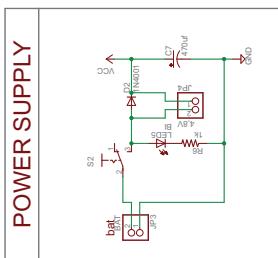
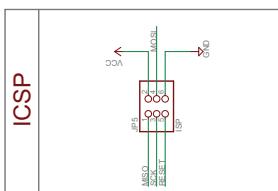
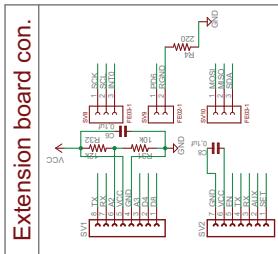
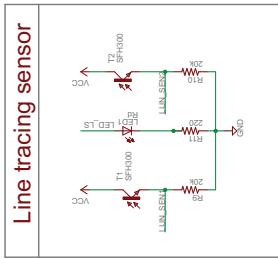
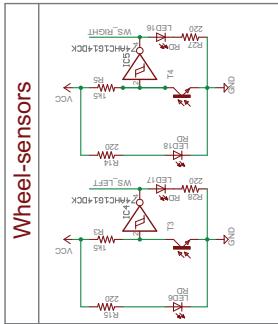
Composant	Valeur	Version
LED8	Rd	LEDCHIP-LED0805
LED9	Rd	LEDCHIP-LED0805
LED10	Rd	LEDCHIP-LED0805
LED11	Rd	LEDCHIP-LED0805
LED12	Gn	LEDCHIP-LED0805
LED13	Rd	LEDCHIP-LED0805
LED14	Bl	LEDCHIP-LED0805
LED16	Rd	LEDCHIP-LED0805
LED17	Rd	LEDCHIP-LED0805
LED18	Rd	LEDIRL80A
Q1	16mhz	CRYSTALHC49UP
R1	20k	R-US_R0805
R2	20k	R-US_R0805
R3	1k5	R-US_R0805
R4	220	R-US_R0805
R5	1k5	R-US_R0805
R6	1k	R-US_R0805
R7	680	R-US_R0805
R8	680	R-US_R0805
R9	20k	R-US_R0805
R10	20k	R-US_R0805
R11	220	R-US_R0805
R12	220	R-US_R0805
R13	10k	R-US_R0805
R14	220	R-US_R0805
R15	220	R-US_R0805
R16	220	R-US_R0805
R17	220	R-US_R0805
R18	220	R-US_R0805
R19	220	R-US_R0805
R20	10k	R-US_R0805
R21	10k	R-US_R0805
R22	10k	R-US_R0805
R23	10k	R-US_R0805
R24	220	R-US_R0805
R25	220	R-US_R0805
R26	220	R-US_R0805
R27	220	R-US_R0805
R28	220	R-US_R0805
R29/C3	0.1uF	R-US_R0805
R31	10k	R-US_R0805
R32	12k	R-US_R0805
S1	TACT SWITCH	TACT_SWITCH
S2	255SB	255SB
SV1	fem header	FE08-1

B. PCB principal face supérieure

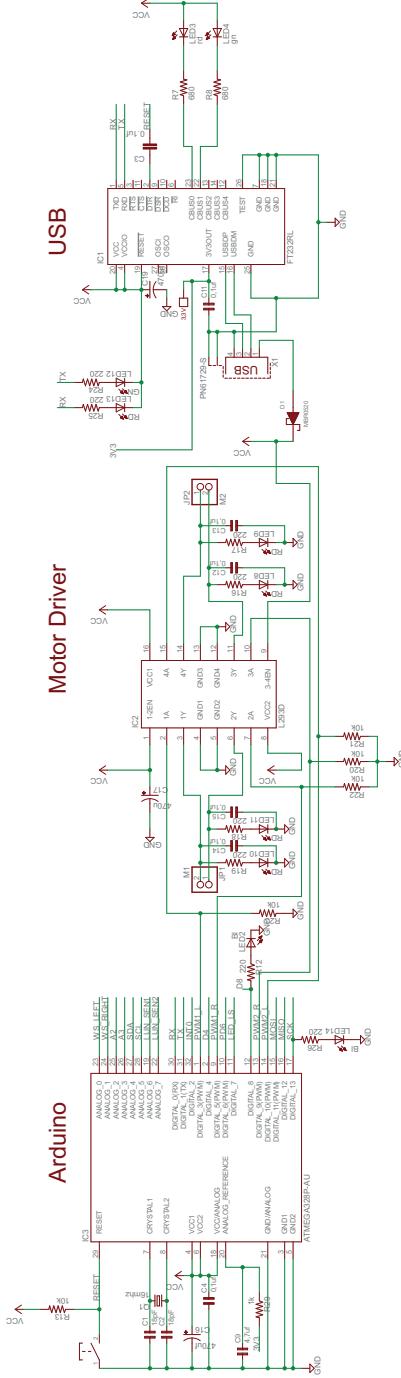


C. PCB principal face inférieure





D. Diagramme de connexion AAR



E. PCB 3D de l'AAR

