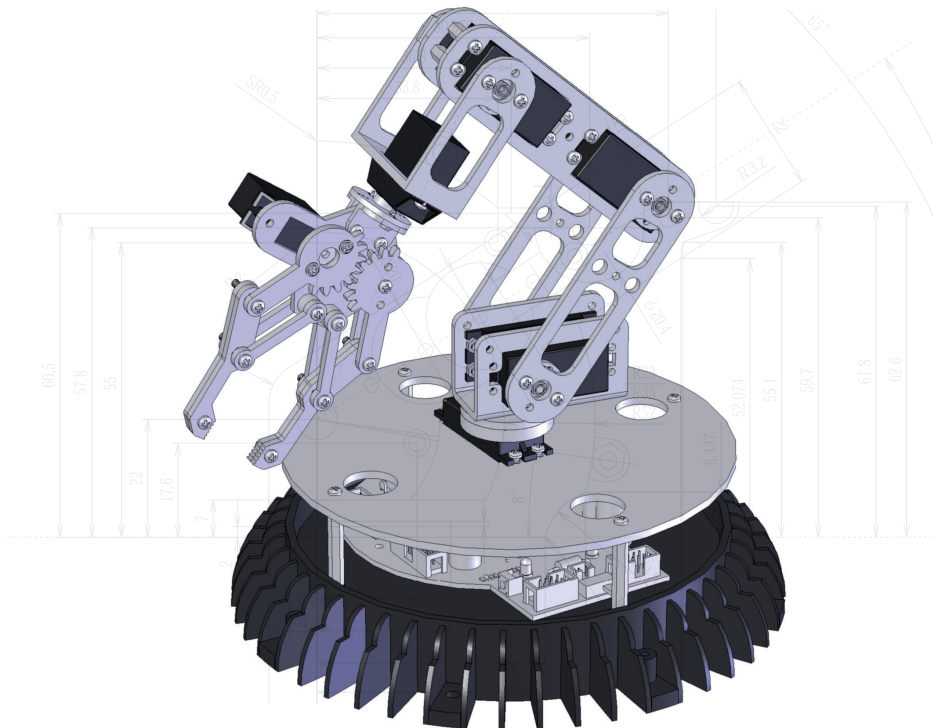




# ROBOT D'APPRENTISSAGE ROBOT ARM PRO

INSTRUCTIONS DE MONTAGE: Modèle RA1-PRO V3



# Table des Matières

1.	Description du produit ROBOT ARM	5
2.	Outils requis	7
3.	Liste des pièces	8
4.	Pieces mécanique	10
5.	Electronique du Robot Arm	14
6.	Installation du logiciel	18
7.	Programmeur et Loader	31
	7.1 Robot loader	32
	7.2 Connexion de l'interface USB sous Windows	29
	7.3 Connexion de l'interface USB sous LINUX	36
	7.4 Test de l'interface USB	37
	7.5 Ouvrir un port sous Linux	38
	7.6 Auto-test	39
	7.7 Calibrage	41
	7.8 Test de clavier	43
8.	RACS	44
9.	Programmation du ROBOT ARM	50

## xx. APPENDICE

A.	Schéma technique Robot Arm	58
B.	Schéma technique Power Supply	59
C.	Schéma technique Connectors	60
D.	Schéma technique Clavier	61
E.	Schéma technique Programmeur adaptateur	62
F.	Platine	63
G.	Schéma technique Adaptateur Programmation	64

**AREXX et ROBOT ARM sont des marques déposées d'AREXX Engineering - PAYS-BAS.**

© Traduction française/French translation (August 2010): AREXX Engineering (NL).

Cette description est protégée par les droits d'auteur. Toute reproduction totale ou partielle est interdite sans l'autorisation écrite de l'importateur européen:

**AREXX Engineering - Zwolle (NL).**

Le fabricant et le distributeur ne sont pas responsables des conséquences d'une fausse manipulation, d'erreurs de montage et/ou d'utilisation de ce produit en cas de non-respect des instructions.

Nous nous réservons de droit de modifier le contenu de ce manuel sans préavis.



**Fabricant:**  
**AREXX Engineering**  
**DAGU HI-TECH**

**Importateur européen**  
**AREXX Engineering**  
**ZWOLLE Die Niederlande**

Support technique lors de la construction du robot:

**WWW.AREXX.COM**  
**WWW.ROBOTERNETZ.DE**

© AREXX Pays-Bas et DAGU Chine

© Traduction française: AREXX - Pays-Bas

<p><b>Mentions légales</b></p> <p>©2013 AREXX Engineering Nervistraat 16 8013 RS Zwolle The Netherlands</p> <p>Tel.: +31 (0) 38 454 2028 Fax.: +31 (0) 38 452 4482</p> <p>E-Mail: Info@arexx.nl</p>	<p>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!</p> <p>Sous réserve de modification des spécifications du produit et des accessoires fournis.</p> <p>Le contenu de ce manuel est susceptible de modifications sans préavis.</p> <p>Vous trouverez des versions plus récentes de ce manuel gratuitement sur <a href="http://www.arexx.com/">http://www.arexx.com/</a></p>
<p>"Robot Arm PRO et Hobby" 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!</p>	
<p><b>Notes concernant les limitations de garantie et de responsabilité</b></p> <p>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.</p> <p>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!</p> <p>Il n'existe aucune garantie que le logiciel fourni satisfasse des attentes individuelles ou fonctionne parfaitement sans interruption ni erreur.</p> <p>Par ailleurs, le logiciel est librement modifiable et chargé par l'utilisateur dans l'appareil. C'est pourquoi l'utilisateur assume l'entière responsabilité quant à la qualité et la performance de l'appareil et de tous les logiciels.</p> <p>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 défectueux ou insuffisants, le client assume tous les frais de service, de réparation et de correction.</p> <p>Nous vous invitons à prendre également connaissance des accords de licence sur le CD-ROM!</p> <p><b>IMPORTANT</b></p> <p>Avant la première utilisation du Robot Arm, 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.</p>	

## 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 l'adaptateur secteur.
- 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 PRODUIT ROBOT ARM

Le ROBOT ARM est un robot à un prix abordable destiné au technicien amateur qui souhaite acquérir les bases de l'électronique, de la mécanique et de la programmation. Le Robot Arm est commandé par un microprocesseur puissant ATMEGA64 qui est programmable en C avec des outils Open Source. L'utilisateur peut télécharger ses propres programmes dans le microprocesseur via l'interface USB fournie et le logiciel de téléchargement Uploader. Les entrées/sorties I/O et le système de bus flexible I2C permettent d'ajouter des modules d'extension au robot afin qu'il puisse réagir à son environnement.

## **Contenu de l'emballage:**

- Robot Arm complet (mécanique et électronique)
- Clavier
- Interface USB avec cordon
- CD-ROM contenant tous les logiciels et manuels requis

## 1.2. Caractéristiques techniques:

- Processeur ATMEGA64
- Différentes entrée/sorties I/O libres
- Bus I2C
- 4 mini-servomoteurs (S05NF)
- 2 maxi-servomoteurs (S06NF)
- Bras en plastique et châssis métallique
- Longueur du bras: 390 mm
- Hauteur: 460 mm
- Diamètre du châssis: 210 mm
- Alimentation: 9-14V / 3-4A

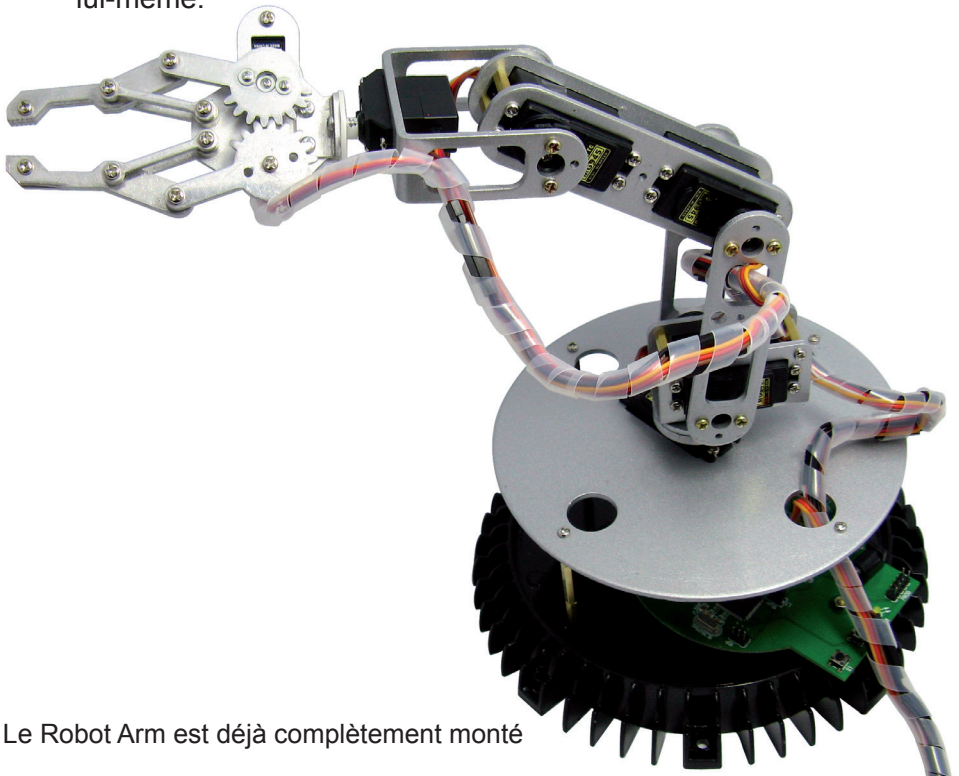


### **Avertissements**

- \* Le droit de retour s'arrête avec l'ouverture du sachet en plastique contenant les composants et pièces.
- \* Lire attentivement le manuel avant la construction.
- \* Manipulez les outils avec précaution.
- \* Ne construisez pas le robot en présence d'enfants en bas âge. Ils pourraient se blesser avec les outils ou avaler des petits composants ou pièces.
- \* Respectez la polarité des piles.
- \* Veillez à ne pas mouiller les piles ni le support de piles. Si le ROBOT ARM est mouillé, retirez les piles et séchez toutes les pièces du mieux possible.
- \* Retirez les piles si le ROBOT ARM ne sera pas utilisé pendant plus d'une semaine.



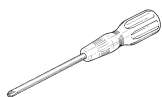
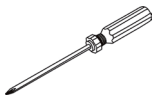
### 1.3. Utilisation du Robot Arm

- Transfert de programmes d'entraînement ou nouveaux dans le Robot Arm.
- Commander le Robot Arm à l'aide d'un clavier
- Commander et programmer le Robot Arm avec le logiciel RACS.
- Commande sans fil via Bluetooth ou un transmetteur APC-220 455MHZ
- Enrichir le Robot Arm avec des modules d'extension prêts à l'emploi afin qu'il puisse entendre, sentir, voir et réagir ainsi à son environnement.
- Tout comme un véritable bras de robot peut p.ex. construire des voitures, celui-ci peut effectuer de petites tâches pour vous.
- L'interface I2C permet au Robot Arm de communiquer avec son environnement et beaucoup d'autres appareils.
- Intelligence artificielle: à l'aide d'un logiciel d'acquisition automatique, le Robot Arm peut améliorer son logiciel lui-même.

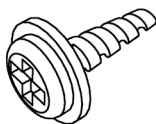


Le Robot Arm est déjà complètement monté

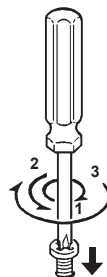
## 2. Outils requis Uniquement requis pour vos propres modifications

Pince plate	Pince coupant de côté	Jeu de tournevis	Tournevis
			
			Livré

### Vis autotaraudeuses (Parker)



Les vis à pas de vis autotaraudeuses se comportent comme des vis à bois, c'est-à-dire elles découpent un pas de vis dans le matériau en effectuant un mouvement circulaire. A cet effet, ces vis ont un pas de vis plus gros et un pointe plus tranchante qu'une vis normale.



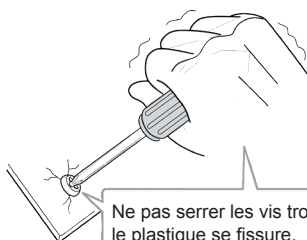
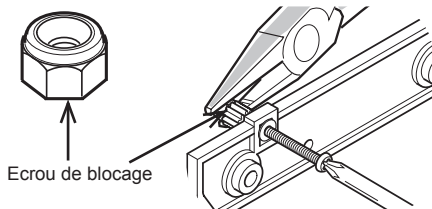
Les vis autotaraudeuses possèdent une encoche à leur pointe qui facilite encore la découpe. La meilleure méthode pour serrer ce type de vis consiste à :

- 1 Entrer la vis
- 2 Retirer légèrement la vis
- 3 Reserrer la vis

*Si les vis sont trop souvent desserrées et reserrées, le trou s'élargit constamment et la vis ne tient plus très bien.*

### Ecrou de blocage

**Fixation d'un écrou de blocage**

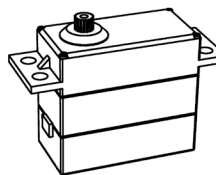


Ne pas serrer les vis trop fort sinon le plastique se fissure.

***Le Robot Arm est déjà complètement monté***

### 3. LISTE DES PIECES Le Robot Arm est déjà complètement monté

Servomoteur



O 2 pcs. S05NF

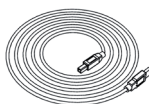
2 pcs S06NF

CD



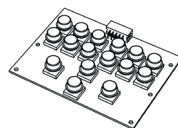
O 1 pc.

Cordon USB



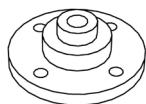
O 1 pc.

Clavier



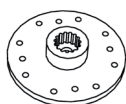
O 1 pc.

Metall-Disque  
avec axe



O 3 pcs.

Servo disque  
Plastik



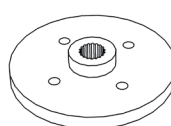
O 1 pc.

Servo disque  
Metall A



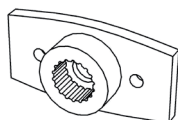
O 1 pc.

Servo disque  
Metall B



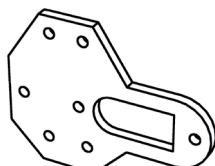
O 1 pc.

Servo-Levier  
Groß



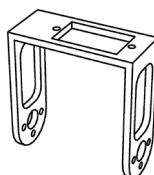
O 3 pcs.

Servo-Support



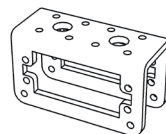
O 1 pc.

Servo-Arbre



O 1 pc.

Montage  
Support



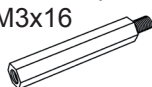
O 1 pc.

Ecrou espaceur  
M3x6



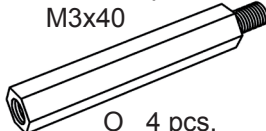
O 4 pcs.

Ecrou espaceur  
M3x16



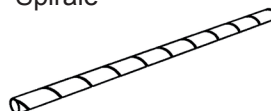
O 2 pcs.

Ecrou espaceur  
M3x40



O 4 pcs.

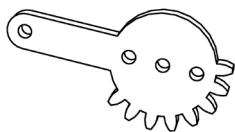
Spirale



O 1 pc.

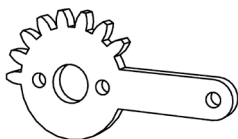


Doigt- A



O 1 pc.

Doigt-Teil B

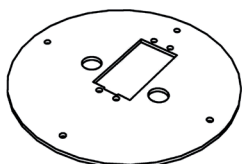


O 1 pc.

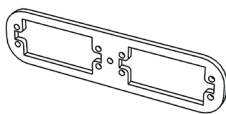
Bout du Doigt



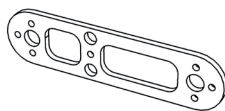
O 4 pcs.

Servo Plaque de  
fond

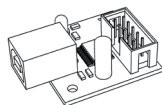
O 1 pc.

Servo-couplage  
M3 - M4

O 2 pcs.

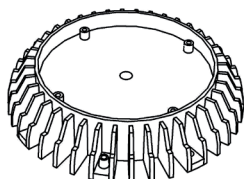
Servo-couplage  
M2 - M3

O 2 pcs.

Adaptateur  
de program-  
mation

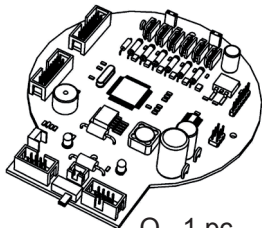
O 1 pc.

Pied du Robot Arm



O 1 pc.

Platine



O 1 pc.

Câble de pro-  
grammation

O 1 St.

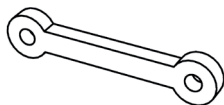
Câble du  
clavier

O 1 pc.

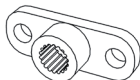
Rallonge pour  
servo

O 1 pc.

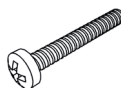
Couplage support



O 5 pcs.

Servos-Levier  
Petit

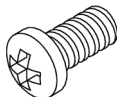
O 1 pcs.

Vis à tête ronde  
M3x20

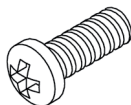
O 9 pcs.

Vis autotaraudeuse  
M3.2x6

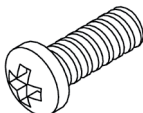
O 4 pcs.

Vis à tête ronde  
M3x6

O 2 pcs.

Vis à tête ronde  
M3x8

O 42 pcs.

Vis à tête ronde  
M3x12

O 9 pcs.

Erou  
M3

O 24 pcs.

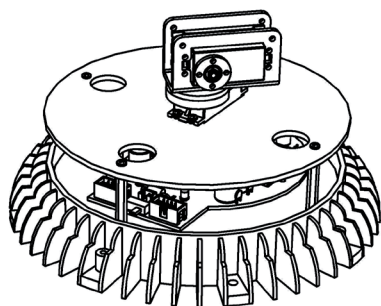
Ecran de  
blocage M2

O 3 pcs.

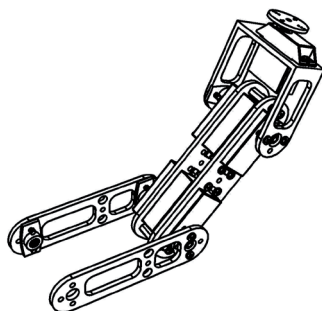
## 4. Pièces mécaniques

Le Robot Arm RA1-PRO est déjà complètement monté

### 4.1. Construction RA1-PRO:

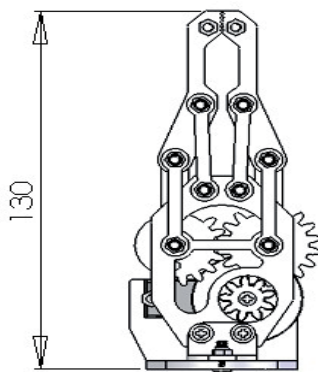
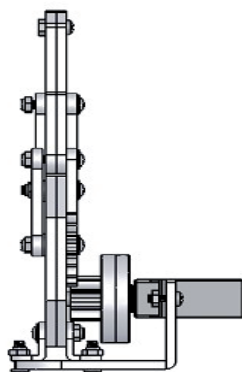


Pied



Bras

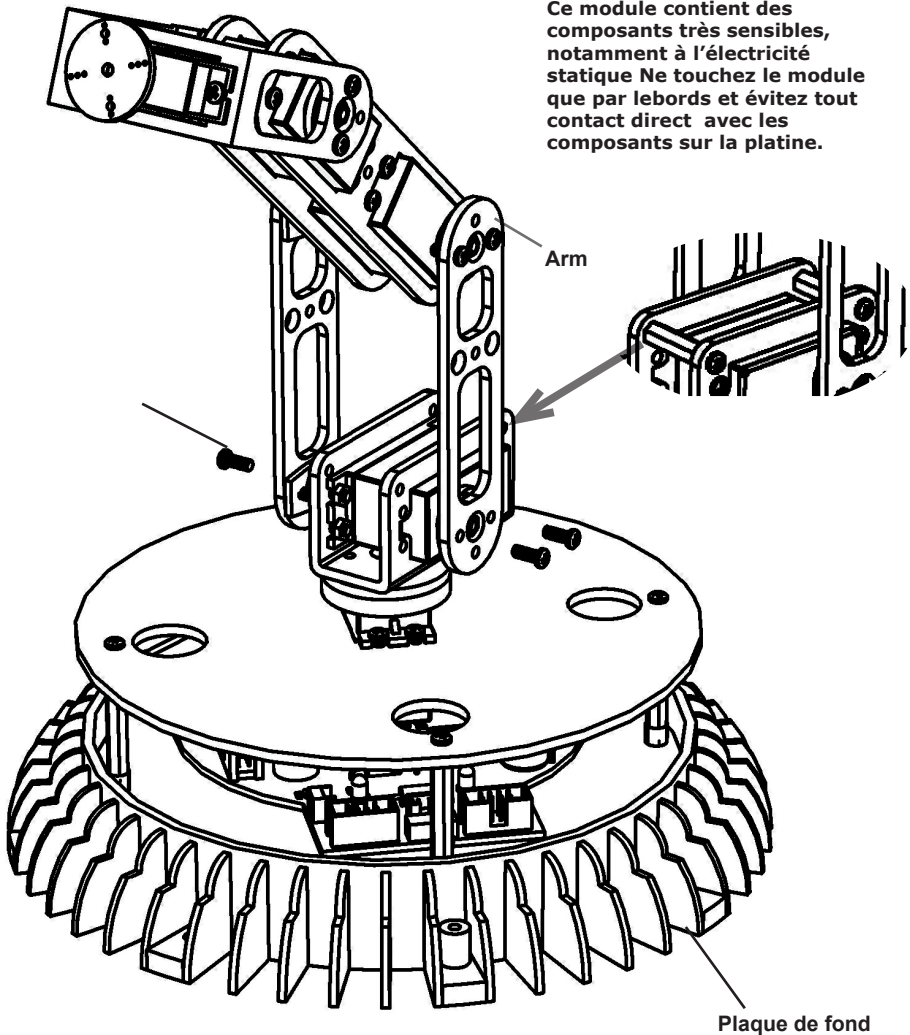
Doigt et disque de transmission



## 4.2. Construction du servo bras:

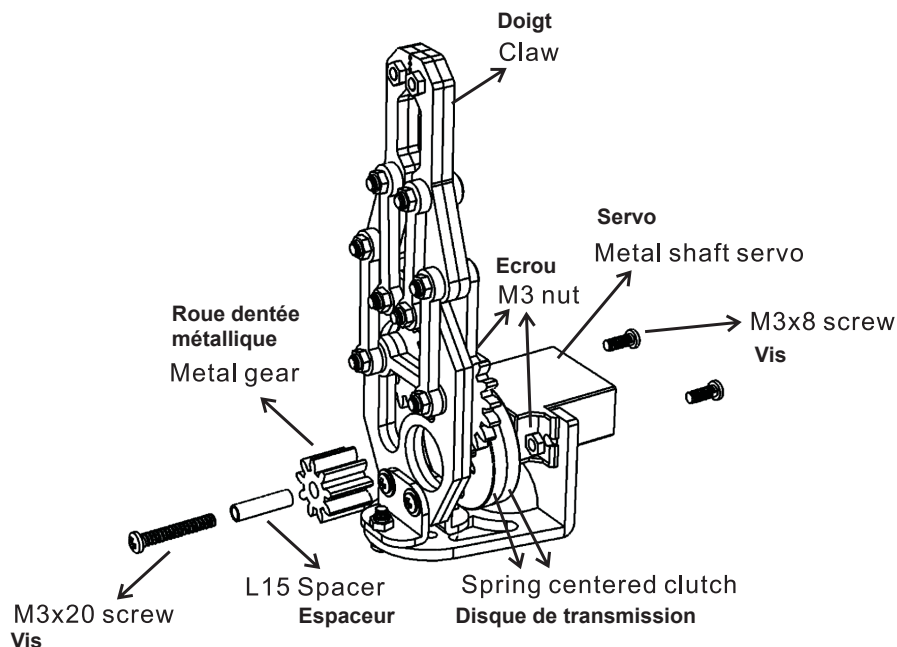


Ce module contient des composants très sensibles, notamment à l'électricité statique. Ne touchez le module que par les bords et évitez tout contact direct avec les composants sur la platine.



**Le Robot Arm RA1-PRO est déjà complètement monté**

### 4.3. Doigt avec disque de transmission

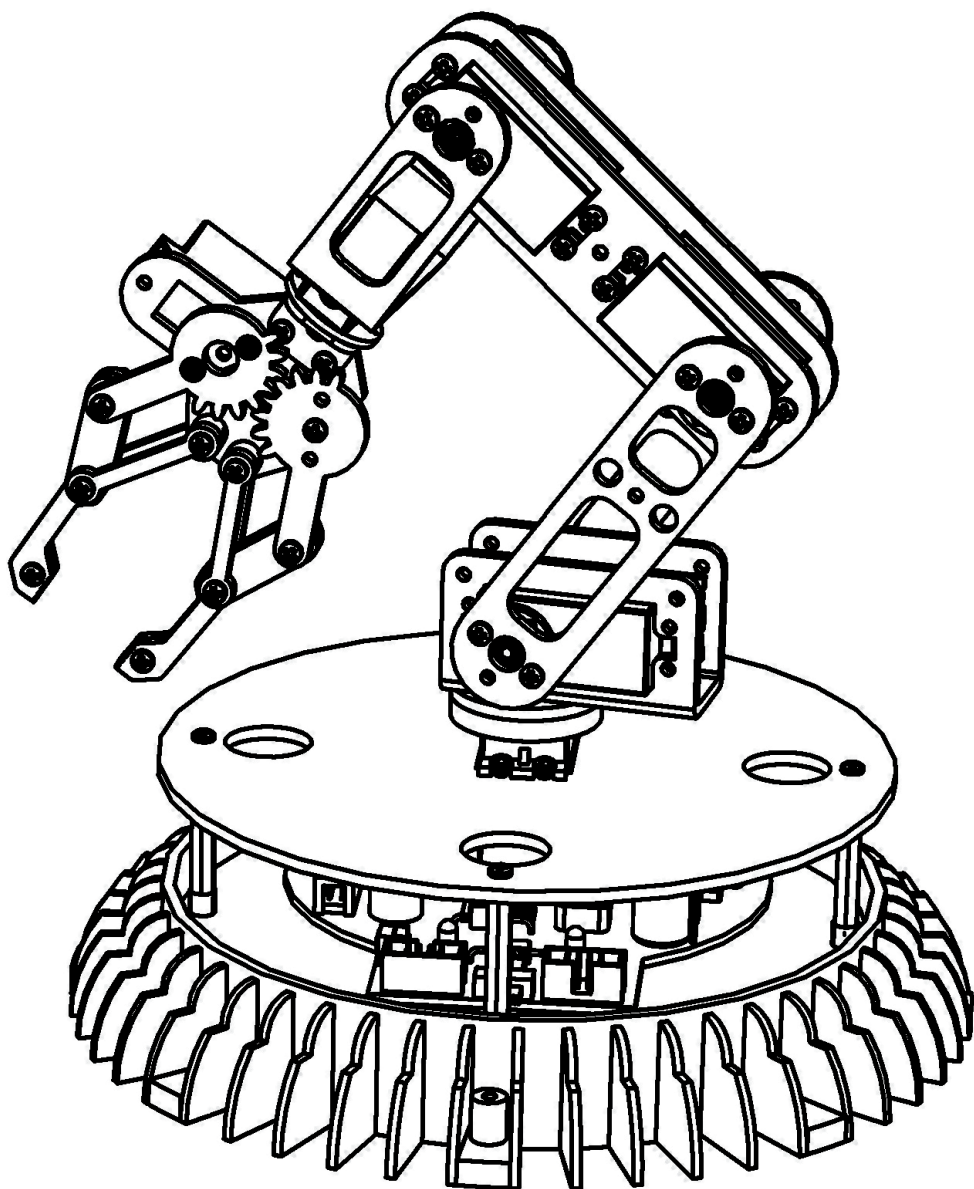


#### ALIGNEMENT ET CONSEILS POUR LE DISQUE DE TRANSMISSION

Si l'embrayage n'est pas suffisamment lubrifié et aligné, il risque d'endommager le servomoteur du doigt !

1. Si le ressort dans le disque ne bouge pas facilement, lubrifiez légèrement la partie intérieure du disque.
2. Si la vis de l'engrenage est trop serrée, elle bloque complètement le disque et le rebord du servomoteur. Augmentez légèrement l'écart ou bien desserrez un peu la vis.
3. Si le disque et l'engrenage ne sont pas correctement alignés l'un sur l'autre, essayez de corriger l'alignement de l'ensemble de façon à ce que le doigt complet soit aligné correctement.

# TERMINE !



# 5. Electronique

## 5.1. Caractéristiques

Ce chapitre donne un aperçu ces capacités du Robot Arm et sert en même temps d'introduction à certaines notions et désignations de composants du robot.

- Micro-contrôleur performant Atmel ATMEGA64 8-Bit
  - ◇ Vitesse 16 MIPS (=16 millions d'instructions par seconde) à une fréquence de 16MHz
  - ◇ Mémoire: Flash ROM 64KB, SRAM 4KB, EEPROM 2KB
  - ◇ Librement programmable en C (avec WinAVR / avr-gcc)!
- Système d'extension flexible, basé sur le bus I<sup>2</sup>C
  - ◇ Ne nécessite que deux lignes de signaux (TWI -> "Two Wire Interface")
  - ◇ Vitesse de transmission allant jusqu'à 400kBit/s
  - ◇ Basé sur le principe maître-esclave
  - ◇ Possibilité de connecter jusqu'à 127 esclaves simultanément
  - ◇ Système de bus très répandu: Il existe un grand nombre de CI, capteurs, etc. standard de plusieurs fabricants qui se branchent pour la plupart directement.
- Possibilité de connecter des modules d'extension sans fil
  - ◇ Modules RP6v2-M256-WIFI pour le contrôle du Robot Arm via un réseau WIFI.
  - ◇ Commande sans fil ARX-APC220 pour RACS 433 MHz
  - ◇ Contrôle sans fil BluetoothARX-BT03 pour ANDROID

***Voir aussi le manuel pour AREXX APC-220, Android et module WIFI***

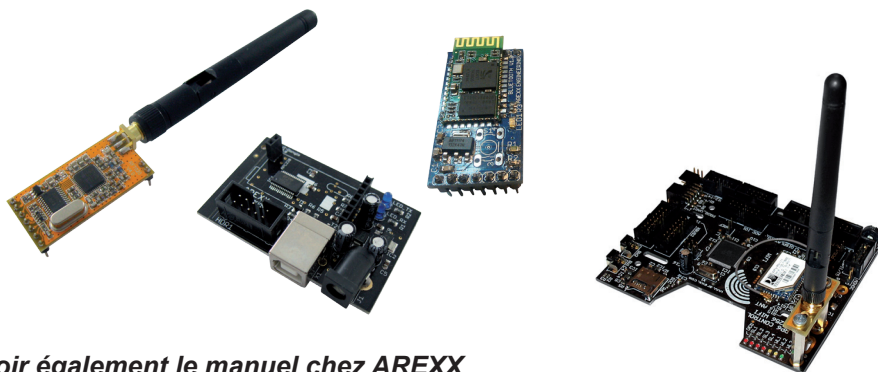
- Interface PC USB pour le téléchargement du programme du PC sur le micro-contrôleur
  - ◇ Liaison filaire permettant une vitesse maximale. Le téléchargement du programme se déroule normalement à 500kBaud. La mémoire complètement vide du micro-contrôleur (30KB, 2KB sont réservés pour le Bootloader) est enregistrée en quelques secondes seulement.
  - ◇ L'interface peut servir à la programmation de tous les modules d'extension équipés d'un micro-contrôleur AVR qui sont disponibles pour le RP6.
  - ◇ Peut servir à la communication avec le robot ou les modules d'extension ce qui facilite considérablement la recherche d'erreurs dans les programmes puisque l'interface permet d'envoyer des valeurs de mesures, des messages de textes et autres données au PC.
  - ◇ Le driver de l'interface génère une interface série virtuelle sous tous les systèmes d'exploitation courants tels que WINDOWS XP/7/8 et Linux.
  - ◇ Afin de faciliter le chargement du programme, le logiciel RobotLoader (qui s'appelait auparavant RP6Loader) est inclus. Il contient également un petit terminal afin de communiquer avec le robot par le biais de messages de texte et tourne sous Windows et Linux.

## 5.2. Changements dans le PCB V3 du type RA1 PRO et RA2-HOBBY

- Nouvel ampli-op (AOP) TI TPS54332, 3500mA
- Nouveau régulateur de tension LDO pour l'alimentation 5V du micro-contrôleur.
- Connecteurs d'alimentation supplémentaires pour l'alimentation du circuit intégré VCC et des servo
- 6 préamplificateurs pour la mesure de courant de tous les servos via des résistances shunt.
- Le bouton reset est maintenant un bouton start/stop à part entière pour démarrer et arrêter l'exécution du programme.
- Autostart (=le programme démarre automatiquement peu de temps après la mise sous tension): est programmable via le RobotLoader si besoin est.
- Connecteurs d'extension supplémentaires avec toutes les broches I/O libres de l'ATMEGA64.
- Connecteur I2C bus compatible avec les modules d'extension RP6 XBUS p.ex. le module WLAN RP6-M256.
- Fiche pour l'APC220 ou le module Bluetooth.
- 4 LED d'état bleues à la place d'une double LED vert/rouge
- Fusible 4A.
- Quartz de 16.000 MHz au lieu de 16.384MHz.
- Des résistances pullup pour le bus I2C se trouvent maintenant sur la platine.
- Un interrupteur M/A sépare l'alimentation de la logique. L'alimentation des servo via l'ampli-up est contrôlée par le micro-contrôleur.
- Tension d'entrée 7 - 14V (max. abs. 18V).

## EXTENSIONS DU ROBOT ARM

- Contrôler à distance via le réseau le module RP6v2-M256-WIFI.
- Télécommande 433MHz ARX-APC220 pour RACS 433 MHz.
- Télécommande ARX-BT03 pour ANDROID.

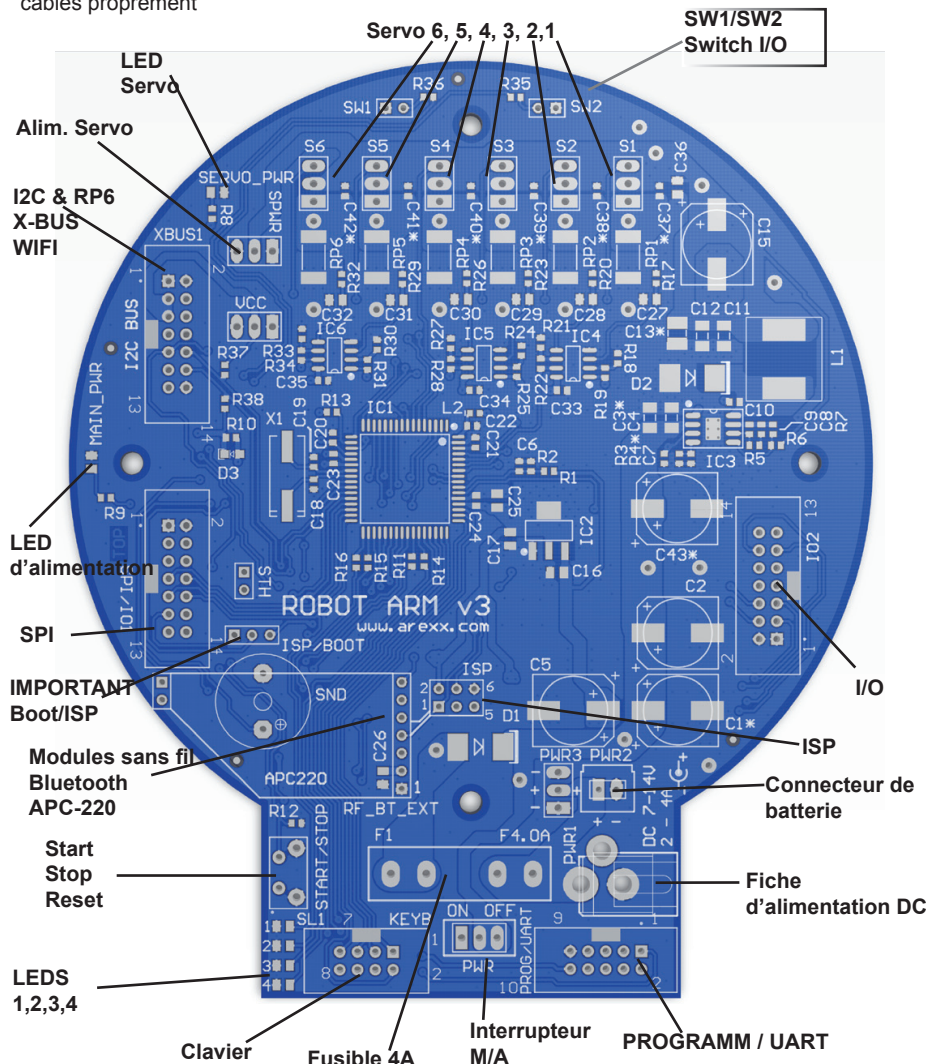


***Voir également le manuel chez AREXX  
APC-220, Android ou module WIFI.***



### 5.3. Implantation de la platine principale

Branchez les servos au moyen de la rallonge (voir p. 51) et utilisez la spirale pour passer les câbles proprement



### 5.4. LEDS

La platine principale du Robot Arm comporte quelques diodes lumineuses. La LED jaune "MAIN\_PWR" indique la présence d'une tension interne et la mise sous tension du robot.

La LED rouge "SERVO\_PWR" ne s'allume que si le convertisseur pour les servo a été activé par le micro-contrôleur. Ceci peut être commandé par le logiciel.

Les 4 LED d'état SL1-4 sont également directement commandées par le micro-contrôleur. Elles peuvent servir d'indicateur d'état dans votre propre programme.



## 5.4. Mise en Route du Robot Arm

1. Vérifiez d'abord les modules mécaniques et électroniques du Robot Arm.
2. Le cas échéant, branchez l'adaptateur secteur (9-14V max.; max.abs. 18V)
3. Mettez le Robot Arm sous tension à l'aide de l'interrupteur général.

### Mise sous tension

#### Alimentation secteur

L'alimentation peut se faire de 2 façons. La solution la plus simple consiste à brancher une alimentation secteur d'une tension de 9-14Vdc/3-4 Ampères sur la fiche DC Power. Ainsi, la tension est branchée sur l'entrée du régulateur de tension.

#### Piles

La deuxième possibilité est le raccordement de piles d'une tension de 9 - 14V

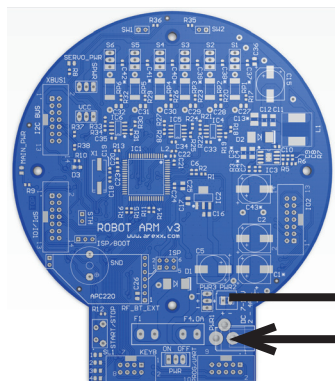
### ATTENTION!

Lorsque la tension est <6,7V, un avertissement apparaît

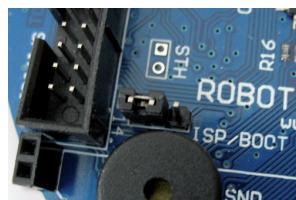


### ATTENTION!

Pour le fonctionnement normal, le jumper "ISP/BOOT" doit se trouver dans la position indiquée sur la photo.



Fiche DC  
9 à 14 Volts



Fiche de batterie  
9 à 14 Volts

Dès que le Robot Arm est branché sur une alimentation, les servos esquissent un léger mouvement et la LED jaune (Power LED) s'allume.

Après ces débuts encourageants, tout porte à croire que le plus gros est fait alors que le vrai travail ne fait que commencer.....!

**Mais.... d'abord nous passons au chapitre 6 dans lequel nous allons installer le logiciel**

## 6. Installation du Logiciel

Nous arrivons maintenant au logiciel qui doit être correctement installé puisque nous en aurons besoin dans tous les chapitres suivants.

Commencez par vous identifier comme administrateur dans votre système puisqu'il faut avoir les droits d'administrateur!

Nous vous conseillons de lire d'abord attentivement le chapitre en entier et d'effectuer ensuite l'installation pas à pas!

Il est impératif de connaître les bases d'un ordinateur équipé d'un système d'exploitation Windows ou Linux ainsi que des programmes courants tels que gestionnaires de fichiers, navigateurs internet, éditeurs de texte, logiciels de compression (WinZip, WinRAR, unzip et autres) et, le cas échéant, Linux-Shell! Si vos connaissances en matière d'informatique sont très limitées, vous devez vous familiariser d'abord avec ce domaine avant d'utiliser le Robot Arm. Ce manuel n'est pas une initiation à l'informatique! Il se limite uniquement au Robot Arm, sa programmation et le logiciel spécifique nécessaire.

### Le CD-ROM du Robot Arm

Le CD-ROM du Robot Arm se trouve probablement déjà dans le lecteur de votre ordinateur - si ce n'est pas encore le cas, insérez-le maintenant. Sous Windows, le menu du CD devrait s'afficher au bout de quelques secondes. Si ce n'est pas le cas, vous pouvez ouvrir le fichier "start.htm" dans le répertoire principal du CD via le gestionnaire de fichiers avec un navigateur internet tel que Firefox. Les fichiers d'installation pour Firefox se trouvent d'ailleurs également sur le CD dans le dossier:

<Lecteur CD>:\Software\Firefox

au cas où vous n'auriez pas encore installé un navigateur Internet à jour (vous devriez avoir au moins la version Firefox 1.x ou Internet Explorer 6...)

Après avoir sélectionné la langue, vous trouverez dans le menu du CD en plus de ce manuel (que vous pouvez également télécharger sur notre page d'accueil), de nombreuses informations, fiches techniques et photos, également l'option du menu "Software" (logiciel). Ici, vous avez accès à tous les outils de logiciel, les pilotes USB et les programmes d'exemples avec le code source pour le Robot Arm.

Selon les réglages de sécurité de votre navigateur Internet, vous pouvez lancer les programmes d'installation directement à partir du CD!

Si votre navigateur ne le permet pas en raison des réglages de sécurité, vous devez d'abord copier les fichiers dans un répertoire sur votre disque dur et les lancer ensuite à partir de là. Vous trouverez plus de détails sur la page de logiciel du menu du CD. Sinon, vous pouvez aussi directement aller sur le lecteur CD à l'aide d'un gestionnaire de fichiers et installer le logiciel à partir du CD. Les noms des répertoires ont été choisis de façon à ce qu'ils puissent être associés clairement aux différents packs de logiciels et systèmes d'exploitation.

## **WinAVR - pour Windows**

Tout d'abord, nous allons installer WinAVR. Comme son nom l'indique, WinAVR n'existe que pour Windows!

### **Les utilisateurs de Linux peuvent sauter ce chapitre.**

WinAVR (qui se prononce comme le mot anglais "whenever") est une collection de nombreux programmes utiles et indispensables pour le développement de logiciels destinés aux microcontrôleurs AVR en langage C. Outre le GCC pour AVR (ce qui s'appelle alors "AVR-GCC", mais nous y reviendrons ultérieurement), WinAVR contient un éditeur de texte source pratique "Programmers Notepad 2", que nous allons utiliser également pour le développement de programme pour le Robot Arm.

WinAVR est un projet privé qui ne dépend d'aucune société. Il est gratuitement disponible sur Internet. Vous trouverez ici de nouvelles versions et d'autres informations:

<http://winavr.sourceforge.net/>

Entretemps, ce projet est officiellement supporté par ATMEL, et l'AVRGCC peut être intégré dans le AVRStudio qui est l'environnement de développement pour les AVR d'ATMEL. Nous n'allons cependant pas l'aborder dans ce manuel car pour nos travaux, le Programmers Notepad convient beaucoup mieux.

Vous trouverez le fichier d'installation de WinAVR dans le classeur sur le CD:

<Lecteur CD>:\Software\AVR-GCC\Windows\WinAVR\

L'installation de WinAVR est très simple et ne nécessite normalement aucun réglage. Donc, cliquez tout simplement toujours sur "Suivant"!

*Si vous travaillez avec Windows Vista ou Windows 7, vous devez impérativement utiliser la dernière version de WinAVR! Cela devrait également fonctionner sans problème avec Windows 2000 et XP. Si ce n'est pas le cas, vous pouvez essayer une des deux versions plus anciennes qui se trouvent également sur le CD (avant toute nouvelle installation, désinstallez systématiquement les versions WinAVR plus anciennes qui sont déjà installées!). Officiellement Win x64 n'est pas encore supporté mais le CD contient un patch pour les systèmes Win x64 en cas de problème. Vous trouverez plus d'informations à ce sujet sur la page de logiciel du menu CD!*

## **AVR-GCC, avr-libc et avr-binutils - pour Linux**

*(Les utilisateurs de Windows peuvent sauter ce paragraphe!)*

C'est un peu plus difficile sous Linux. Certes, les paquets nécessaires existent déjà pour certaines distributions mais il s'agit souvent de versions obsolètes. C'est pourquoi vous devez compiler et installer des versions plus récentes. Il est impossible de rentrer dans le détail des nombreuses distributions Linux telles que SuSE, Ubuntu, RedHat/Fedora, Debian, Gentoo, Slackware, Mandriva etc. en x versions différentes avec leurs particularités individuelles et nous nous tiendrons ici aux généralités.

Ceci est valable pour tous les autres paragraphes traitant de Linux dans ce chapitre!

Le procédé décrit ici ne sera donc pas obligatoirement couronné de succès pour vous. Il est souvent utile de chercher sur Internet p.ex. "<LinuxDistribution> avr gcc" et similaire (essayez des orthographes différents!). C'est également valable pour tous les autres paragraphes sur Linux - évidemment avec des mots-clé adaptés. Si vous rencontrez des problèmes lors de l'installation de l'AVR-GCC, vous pouvez également regarder sur notre site ou dans un des nombreux forums de robotique ou de Linux. Commencez par désinstaller des versions déjà installées de l'avr-gcc, des avr-binutils et de l'avr-libc car elles sont généralement déjà obsolètes. Vous pouvez le faire à l'aide du gestionnaire de paquets de votre distribution en effectuant une recherche sur la mise en service d'„avr“ et en désinstallant les trois paquets nommés ci-dessus - dans la mesure où ils existent! Une ligne de commande telle que

```
> which avr-gcc
```

vous permet de savoir facilement si l'avr-gcc est déjà installé ou non, et si oui où il se trouve.

Si un chemin s'affiche, une version est déjà installée. Dans ce cas, saisissez simplement:

```
> avr-gcc --version
```

et regardez l'édition. Si le numéro de la version est inférieur à 3.4.6, vous devez absolument désinstaller cette ancienne version.

Si le numéro de la version se situe entre 3.4.6 et 4.1.0, vous pouvez essayer dans un premier temps de compiler les programmes (voir chapitre suivant). En cas d'échec, vous pouvez alors installer les nouveaux outils. Nous installons dans ce manuel la version actuelle 4.1.1 (mise à jour de mars 2007) avec quelques patches importants.

Si les paquets ci-dessus ne sont pas indiqués dans le gestionnaire de paquets bien qu'il soit certain qu'un avr-gcc existe déjà, vous devez effacer manuellement les fichiers binaires correspondants. Cela veut dire effectuer une recherche dans les répertoires /bin, /usr/bin, etc. pour tous les fichiers qui commencent par "avr" et les effacer (évidemment UNIQUEMENT ces fichiers et rien d'autre!). Des répertoires éventuellement existants comme /usr/avr ou /usr/local/avr doivent également être effacés.

Attention: Vous devez impérativement vous assurer que les outils de développement Linux normaux comme GCC, make, binutils, libs, etc. sont installés avant de pouvoir commencer avec la compilation et l'installation! Le mieux est de le faire à l'aide du gestionnaire de paquets de votre distribution. Chaque distribution Linux devrait avoir les paquets nécessaires sur le CD d'installation fourni ou bien mettre à disposition sur Internet des paquets actualisés.

Assurez-vous que le programme "textinfo" est installé. Le cas échéant, installez le paquet correspondant avant de continuer, sinon cela ne fonctionnera pas!

Lorsque ce sera fait, vous pouvez commencer avec l'installation proprement dite.

Il n'y a qu'une alternative: soit vous faites tout manuellement, soit vous utilisez un script d'installation très facile à utiliser.

Nous vous conseillons d'essayer dans un premier temps le script. Si cela ne fonctionne pas, vous pouvez toujours installer le compilateur manuellement!

**Attention:**

Vous devez disposer de suffisamment d'espace sur le disque dur pour l'installation! Temporairement, il faut avoir plus de 400MB de libre dont plus de 300MB pourront être effacés après l'installation. Cependant, il vous faut l'espace total pendant la transcription.

Un grand nombre des étapes d'installation suivantes requièrent des droits **ROOT**, donc connectez-vous de préférence avec "su" en tant que root ou bien exécutez des commandes critiques avec "sudo" ou équivalent comme on doit le faire p.ex. sous Ubuntu (le script d'installation, mkdir dans les répertoires /usr/local et make install nécessite des droits ROOT).

**Respectez l'orthographe EXACTE de toutes les commandes qui suivent!**

Chaque caractère a son importance même si certaines commandes semblent un peu étranges - tout est correct et il ne s'agit pas d'une erreur de frappe!

Il faut évidemment tout de même remplacer( <Lecteur CD-ROM>) par le chemin du lecteur CD-ROM!

Dans le dossier:

<Lecteur CD-ROM>:\Software\avr-gcc\Linux

sur le CD, vous trouverez tous les fichiers d'installation pour l'avr-gcc, avr-libc et binutils qui nous intéressent.

D'abord vous devez copier tous les fichiers d'installation dans un répertoire sur votre disque dur - **cela s'applique aux deux variantes d'installation!**

Nous utiliserons ici le répertoire Home (l'abréviation usuelle pour le répertoire Home actif est un tilde: „~“):

```
> mkdir ~/Robot Arm
> cd <Lecteur CD-ROM>/Software/avr-gcc/Linux
> cp * ~/Robot Arm
```

Lorsque l'installation a réussi, vous pouvez effacer ces fichiers pour libérer de l'espace!

## Script d'installation automatique

Si on rend le script exécutable avec `chmod`, on peut commencer tout de suite:

```
> cd ~/Robot Arm
> chmod -x avrgcc_build_and_install.sh
> ./avrgcc_build_and_install.sh
```

Répondez “y” à la question si vous voulez faire l'installation avec cette configuration ou non.

ATTENTION: La traduction et l'installation peuvent prendre quelque temps en fonction de la puissance de votre système (p.ex. env. 15 minutes sur un notebook 2GHz CoreDuo - et un peu plus sur des systèmes un peu plus lents).

Le script copie également quelques patches - ce sont tous les fichiers `.diff` qui se trouvent dans le répertoire.

Si tout se passe bien, voici ce qui devrait s'afficher tout à la fin:

```
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) installation of avr GNU tools complete
(./avrgcc_build_and_install.sh) add /usr/local/avr/bin to your path to use the avr GNU tools
(./avrgcc_build_and_install.sh) you might want to run the following to save disk space:
(./avrgcc_build_and_install.sh)
(./avrgcc_build_and_install.sh) rm -rf /usr/local/avr/source /usr/local/avr/build
```

Alors, vous pourrez accepter la proposition et exécuter

```
rm -rf /usr/local/avr/source /usr/local/avr/build
```

Cela efface tous les fichiers temporaires dont vous n'aurez normalement plus besoin.

Vous pouvez sauter le prochain paragraphe et mettre le chemin sur les outils `avr`.

Si l'exécution du script échoue, regardez attentivement les messages d'erreurs (monter aussi l'écran) - généralement il manque des programmes qu'il aurait fallu installer auparavant (comme p.ex. le texinfo mentionné ci-dessus). Avant de continuer après une erreur, il est prudent d'effacer les fichiers déjà générés dans le répertoire d'installation standard „`/usr/local/avr`” – il est même préférable d'effacer tout le répertoire.

Si vous ne savez pas exactement ce qui s'est mal passé, enregistrez toutes les éditions de commande dans un fichier et adressez-le au support. Il est important d'envoyer toujours un maximum d'informations! Il sera ainsi plus facile de vous aider.

## **GCC pour l'AVR**

Le GCC est patché, traduit et installé à peu près comme les Binutils:

```
> cd ~/Robot Arm> bunzip2 -c gcc-4.1.1.tar.bz2 | tar xf -
> cd gcc-4.1.1
> patch -p0 < ../gcc-patch-0b-constants.diff
> patch -p0 < ../gcc-patch-attribute_alias.diff
> patch -p0 < ../gcc-patch-bug25672.diff
> patch -p0 < ../gcc-patch-dwarf.diff
> patch -p0 < ../gcc-patch-libiberty-Makefile.in.diff
> patch -p0 < ../gcc-patch-newdevices.diff
> patch -p0 < ../gcc-patch-zz-atmega256x.diff
> mkdir obj-avr
> cd obj-avr
> ../configure --prefix=$PREFIX --target=avr --enable-languages=c,c++ \
--disable-nls --disable-libssp --with-dwarf2
> make
> make install
```

Après le \ appuyez tout simplement sur Entrée et continuez à écrire. Ainsi la commande peut être répartie sur plusieurs lignes mais on peut également le laisser complètement de côté.

## **AVR Libc**

Enfin, il reste encore l'AVR libc:

```
> cd ~/Robot Arm
> bunzip2 -c avr-libc-1.4.5.tar.bz2 | tar xf -
> cd avr-libc-1.4.5
> ./configure --prefix=$PREFIX --build=`./config.guess` --host=avr
> make
> make install
```



Attention: dans la commande: `--build= `./config.guess`` veuillez à bien mettre l'accent grave (touche alt gr + 7 suivi d'un espace). Ne pas confondre avec l'apostrophe ou les guillemets car cela ne fonctionnera pas.

## Inscrire le chemin

Vous devez veiller maintenant à ce que le répertoire `/usr/local/avr/bin` soit inscrite dans la variable du chemin sinon il sera impossible d'appeler l'`avr-gcc` de l'écran ou des makefiles. A cet effet, vous devez inscrire le chemin dans le fichier `/etc/profile` ou bien `/etc/environment` ou autre (change d'une distribution à l'autre) en le séparant d'un double point `“:”` des autres inscriptions déjà existantes. Le fichier pourrait donc se présenter de la manière suivante:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/local/avr/bin"
```

*Entrer maintenant dans un écran quelconque “avr-gcc --version” comme décrit ci-dessus. Si cela fonctionne, l'installation a réussi!*

## Installation manuelle

Si vous préférez installer le compilateur manuellement ou si l'installation avec le script ne fonctionne pas, vous pouvez suivre les instructions dans le paragraphe suivant.

La description qui suit se base sur l'article suivant:

[http://www.nongnu.org/avr-libc/user-manual/install\\_tools.html](http://www.nongnu.org/avr-libc/user-manual/install_tools.html)

qui se trouve aussi dans la documentation AVR Libc au format PDF sur le CD:

*<Lecteur CD-ROM>:\Software\Documentation\avr-libc-user-manual-1.4.5.pdf*

Nous abrégeons un peu ici mais nous utilisons quelques patches importants sans lesquels certaines choses ne fonctionnent pas.

Nous devons d'abord créer un répertoire dans lequel nous installons tous les outils. Ce répertoire devrait être: `/usr/local/avr`.

Donc saisissez dans l'interface en ligne de commande EN TANT QUE ROOT:

```
> mkdir /usr/local/avr  
> mkdir /usr/local/avr/bin
```

Il ne doit pas s'agir obligatoirement de ce répertoire. Nous créons tout simplement la variable `$PREFIX` pour ce répertoire:

```
> PREFIX=/usr/local/avr  
> export PREFIX
```

Voici ce qu'il faut impérativement ajouter à la variable `PATH`:

```
> PATH=$PATH:$PREFIX/bin  
> export PATH
```

## Binutils pour AVR

Vous devez décompresser maintenant le code source des binutils et faire intervenir quelques patches.

Nous supposons ici que vous avez tout copié dans le répertoire Home ~/Robot Arm:

```
> cd ~/Robot Arm
> bunzip2 -c binutils-2.17.tar.bz2 | tar xf -
> cd binutils-2.17
> patch -p0 < ../binutils-patch-aa.diff
> patch -p0 < ../binutils-patch-atmega256x.diff
> patch -p0 < ../binutils-patch-coff-avr.diff
> patch -p0 < ../binutils-patch-newdevices.diff
> patch -p0 < ../binutils-patch-avr-size.diff
> mkdir obj-avr
> cd obj-avr
```

Maintenant nous exécutons le script configure:

```
> ../configure --prefix=$PREFIX --target=avr --disable-nls
```

Ce script établit ce qui existe dans votre système et génère des fichiers makefiles en conséquence. Maintenant vous pouvez traduire et installer les binutils.

```
> make
> make install
```

Selon la capacité de votre système, cela peut prendre quelques minutes. Cela s'applique aussi aux deux autres paragraphes, surtout pour le GCC.

## Java 6

Le RobotLoader (voir ci-dessous pour d'autres informations) a été développé pour la plateforme Java et fonctionne avec Windows et Linux (en théorie aussi avec d'autres systèmes d'exploitation comme OS X mais malheureusement AREXX Engineering ne pourra pas encore vous apporter une assistance officielle). Pour que cela fonctionne, il est nécessaire d'installer une version actualisée de Java Runtime Environment (JRE). Vous l'avez souvent déjà sur l'ordinateur mais il doit s'agir au moins de la version 1.6 (= Java 6)! Si aucun JRE ni JDK n'est encore installé, vous devez d'abord installer le JRE 1.6 fourni sur le CD de la société SUN Microsystems ou bien télécharger une version plus récente à partir de <http://www.java.com> ou <http://java.sun.com>.

## **Windows**

JRE 1.6 pour Windows se trouve dans le dossier suivant:

<Lecteur CD-ROM>:\Software\Java\JRE6\Windows\

Sous Windows, l'installation de Java est très simple: Démarrez le setup et suivez les instructions à l'écran. C'est tout. Vous pouvez sauter le paragraphe suivant.

## **Linux**

En général, l'installation sous Linux se déroule sans problème mais certaines distributions exigent un peu de travail manuel.

Dans le dossier:

<Lecteur CD-ROM>:\Software\Java\JRE6\

vous trouverez le JRE1.6 comme RPM (SuSE, RedHat etc.) et comme archive à auto-extraction „.bin“. Sous Linux, il est recommandé de chercher d'abord les paquets Java (mots-clé de recherche p.ex. „java“, „sun“, „jre“, „java6“ ...) dans le gestionnaire de paquets de votre distribution et d'utiliser ces paquets propres à votre distribution au lieu de prendre ceux du CD-ROM. Veuillez cependant à installer absolument Java 6 (=1.6) ou une version plus récente mais en aucun cas une version plus ancienne!

Sous Ubuntu ou Debian, l'archive RPM ne fonctionne pas directement. Vous devez faire appel aux gestionnaires de paquet de votre distribution pour trouver un paquet d'installation approprié. Cependant le RPM devrait fonctionner correctement avec beaucoup d'autres distributions telles que RedHat/Fedora et SuSE. Si ce n'est pas le cas, il vous reste encore la possibilité de décompresser JRE de l'archive à autoextraction (.bin) (p.ex. dans /usr/lib/Java6) et d'indiquer manuellement les chemin vers JRE (PATH et JAVA\_HOME etc.).

Suivez toujours les instructions d'installation de Sun qui se trouvent également dans le répertoire ci-dessus et sur le site Internet de Java (voir ci-dessus).

Dans l'interface en ligne de commande, vous pouvez vérifier si Java a été installée correctement en exécutant la commande „java -version“ . Voici ce qui devrait à peu près s'afficher:

```
java version "1.6.0"  
Java(TM) SE Runtime Environment (build 1.6.0-b105)  
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

Si le message est complètement différent, vous avez soit installé la mauvaise version, soit une autre version Java est également installée sur votre système.

## **Robot Loader**

Le Robot Loader a été développé pour pouvoir charger facilement de nouveaux programmes dans le Robot Arm ainsi que tous les modules d'extension (dans la mesure où ils disposent d'un microcontrôleur avec un bootloader compatible). Par ailleurs, il comporte quelques fonctions supplémentaires utiles telles qu'un programme de terminal simple.

Il n'est pas nécessaire d'installer le Robot Loader lui-même. Il suffit de copier le programme quelque part dans un nouveau dossier sur le disque dur. Le Robot loader se trouve dans l'archive zip sur le CD-ROM:

```
<Lecteur CD-ROM>:\Software\RobotLoader\RobotLoader.zip
```

Vous devez maintenant le dézipper quelquepart sur le disque dur, p.ex. dans un nouveau dossier C:\Programme\RobotLoader (ou similaire). Dans ce dossier, vous trouverez le fichier RobotLoader.exe que vous pouvez démarrer avec un double-clic.

Le véritable programme Robot Loader se trouve dans l'archive Java (JAR) RobotLoader\_lib.jar. Vous pouvez aussi le démarrer à partir de la ligne de commande.

## **Sous Windows:**

```
java -Djava.library.path=".lib" -jar RobotLoader_lib.jar
```

## **Linux:**

```
java -Djava.library.path="./lib" -jar RobotLoader_lib.jar
```

Cette longue option D est nécessaire pour que JVM puisse trouver toutes les bibliothèques utilisées. Ce n'est pas nécessaire sous Windows où il suffit de démarrer tout simplement à l'aide du fichier .exe. Pour Linux, il existe un script shell „RobotLoader.sh“. Il sera éventuellement nécessaire de rendre le script exécutable (chmod -x ./RobotLoader.sh). Ensuite on peut le démarrer dans une interface en ligne de commande avec „./RobotLoader.sh“.

Il est conseillé de créer un raccourci sur le bureau ou dans le menu Démarrer pour pouvoir démarrer facilement le Robot Loader. Sous Windows, c'est très simple: Vous faites un clic droit sur le fichier RobotLoader.exe et dans le menu "Envoyer à" vous cliquez sur "Bureau (Créer un raccourci)".

### **Robot Arm Library, Robot Arm CONTROL Library et programmes d'exemple**

La Robot Arm Library et les programmes d'exemples afférents se trouvent dans une archive zip sur le CD:

<Lecteur CD-ROM>:\Software\Robot Arm Examples\Robot ArmExamples **[PRO]**.zip

Vous pouvez les décompresser directement dans un répertoire de votre choix sur le disque dur. Il est recommandé de décompresser les fichiers d'exemples dans un dossier sur un fragment de données. Ou bien dans le dossier "Mes Documents" dans un sous-dossier "Robot Arm\Examples\" ou bien sous Linux dans le répertoire Home. C'est à vous de voir.

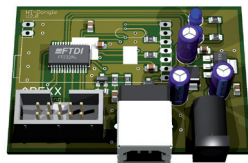
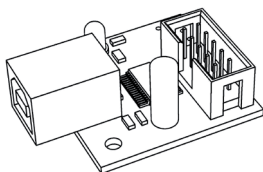
Nous aborderons les différents programmes d'exemple ultérieurement dans le chapitre du logiciel.

## 7. Programmer et Loader

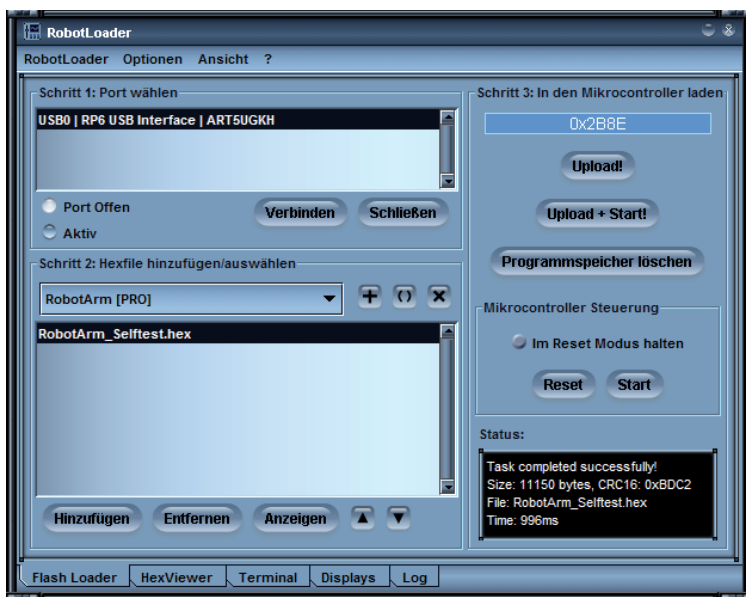
Pour charger un programme HEX du PC dans notre Robot Arm, nous allons utiliser l'adaptateur de programmation USB et notre logiciel RobotLoader.

Le récepteur/émetteur (transmetteur) USB fourni doit être connecté d'un côté sur un port USB de l'ordinateur et de l'autre côté au port Prog/UART de la platine du Robot Arm.

Le chargement d'un programme dans le Robot Arm efface automatiquement le programme précédent déjà présent.



**Adaptateur de programmation USB**



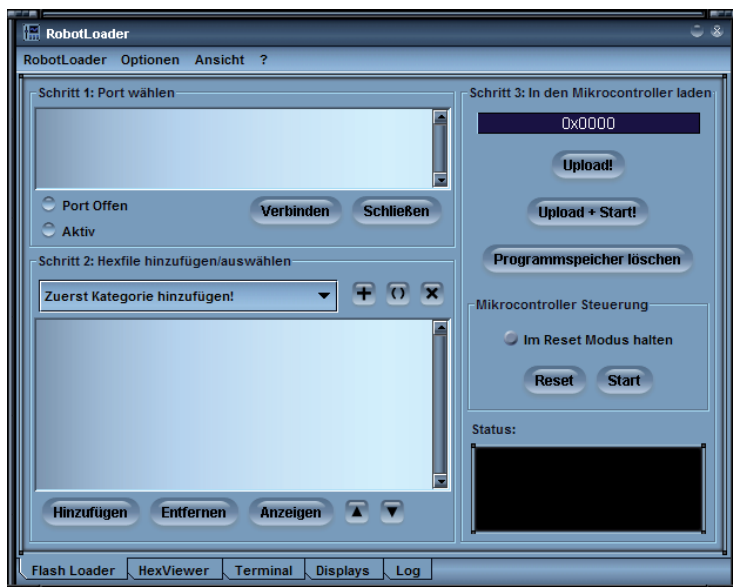
**Logiciel RobotLoader**

## 7.1. Robot Loader

Le RobotLoader a été développé pour permettre de transférer facilement de nouveaux programmes dans le Robot Arm ainsi que dans tous nos robots (dans la mesure où ils disposent d'un microcontrôleur et d'un bootloader compatible).

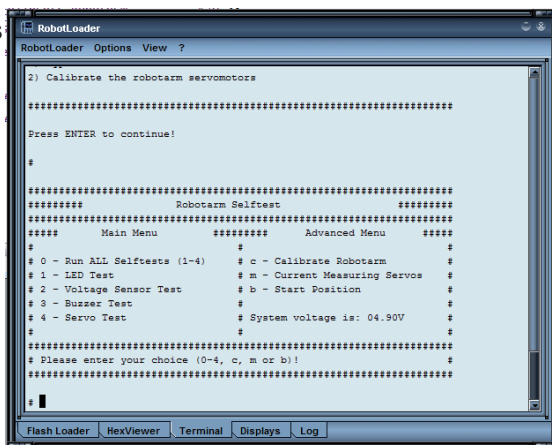
### RobotLoader

*Lorsque la tension < 6,7 Volt, s'affiche, un avertissement apparaît.*



Il contient par ailleurs quelques fonctions supplémentaires utiles tel qu'un programme de terminal simple.

### Fenêtre du terminal





## 7.2. Connexion de l'interface USB pour Windows

Les utilisateurs de Linux peuvent sauter ce chapitre.

L'installation d'une interface USB peut se faire de différentes façons. La plus simple consiste à **installer le driver AVANT la connexion du matériel**.

Le CD contient un programme d'installation pour le driver.

Pour les systèmes **32 et 64 Bit Windows 7, XP, Vista, Server 2003 et 2000**:

<Lecteur CD-ROM>:\Software\USB\_DRIVER\Win2k\_XP\CDM\_Setup.exe

Malheureusement un programme aussi facile n'existe pas pour les anciens systèmes **Win98SE/Me**. Il faut installer manuellement un driver plus ancien après avoir branché le matériel (voir ci-dessous).

Il suffit d'exécuter le programme d'installation CDM. A part une simple confirmation après l'installation du pilote, il ne se passe rien.

Ensuite vous pouvez brancher l'interface USB sur le PC. **NE LE BRANCHEZ PAS ENCORE SUR LE ROBOT!** Branchez uniquement le cordon USB sur le PC en veillant à tenir la platine de l'interface USB seulement par le bord ou par la fiche USB ou bien par la coque en plastique de la fiche de programmation (voir consignes de sécurité sur les décharges électrostatiques!). Evitez de toucher des composants sur la platine, des points de soudure ou des contacts de la fiche sauf en cas de nécessité absolue afin d'éviter des décharges électrostatiques!

Le driver installé auparavant est utilisé maintenant pour le matériels sans que vous ayez encore quelque chose à faire. Sur Windows XP/2000, de petites bulles de texte apparaissent au-dessus de la barre des tâches. Le dernier message devrait être quelque chose comme "Le matériel a été installé et prêt à l'emploi".

**Si vous avez branché l'interface USB avant l'installation (ou si vous utilisez Win98/Me)** – ce n'est pas grave. Windows vous demandera le pilote. Cette méthode d'installation est également possible. Le pilote se trouve également sous forme décompressée sur le CD!

Si c'est votre cas, une fenêtre de dialogue apparaît généralement (sous Windows) pour installer le nouveau pilote. Vous devez indiquer le chemin au système où il peut trouver le pilote. Sous Windows 2000/XP, vous devez d'abord sélectionner l'installation manuelle et ne pas chercher un service Internet ou autre. Dans notre cas, le pilote se trouve sur le CD dans les répertoires indiqués ci-dessus.

Donc, indiquez simplement le répertoire selon votre version Windows et éventuellement encore quelques fichiers que le système ne trouve pas tout seul (ils sont tous dans les répertoires indiqués ci-après!)

Sous Windows XP et des versions ultérieures, il y a souvent une fenêtre (mais normalement pas ici puisque les pilotes FTDI sont signés) qui prévient que le pilote n'a pas été signé/vérifié par Microsoft. Cela n'a pas d'importance et vous pouvez continuer l'installation.

## **Mise en Service**

Pour les système **32 et 64 Bit Windows 7, XP, Vista, Server 2003 et 2000:**

<Lecteur CD-ROM>:\Software\USB\_DRIVER\Win2k\_XP\FTDI\_CDM2\

Pour des systèmes plus anciens **Windows 98SE/Me:**

<Lecteur CD-ROM>:\Software\USB\_DRIVER\Win98SE\_ME\FTDI\_D2XX\

Certaines versions Windows plus anciennes comme Win98SE peuvent nécessiter un redémarrage. ATTENTION: Sous Win98/Me seulement un des deux pilotes fonctionne: Virtual Comport ou le pilote D2XX de chez FTDI! Malheureusement il n'y a pas de driver qui intègre les deux fonctions et il n'existe normalement pas de port COM virtuel puisque le RP6Loader utilise d'office le driver D2XX sous Windows (vous pouvez changer cela - contactez éventuellement notre équipe d'assistance!).

## **Vérifiez la bonne connexion du matériel**

Afin de vérifier si le matériel a été correctement installé, vous pouvez utiliser aussi le gestionnaire de périphériques sous Windows XP, 2003 et 2000 à la place du RobotLoader: Clic droit sur le poste de travail --> Propriétés --> Matériel --> Gestionnaire de périphériques.

OU BIEN: Démarrer --> Paramètres --> Panneau de configuration --> Performances et maintenance --> Système --> Matériel --> Gestionnaire de périphériques et regarder dans l'arborescence sous "Ports (COM et LPT) si vous voyez un "Port série USB (COMX)" - le X représentant le numéro du port ou bien regarder sous "Contrôleur de bus USB" si vous trouvez un "USB Serial Converter".

### **Désinstallation ultérieure du pilote**

Si vous souhaitez désinstaller le driver un jour (mais pas maintenant! Ce n'est qu'une indication pour le jour où le problème se présentera): Si vous avez utilisé le programme d'installation CDM, vous pouvez le faire directement par Démarrer --> Paramètres --> Panneau de configuration --> Logiciel. Dans la liste qui s'ouvre, vous trouverez "FTDI USB Serial Converter Drivers" – sélectionnez le pilote et cliquez sur "désinstaller".

Si vous avez installé le driver manuellement, vous pouvez exécuter le programme "FTUNIN.exe" dans le répertoire du pilote USB correspondant à votre système! Attention: L'adaptateur USB-->RS232 avec puce FTDI utilise généralement aussi ce pilote!

## 7.3. Connexion de l'interface USB sous Linux

Les utilisateurs de Windows peuvent sauter ce chapitre.

Linux avec un noyau 2.4.20 ou plus possède déjà le driver nécessaire (tout du moins pour le prédécesseur compatible de la puce FT232BM sur notre interface USB, la FT232R). Le matériel est reconnu automatiquement et vous n'avez rien d'autre à faire. En cas de problème, vous obtiendrez des pilotes Linux (et le support ainsi qu'éventuellement des drivers plus récents) directement chez FTDI:

<http://www.ftdichip.com/>

Sous Linux, après avoir branché le matériel, vous pouvez voir avec:  
`cat /proc/tty/driver/usbserial`

si le port série USB a été correctement installé. C'est tout ce que vous avez à faire normalement.

Il faut savoir que sous Windows le RobotLoader utilise les drivers D2XX et que la totalité des désignations USB apparait dans la liste des ports (p.ex. „USB0 | Robot USB Interface | serialNumber“) tandis que sous Linux, ce sont les désignations des ports COM virtuels `/dev/ttyUSB0`, `/dev/ttyUSB1` etc.. Les ports COM normaux apparaissent aussi comme `“dev/ttyS0“` etc.. Faites des essais pour trouver le bon port.

Pour Linux, il n'existe malheureusement pas de driver si facile à installer qui fasse les deux. C'est pourquoi il était plus sensé ici d'utiliser les drivers Virtual Comport qui existent de toute façon déjà dans le noyau. Les drivers D2XX nécessiteraient par ailleurs beaucoup de travail manuel lors de l'installation...

### Terminer l'installation du logiciel

Voilà tout ce qu'il y avait à faire pour l'installation du logiciel et de l'interface USB. Copiez maintenant les fichiers les plus importants du CD sur le disque dur (surtout le dossier complet “Documentation” et, si cela n'a pas encore été fait, les programmes d'exemple). Cela vous évite de chercher constamment le CD si vous avez besoin de ces fichiers. Les dossiers sur le CD sont tous nommés de telle façon que vous pouvez les attribuer facilement aux paquets de logiciel ou à la documentation concernés.

Si vous égarez un jour le CD, vous pouvez télécharger les fichiers les plus importants comme ce manuel, le RobotLoader et les programmes d'exemple du site d'AREXX. Vous y trouverez également les liens vers les autres paquets de logiciel dont vous aurez besoin.

## 7.4. Test de l'Interface USB et Démarrage du RobotLoader

Maintenant nous allons tester le téléchargement du programme via l'interface USB. Branchez l'interface USB sur le PC (commencez toujours par le PC) et ensuite le câble en nappe à 10 contacts sur le connecteur "PROG/UART" du Robot Arm! (Le Robot Arm DOIT ETRE HORS TENSION!) Le câble en nappe est protégé mécaniquement contre l'inversion de polarité. Dans la mesure où on ne le force pas, il ne peut pas être branché à l'envers.



Démarrez maintenant le RobotLoader. Selon la langue sélectionnée, les menus peuvent avoir des noms un peu différents. Sur les impressions écran apparaît la version anglaise. Vous pouvez sélectionner la langue par le menu „Options->Preferences“ et choisir dans „Language /Sprache“ la langue allemande ou anglaise et cliquer sur OK.

Une fois que vous avez sélectionné la langue, il faut redémarrer le RobotLoader pour que les changements prennent effet.

### Ouvrir un port sous Windows



Vous pouvez sélectionner le port USB. Si aucun autre adaptateur USB->Série avec contrôleur FTDI est branché sur le PC, vous ne voyez qu'une seule option dans la liste des ports que vous devez alors sélectionner.

S'il existe plusieurs ports, vous pouvez identifier le port à l'aide du nom "Robot USB Interface" (ou „FT232R USB UART“). Le numéro de série programmée apparaît derrière le nom.

Si aucun port n'est affiché, vous pouvez actualiser la liste dans le menu par "RobotLoader-->Refresh Portlist" („RobotLoader-->mettre à jour la liste des ports“).



### ATTENTION!

*la tension est < 6,7 Volts, un avertissement s'affiche.*

## 7.5. Ouvrir un port sous Linux

Sous Linux, l'adaptateur USB-Série est traité comme un port COM ordinaire. L'installation du pilote D2XX de FTDI sous Linux ne serait pas si simple et les pilotes normaux Virtual Comport (VCP) sont de toute façon déjà contenus dans les noyaux Linux actuels. Tout fonctionne presque exactement comme sous Windows, il faut juste savoir quel nom porte l'interface USB du Robot Arm et veiller à ne pas déconnecter le port USB du PC tant que la connexion est ouverte (sinon il faudra éventuellement redémarrer le RobotLoader pour que la connexion fonctionne à nouveau). Sous Linux, les ports COM virtuels s'appellent `"/dev/ttyUSBx"`, x étant un numéro p.ex. `"/dev/ttyUSB0"` ou `"/dev/ttyUSB1"`. Les ports COM normaux s'appellent sous Linux `"/dev/ttyS0"`, `"/dev/ttyS1"` etc.. Ils apparaissent également dans la liste des ports s'ils existent.

Si vous disposez de plusieurs ports, le RobotLoader se rappelle quel port vous avez utilisé en dernier et le sélectionne automatiquement lors de chaque démarrage du programme (en général la plupart des réglages et sélections sont maintenus).

Cliquez maintenant sur le bouton "Connect" (=„connecter“) ! Le RobotLoader ouvre le port et teste la communication avec le bootloader sur le robot. Dans la zone noire "Status" en bas vous devez lire le message "Connected to: Robot Arm ..." avec une information sur la tension mesurée à l'instant. Sinon, essayez à nouveau. Si cela ne fonctionne toujours pas, il y a une erreur! Eteignez immédiatement le robot et commencez à chercher l'erreur.



**ATTENTION!**

***Lorsque la tension est < 6,7 Volts, un avertissement s'affiche.***

## 7.6. AUTO-TEST

Dès que le Robot Arm est mis sous tension, la LED de tension jaune s'allume.

La LED d'état s'éteint lors du téléchargement d'un fichier HEX. Dès le démarrage d'un programme, la LED d'état s'allume en rouge. Lorsque l'état du robot passe en "Terminé", elle s'allume en vert.

Lorsque cela a fonctionné, vous pouvez exécuter un petit programme d'auto-test afin de vérifier le bon fonctionnement de tous les systèmes du robot. A cet effet, cliquez sur le bouton "Add" (ajouter) en bas dans la fenêtre du Robot-Loader et sélectionnez le fichier RobotArmExamples [MINI], „Example\_11\_Selftest\RobotArm\_Selftest.hex“ dans le répertoire d'exemples. Ce fichier contient le programme d'auto-test au format hexadécimal, ce qui explique leur nom. Le fichier que vous venez de sélectionner, va apparaître ensuite dans la liste. Plus tard, vous pourrez ajouter d'autres fichiers hexadécimaux de vos propres programmes et des programmes d'exemple (voir impression écran: ici quelques fichiers hexadécimaux ont déjà été ajoutés). Le Robot Loader est en mesure de gérer des catégories diverses de fichiers hexadécimaux.

Cela permet de trier les fichiers d'une manière plus claire, p.ex. lorsqu'on a plusieurs modules d'extension programmables sur le Robot Arm ou si on utilise différentes variantes de programmes. A la fin du programme, la liste est toujours sauvegardée automatiquement. Evidemment seuls les chemins vers les fichiers hexadécimaux sont sauvegardés - et non pas les fichiers eux-mêmes. Lorsque vous travaillez sur un programme, vous n'avez à ajouter et à sélectionner le fichier hexadécimal qu'une seule fois. Ensuite, après chaque nouvelle traduction du programme, vous pouvez charger le nouveau programme dans le microcontrôleur (également avec les touches [STRG+D] ou [STRG+Y] pour démarrer le programme immédiatement après le transfert). Selon le système d'exploitation, les noms des chemins varient fortement. Néanmoins vous pouvez utiliser le RobotLoader directement sous Windows ou Linux car il y a des listes séparées pour Windows et Linux.

Vous pouvez continuer maintenant avec les autres programmes d'exemples (Exemples) du Robot Arm ou bien commencer à programmer votre propre logiciel.



**ATTENTION!**

***Lorsque la tension est < 6,7 Volts, un avertissement s'affiche.***

Pour ce faire, sélectionnez le fichier „RobotArm\_Selftest.hex“ dans la liste et cliquez sur le bouton “Upload!” en haut à droite en dessous de la barre de progression.

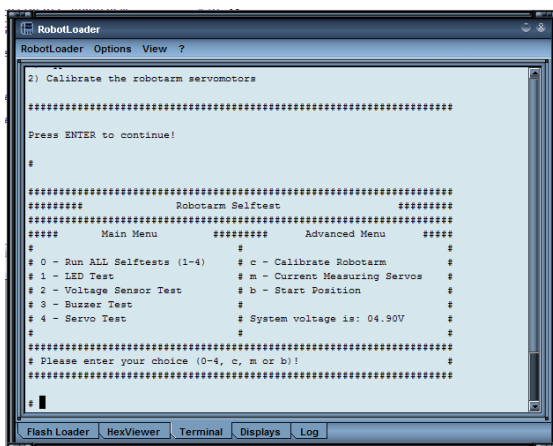
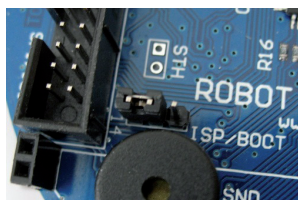
Le programme est maintenant transféré dans le processeur MEGA64 sur le Robot Arm. Cela ne devrait pas prendre plus que quelques secondes (au maximum 5 secondes pour le programme d’auto-test).

Basculez ensuite sur l’onglet “Terminal” (= „Tabs“ tout en bas de la fenêtre du programme!) ! Vous pouvez également aller dans le terminal par le menu “View” (“Affichage”).



### ATTENTION!

*Pour le fonctionnement normal, le jumper “ISP/BOOT” doit se trouver dans la position indiquée sur la photo.*



Vous pouvez maintenant effectuer l’auto-test et le calibrage du Robot Arm. Démarrez le programme en appuyant sur le commutateur Start/Stop Reset sur le Robot ARM. Ultérieurement, vous pourrez bien sûr lancer le programme par le menu RobotLoader -->Start ou la combinaison de touches [STRG]+[S]. De cette manière, vous pouvez tester en même temps si le commutateur fonctionne correctement!

Si une erreur se produit dans l’auto-test, coupez tout de suite l’alimentation du robot et commencez à chercher l’erreur!

**COMMENCEZ DE PREFERENCE AVEC LE CALIBRAGE DU ROBOT ARM!  
VOIR PAGE 41.**



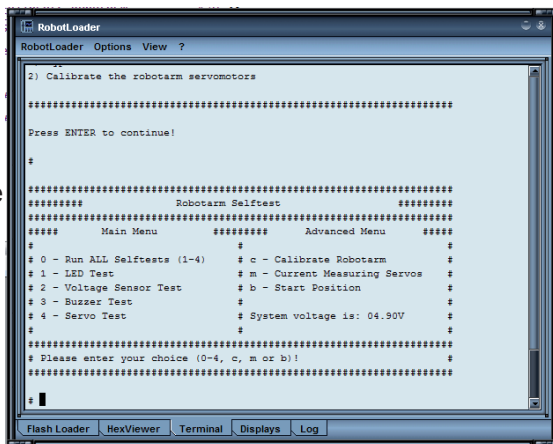
## 7.7. Calibrage

Démarrez le programme de calibrage pour calibrer le Robot Arm.

Pour cela, cliquez en bas de la fenêtre du RobotLoader sur le bouton “Add” (ajouter) et sélectionnez le fichier RobotArmExamples [MINI], „Example\_11\_Selftest\RobotArm\_Selftest.hex“ dans le répertoire d'exemples.

Ce fichier contient le programme d'auto-test au format hexadécimal. Le fichier sélectionné apparaît ensuite dans la liste (voir impression écran).

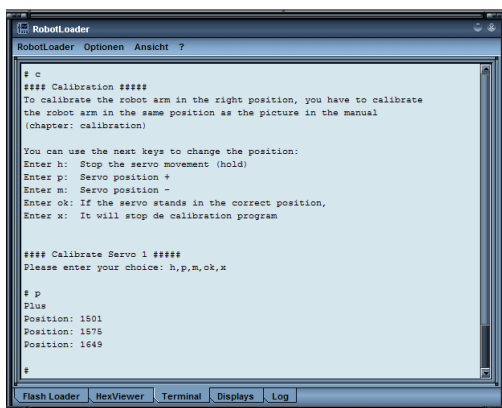
Sélectionnez C  
(C - Calibrate) du programme  
de calibrage pour  
démarrer le calibrage.



```
RobotLoader
RobotLoader Options View ?

2) Calibrate the robotarm servomotors

*****
Press ENTER to continue!
#
*****
***** Robotarm Selftest *****
***** Main Menu ***** Advanced Menu *****
# # #
# 0 - Run ALL Selftests (1-4) # c - Calibrate Robotarm #
# 1 - LED Test # m - Current Measuring Servos #
# 2 - Voltage Sensor Test # b - Start Position #
# 3 - Buzzer Test # #
# 4 - Servo Test # System voltage is: 04.90V #
# #
*****
# Please enter your choice (0-4, c, m or b)! #
*****
#
```



```
RobotLoader
RobotLoader Optionen Ansicht ?

# c
#### Calibration ####
To calibrate the robot arm in the right position, you have to calibrate
the robot arm in the same position as the picture in the manual
(chapter: calibration)

You can use the next keys to change the position:
Enter h: Stop the servo movement (hold)
Enter p: Servo position +
Enter m: Servo position -
Enter ok: If the servo stands in the correct position,
Enter x: It will stop de calibration program

#### Calibrate Servo 1 ####
Please enter your choice: h,p,m,ok,x

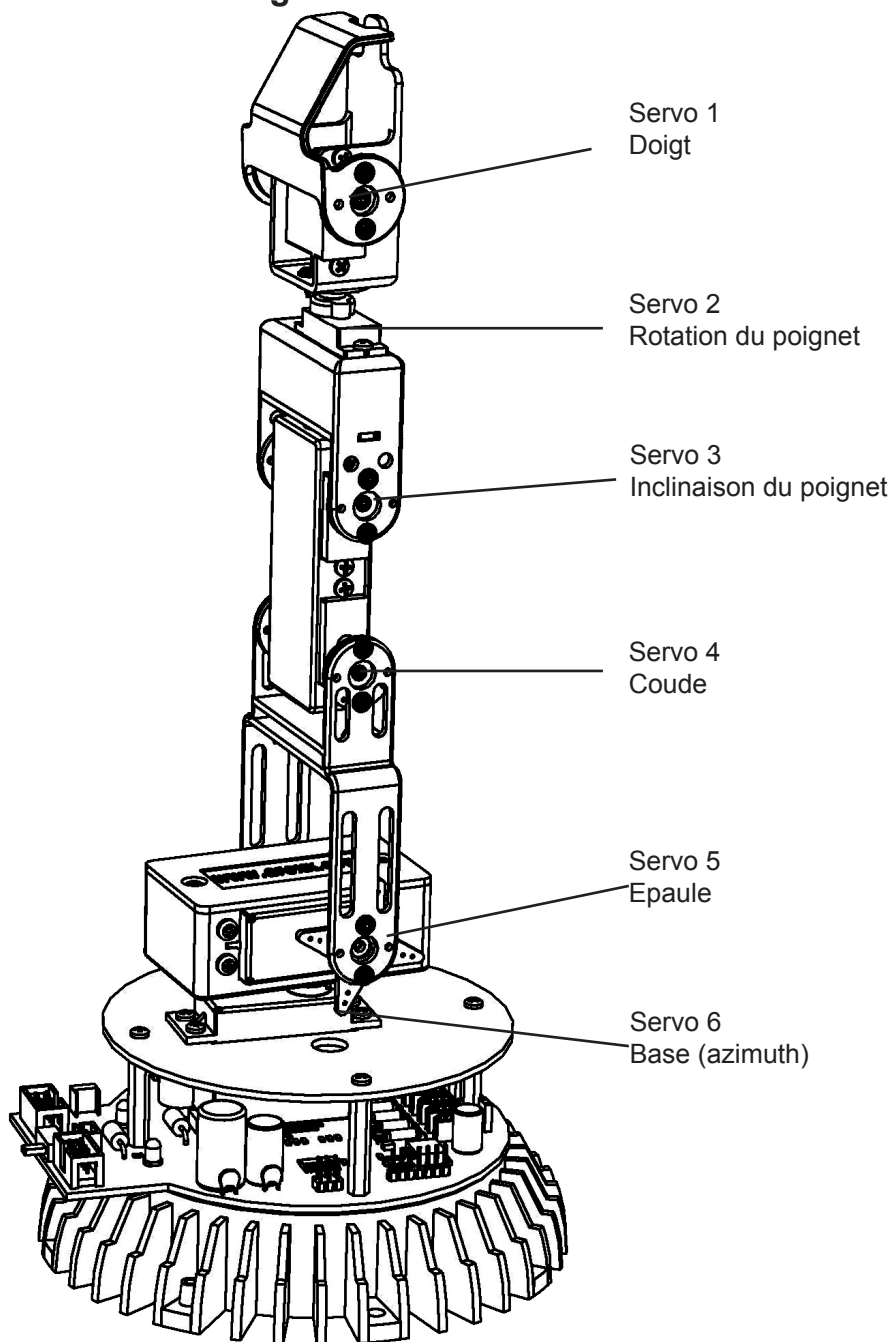
# p
Plus
Position: 1501
Position: 1575
Position: 1649
#
```

Tournez tous les servo moteurs  
en position centrale pour que le  
Robot Arm se présente comme  
en page 42.

Les servo moteurs 2-6 se trouvent à  
peu près en position centrale et  
le doigt (servo 1) est presque fermé.

Lorsque le calibrage (C - Calibrate) est terminé, vous pouvez effectuer l'auto-test suivant du Robot Arm. Le résultat du calibrage est enregistré dans l'ATMEGA.

## Position de calibrage



## 7.8. Test Clavier

Ce kit est livré avec un clavier qui peut être connecté sur le Robot Arm. C'est une option intéressante pour des démonstrations simples et nous permet de nous entraîner en même temps à piloter un bras de robot via un clavier.

Le clavier possède 6 touches pour piloter le Robot Arm et 4 touches supplémentaires pour des extensions ultérieures.

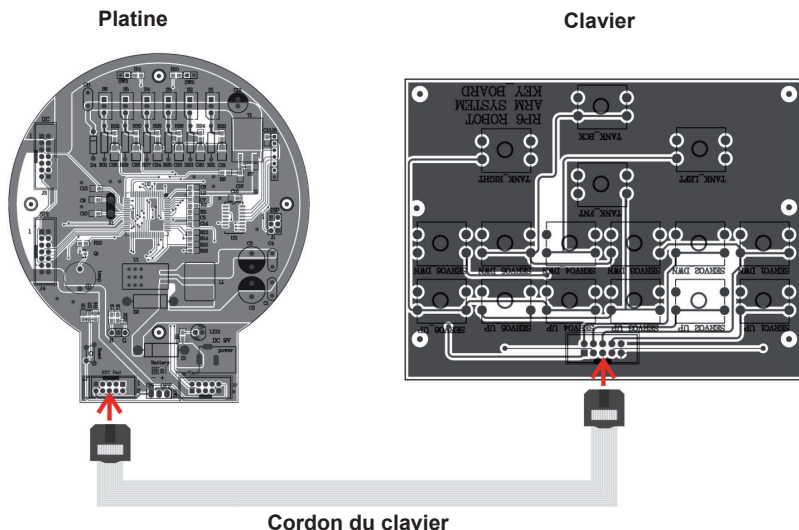
Pour guider le Robot Arm via le clavier, nous devons transférer le programme HEX nécessaire dans le microprocesseur du Robot Arm.

Cliquez en bas de la fenêtre du RobotLoader sur le bouton "Add" (ajouter) et sélectionnez le fichier RobotArmExamples, „RobotArm\_Key\_Board.hex“ dans le répertoire d'exemples.

Sélectionnez le fichier "RobotArm\_Key\_Board.hex" dans la liste et cliquez sur la touche "Upload!" en haut à droite en dessous de la barre de progression.

Ensuite vous pouvez commander le Robot Arm tout simplement à l'aide des touches sur le clavier.

### ***UTILISEZ LE BON LOGICIEL V 3PRO OU V3 MINI !***



## 8.0. Logiciel RACS

Le RACS (Robot Arm Control Software) est la méthode la plus simple pour contrôler et programmer le Robot Arm. La programmation par la méthode RACS fait appel au logiciel RobotLoader et l'adaptateur de programmation USB.

Avant de pouvoir utiliser le robot, vous devez charger le logiciel hexadécimal RAC-PRO.hex dans la mémoire flash du processeur.

Branchez le cordon de programmation/contrôle sur le port USB de l'ordinateur et démarrez ensuite le logiciel Loader. L'interface utilisateur suivante s'affiche :

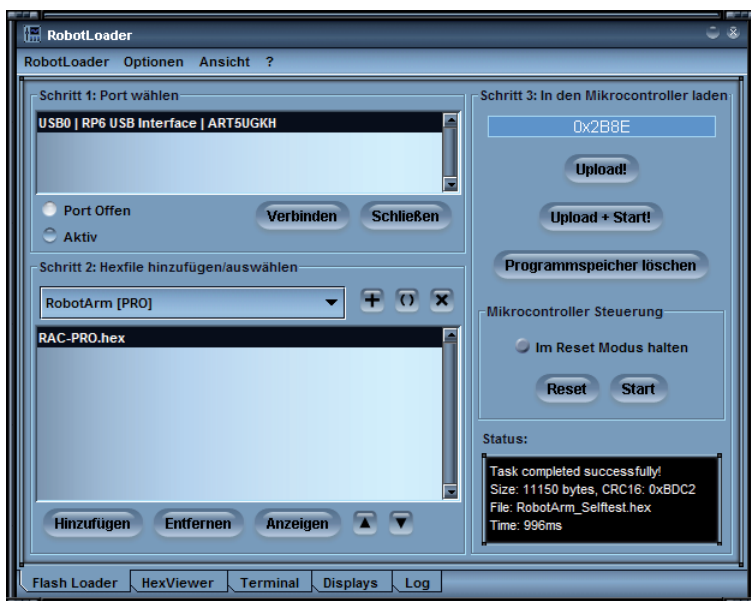


Fig. 1

Si aucun port USB ne s'affiche dans la liste „Schritt 1: Port wählen“ (Etape 1: Sélectionner le port), assurez-vous que le cordon est bien connecté et que les pilotes du programmeur sont installés. Vous pouvez interroger de nouveau la liste des ports dans le menu:

**RobotLoader -> Portliste aktualisieren (actualiser la liste des ports).**

Sélectionnez ensuite le port et cliquez sur “Verbinden” (connecter).

Sélectionnez le bon fichier .hex à l'étape 2.

– Pour cela, cliquez simplement sur “Hinzufügen” (Ajouter): **RAC-PRO.HEX**

**UTILISEZ LE BON LOGICIEL V3PRO OU V3 MINI !**

A l'étape 3, le fichier est transféré en cliquant sur le bouton "Upload".

Pour l'utilisation du Robot Arm, il faut fermer la connexion en cliquant sur le bouton "Schliessen" (fermer) à l'étape 1. Si vous fermez le programme, la connexion est automatiquement interrompue.

Veillez à ce qu'il n'y ait aucune connexion entre le logiciel Loader et le Robot Arm, sinon vous ne pouvez pas commander le robot avec le logiciel RACS.

## 8.1. Manuel d'utilisation du RACS

Le logiciel RACS permet de commander le Robot Arm d'une façon très simple. Une connexion est établie par le cordon de programmation/commande et ensuite les moteurs du Robot Arm sont commandés par la souris qui déplace les curseurs. Les positions actuelles peuvent être mémorisées, modifiées et effacées dans la fenêtre de liste dans la partie inférieure de l'interface utilisateur. De cette façon, vous établissez une liste avec les différentes positions qui peut être sauvegardée sur l'ordinateur sous forme de fichier. Pour cela, cliquez sur le bouton "Save" (enregistrer). Vous pouvez charger cette liste à tout moment.



***Le robot surveille le courant moteur de chaque servomoteur. Si la valeur limite d'un servo est dépassée, p.ex. dans le cas d'une collision ou d'une surcharge, le texte dans le logiciel RACS commence à clignoter. Dans ce cas, le robot doit être remis immédiatement dans sa dernière position ou bien l'alimentation servo dans le logiciel RACS doit être désactivée (décochez la case "servo").***

***Sinon le robot risque d'être endommagé définitivement !!!***

## 8.2. Raccordement RACS

1. Démarrez le logiciel Robot Arm Control par un double-clic.  
La fenêtre suivante s'affiche:

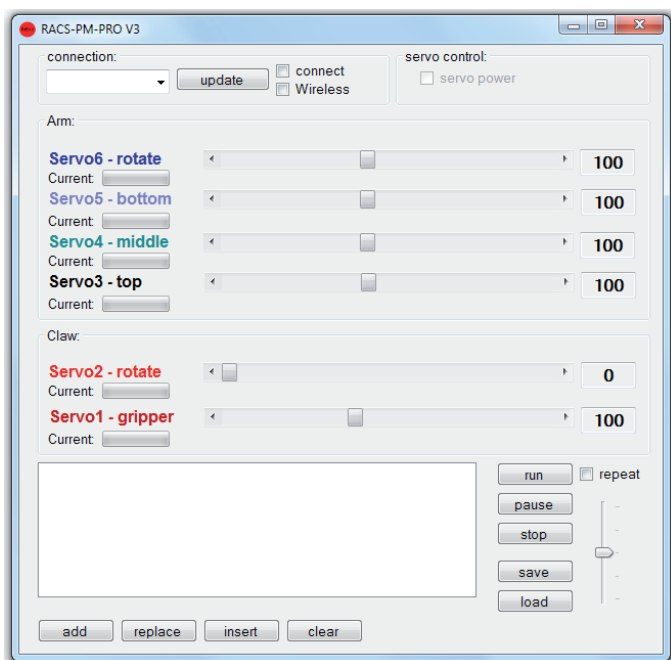


Fig. 2

2. Dans le menu déroulant apparaissent toutes les interface série

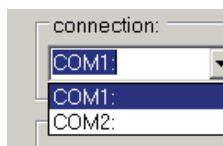


Fig. 3

3. Branchez le programmeur USB
4. Cliquez sur le bouton "Update". En regardant à nouveau le menu déroulant, une interface supplémentaire s'affiche.  
Cette interface a été initialisée par le branchement du programmeur.

**Attention: La dénomination des interfaces varie d'un ordinateur à l'autre!**

5. Marquez cette nouvelle interface

**Fig. 4**



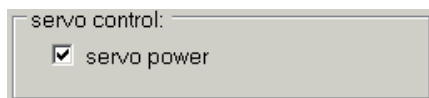
6. Cochez la case “Connect”

**Fig. 5**



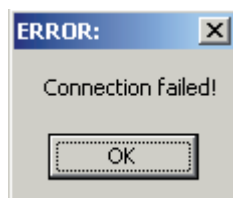
7. Cochez la case “servo power”

**Fig. 6**



8. Pilotez les servos en déplaçant les curseurs.  
Si une erreur s'est produite lors du raccordement, le message ci-dessous s'affiche. Vous devez recommencer le processus de raccordement (répéter les étapes 2 à 6 et vérifier l'interface).

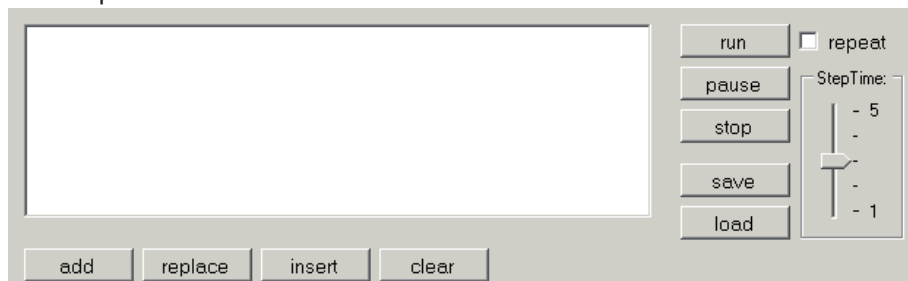
**Fig. 7**



**UTILISEZ LE BON LOGICIEL V 3PRO OU V3 MINI !**

### 8.3. RACS – Commande de position automatisée

Vous disposez des éléments de commande ci-dessous:



- Add:** La position actuelle du curseur est ajoutée à la liste
- Replace:** Le point de la liste sélectionné sera remplacé par les positions actuelles du curseur
- Insert:** Les positions actuelles du curseur sont insérées dans la liste au-dessus de l'élément de liste sélectionné
- Clear:** L'élément de liste sélectionné est effacé
- Save:** Les éléments de liste sont enregistrés dans un fichier
- Load:** Les éléments de liste sont importés d'un fichier (Attention: les éléments présents seront écrasés!)
- Run:** Les éléments de liste sont exécutés en partant du haut. Si l'option "Repeat" est activée, le Robot Arm exécute les éléments en boucle.
- Step Time:** Le Step Time détermine le temps d'attente (en secondes) du robot avant d'exécuter la position suivante sur la liste. Si la liste ne contient que des parcours de déplacement très courts, cette durée peut être très courte. Si, au contraire, des parcours très longs ont été programmés, tels que des mouvements à 180° des servos, il faut choisir une durée plus longue car le robot risque de ne pas atteindre sa position cible et de se diriger prématurément vers la position suivante de la liste.
- Pause:** L'exécution de la liste est mise en pause
- Stop:** L'exécution est arrêtée

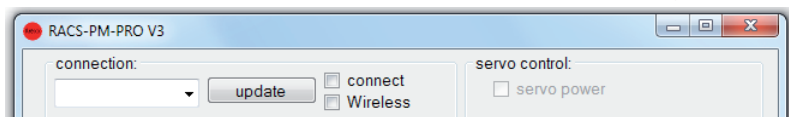


## 8.4. RACS – SANS FIL

Le logiciel RACS et le set ARX-APC-220 d'AREXX vous permet de commander le Robot Arm sans fil. Nous vous expliquons ci-après étape par étape comment cela fonctionne.

- Branchez le programmeur RP6v2 et l'APC-220 sur le PC tel que décrit dans le CD.
- Branchez l'autre APC sur le PCB du Robot Arm (voir p.16).
- Chargez le fichier HEX Wireless Racs dans le processeur du Robot Arm retirez le cordon USB.
- Sélectionnez le bon port COM dans le logiciel RACS.
  - \* **Appuyez sur l'interrupteur Start (Platine du Robot Arm)**
  - \* Activez --> **Wireless - Connect** et **Servo Power**

Maintenant vous pouvez commander le Robot Arm par radio.



## 8.5. PROGRAMME ANDROID

### 8.5.1 PROGRAMME ANDROID

Notre set ARX-BT03 permet également de commander le Robot Arm par Bluetooth et une application ANDROID.

#### Procédez de la manière suivante:

- Branchez le module Bluetooth sur la platine du Robot Arm (voir p. 16).
- Chargez le fichier HEX ANDROID dans le processeur du Robot Arm.
- Chargez le fichier APK Robot Arm AREXX dans votre téléphone portable ou votre tablette. Ce fichier se trouve sur le CD et dans le Google Play Store.



***Voir également le manuel chez AREXX  
APC-220, Android ou module WIFI.***



## 9.0. Programmation du Robot Arm

Nous arrivons progressivement à la programmation du robot.

### Installation de l'éditeur de texte source

Tout d'abord, nous devons nous créer un petit environnement de développement puisqu'il faudra bien rentrer le "texte source" (appelé aussi code source ou "sourcecode" en anglais) de notre programme C dans l'ordinateur!

Pour ce faire, nous n'allons certainement pas utiliser des programmes tels que OpenOffice ou Word! Nous le soulignons ici car ce n'est peut-être pas une évidence pour tout le monde. Ces programmes sont parfaits pour écrire des manuels comme celui-ci mais ne conviennent pas du tout à la programmation. Un texte source est du texte pur sans aucune mise en forme. Le compilateur ne s'intéresse pas à la taille de la police ou à la couleur...

Pour une personne, il est évidemment plus clair lorsque certains mots-clé ou des types de texte sont mis en évidence automatiquement par une couleur et c'est ce que fait, entre autres, l'éditeur de texte source que nous allons utiliser: Programmers Notepad 2 (abrégié ci-après par PN2) (ATTENTION: Sous Linux, vous devez utiliser un autre éditeur qui offre à peu près les mêmes fonctions que PN2. Généralement, plusieurs éditeurs sont déjà préinstallés tels que kate, gedit, exmacs et autres). Outre la mise en évidence de mots-clé et autre (appelé "coloration syntaxique"), il offre une gestion de projet rudimentaire. Ainsi, vous pouvez organiser plusieurs fichiers de texte source en projets et faire apparaître dans une liste tous les fichiers appartenant à un projet. Par ailleurs, vous pouvez facilement appeler des programmes comme l'AVR-GCC à partir de PN2 et faire traduire aisément les programmes par un élément de menu. L'AVR-GCC est normalement un pur programme de lignes de commandes sans surface graphique...

Vous trouverez des versions récentes de Programmers Notepad sur la page d'accueil projet: <http://www.pnotepad.org/>

**Les versions les plus récentes de WINAVR dispensent de la création d'options de menu!**

### ATTENTION:

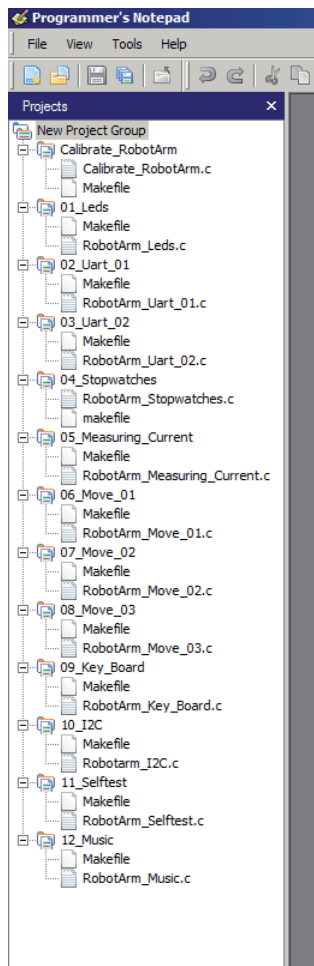
**Dans ce chapitre nous n'expliquons plus comment vous devez créer des options de menu puisqu'elles sont déjà créées dans les versions WINAVR les plus récentes!**

Voir page 56 “Ouvrir et compiler un projet d'exemple” pour savoir comment procéder.

Lorsque vous avez ouvert un projet d'exemple, il devrait se présenter à l'écran PN2 de la manière suivante:

***Fichier „Robot ArmExamples.ppg”.***

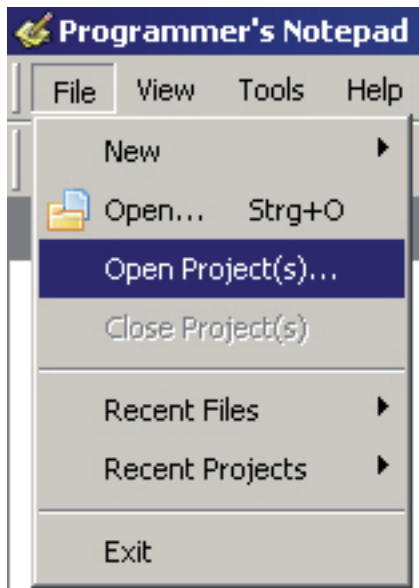
Ceci est un groupe de projet pour PN2 qui charge tous les programmes d'exemple ainsi que la RobotArm Library dans la liste des projets (“Projects”).



A gauche, vous voyez tous les projets d'exemples, à droite, l'éditeur de texte source (avec la coloration syntaxique) et en bas, l'édition des outils (dans ce cas l'édition du compilateur).

Vous pouvez faire encore beaucoup d'autres choses avec PN2 qui offre de nombreuses fonctions utiles.

## Ouvrir et compiler un projet d'exemple



Nous allons tester maintenant si tout fonctionne bien et ouvrons les projets d'exemple:

Dans le menu „File“ sélectionner l'option „Open Project(s)“.

Une fenêtre de sélection de fichier normale s'ouvre. Cherchez le dossier "Robot Arm\_Examples [MINI]" dans le dossier dans lequel vous avez enregistré les programmes d'exemple.

Ouvrez le fichier "Robot ArmExamples.ppg". C'est un groupe de projets pour PN2 qui charge tous les programmes d'exemple ainsi que la Robot Arm Library dans la liste de projets ("Projects").

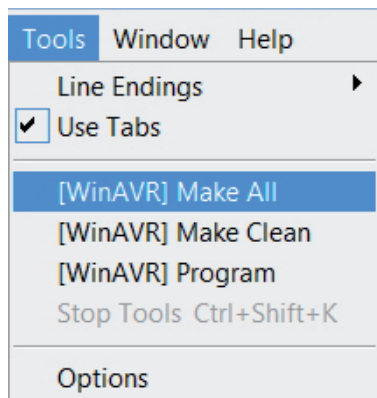
Ainsi tous les programmes d'exemples sont toujours à disposition et permettent de s'y reporter plus facilement ou rechercher des fonctions dans la Robot Arm Library etc..

Ouvrez le premier programme tout en haut de la liste ("01\_Leds" et sélectionnez le fichier "01\_Leds" ) qui apparaît au bord gauche de la fenêtre du programme. Pour cela, double-cliquez sur "01\_Leds.c"! Un éditeur de texte source s'affiche dans une fenêtre à l'intérieur du programme.

En bas de la fenêtre du programme de PN2, vous devez avoir une zone d'édition. Si ce n'est pas le cas, vous devez activer cette zone par le menu "View" --> "Output" OU, si elle est trop petite, la "tirer" avec la souris (la flèche de la souris se transforme en une double flèche en bas de la fenêtre de programme sur le bord supérieur de la zone grise dans laquelle est inscrit "Output" et vous pouvez agrandir la zone en tirant la double flèche).

Vous pouvez déjà jeter un coup d'oeil sur le programme dans l'éditeur de texte source qui vient de s'ouvrir. Cependant ce n'est pas encore nécessaire de comprendre ce qui s'y passe exactement. Nous allons l'expliquer plus loin. Pour commencer: le texte vert sont des commentaires qui ne font pas partie du programme proprement dit et ne servent qu'à la description/documentation.

Nous allons y revenir par la suite plus en détails (il existe aussi une version de ce programme SANS les commentaires pour montrer à quel point le programme est court. Les commentaires le gonflent énormément mais sont nécessaires à l'explication. La version non-documentée est également pratique pour copier le code dans vos propres programmes!).



Nous allons tester dans un premier temps si la traduction de programmes fonctionne correctement.

Le menu Tools devrait montrer les éléments de menu qui viennent d'être créés (voir fig.) ou les éléments [WinAVR] qui existent en standard dans PN. Cela n'a pas d'importance puisque cela fonctionne normalement avec les deux).

Cliquez maintenant sur "MAKE ALL"!

PN2 appelle maintenant le fichier batch „make\_all.bat“. Celle-ci appelle à son tour le programme “make“. Nous reviendrons sur “make” par la suite.

Le programme d'exemple est maintenant traduit (cela s'appelle “compiler” du mot anglais “to compile” ou bien „Compilateur“=„traducteur“) et un fichier hexadécimal est généré. Celui-ci contient le programme dans la forme traduite pour le microcontrôleur et pourra être chargé et exécuté par ce dernier. Pendant la compilation, un grand nombre de fichiers temporaires est généré (avec des terminaisons comme “.o, .lss, .map, .sym, .elf, .dep“). N'en tenez pas compte! Vous pourrez les effacer facilement avec l'outil fraîchement créé “make clean“. Seul le fichier hexadécimal vous intéresse. D'ailleurs, le fichier hexadécimal n'est pas effacé par la commande “make clean“.

Après la commande MAKE ALL du menu, l'édition suivante devrait s'afficher (fortement abrégé ici! Certaines lignes peuvent être un peu différentes):

```
> "make.exe" all
```

```
----- begin -----
```

```
avr-gcc (WinAVR 20100110) 4.3.3
```

```
Copyright (C) 2008 Free Software Foundation, Inc.
```

```
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
Size before:
```

```
AVR Memory Usage
```

```
-----
```

```
Device: atmega64
```

```
Program: 3074 bytes (4.7% Full)
```

```
(.text + .data + .bootloader)
```

```
Data: 68 bytes (1.7% Full)
```

```
(.data + .bss + .noinit)
```

```
EEPROM: 14 bytes (0.7% Full)
```

```
(.eeprom)
```

```
Compiling C: Robot Arm_Leds.c
```

```
avr-gcc -c -mmcu=atmega64 -I.
```

```
-gdwarf-2 -DF_CPU=16000000UL -Os -funsigned-char -funsigned-bitfields -fpack-  
struct -fshort-enums -Wall
```

```
-Wstrict-prototypes -Wa,-adhlns=./Robot Arm_Leds.lst -std=gnu99 -MMD -MP -MF  
.dep/Robot Arm_Leds.o.d Robot Arm_Leds.c -o Caterpillar_Leds.o
```

```
Linking: Robot Arm_Leds.elf
```

```
avr-gcc -mmcu=atmega16 -I. -gdwarf-2 -DF_CPU=16000000UL -Os -funsigned-char -funsigned-  
bitfields
```

```
Creating load file for Flash: Robot Arm_Leds.hex
```

```
Creating load file for EEPROM: Robot Arm_Leds.eep
```

```
avr-objcopy -j .eeprom --set-section-flags=.eeprom="alloc,load" \
```

```
--change-section-lma .eeprom=0 --no-change-warnings -O ihex Robot Arm_Leds.elf
```

```
Robot Arm_Leds.eep || exit 0
```

```
Size after:
```

```
AVR Memory Usage
```

```
-----
```

```
Device: atmega64
```

```
Program: 3074 bytes (4.7% Full)
```

```
(.text + .data + .bootloader)
```

```
Data: 68 bytes (1.7% Full)
```

```
(.data + .bss + .noinit)
```

```
EEPROM: 14 bytes (0.7% Full)
```

```
(.eeprom)
```

```
----- end ----->
```

```
Process Exit Code: 0
```

```
> Time Taken: 00:04
```

Le "Process Exit Code: 0" tout en bas est très important. Il signifie qu'il n'y a pas eu d'erreur lors de la compilation. Si un autre code apparaît à la place, il y a une erreur dans le code source qu'il faut corriger avant que cela puisse fonctionner. Dans ce cas, le compilateur édite plus haut divers messages d'erreur qui vous donneront plus d'informations.

Notez quand-même que "Process Exit Code: 0" n'est pas synonyme d'un programme 100% sans erreurs! Le compilateur ne détectera évidemment pas d'erreurs de réflexion dans votre programme et il n'empêchera pas non plus le robot de rentrer dans un mur ;)

**IMPORTANT:** Il peut y avoir d'autres avertissements qui sont souvent très utiles et désignent presque toujours des problèmes importants! C'est pourquoi il faut toujours les éliminer! PN2 met les avertissements et erreurs en évidence par des couleurs afin de faciliter leur identification. Il indique aussi le numéro de ligne qui pose un problème au compilateur. Si on clique sur le message coloré, le PN2 saute directement à la ligne en question dans l'éditeur concerné.

L'indication à la fin "AVR Memory Usage" est également très utile.

```
-----  
Size after:  
AVR Memory Usage  
-----  
Device: atmega64
```

```
Program:  3074 bytes (4.7% Full)  
(.text + .data + .bootloader)
```

```
Data:      68 bytes (1.7% Full)  
(.data + .bss + .noinit)
```

Cela signifie ici pour le processeur Atmega 64 que notre programme pèse 3074 bytes et que 68 bytes de RAM sont réservés pour des variables statiques (à cela s'ajoutent les zones dynamiques pour tas et pile mais cela nous mènerait trop loin.... Contentez-vous de garder toujours au moins quelques centaines de bytes de mémoire de libre). Nous avons en tout 64ko (65536 Bytes) de ROM flash et 2ko (2048 Bytes) de RAM. Sur les 64ko, 2ko sont occupés par le bootloader - donc nous ne pouvons utiliser que 62ko. Veillez toujours à ce que le programme puisse tenir dans la mémoire disponible! (Le RobotLoader ne transfère pas le programme s'il est trop grand!)

Donc, dans le programme d'exemple précédent, il reste 60414 bytes d'espace libre. D'ailleurs, le programme `Example_01_Leds.c` qui est en soi plutôt court, ne prends autant d'espace que parce que la `Robot ArmBaseLibrary` y est intégrée. Par conséquent, pas de soucis, il y a assez d'espace pour vos programmes et de si petits programmes n'ont normalement pas besoin d'autant d'espace. La bibliothèque de fonctions occupe à elle seule déjà plusieurs Ko de la mémoire flash mais vous évite aussi beaucoup de travail et c'est pour cela que vos propres programmes seront souvent relativement petits par rapport à la `Robot ArmBaseLibrary`.

Le programme que vous venez de compiler, peut maintenant être chargé dans le robot via le `RobotLoader`. A cet effet, vous devez ajouter le fichier hexadécimal que vous venez de créer, dans la liste du `RobotLoader` au moyen de la fonction "Add" (ajouter). Sélectionnez le fichier et cliquez sur le bouton "Upload!" comme vous l'avez déjà fait pour le programme d'auto-test. Ensuite vous pouvez revenir sur le terminal et regarder l'édition du programme. Il faut évidemment démarrer d'abord l'exécution du programme. Cela pourra se faire sur le terminal en appuyant sur les touches [STRG]+[S] sur le clavier ou en utilisant le menu (ou en envoyant simplement un "s" - cependant après un reset, vous devez toujours attendre un peu jusqu'à ce que le message "[READY]" s'affiche sur le terminal!). [STRG]+ [Y] est également une combinaison de touches très utile puisqu'elle charge le programme sélectionné dans le Robot Arm et le démarre immédiatement après! Cela évite de revenir du terminal sur l'onglet "Flash Loader" ou d'utiliser le menu.

Le programme d'exemple est très simple et ne consiste qu'en un petit séquenceur à LED et un peu de texte.

-



## Le mot de la fin

Nous espérons que nos robots vous auront rapproché du monde de la robotique! Tout comme nos amis japonais, nous sommes convaincus que les robots constitueront la prochaine révolution technologique après les ordinateurs et les téléphones portables. Cette révolution déclenchera également de nouvelles impulsions économiques. Malheureusement, le Japon, d'autres pays asiatiques et aussi les Etats-Unis ont d'ores et déjà largement dépassé l'Europe dans ce domaine. A la différence de l'Europe, l'enseignement technique commence en Asie déjà à l'école primaire et constitue une part importante de la formation.

Notre objectif lors du développement de nos robots ASURO, AAR-04, YETI, Caterpillar et Robot Arm est donc:

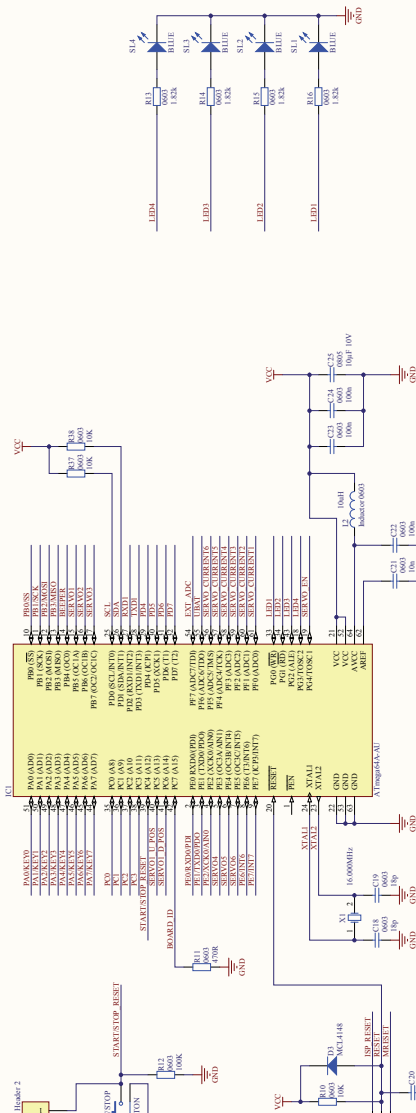
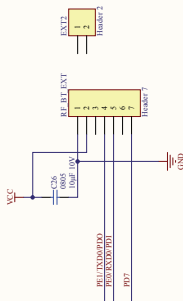
## ***TO TRAIN A SCIENTIFIC MIND***



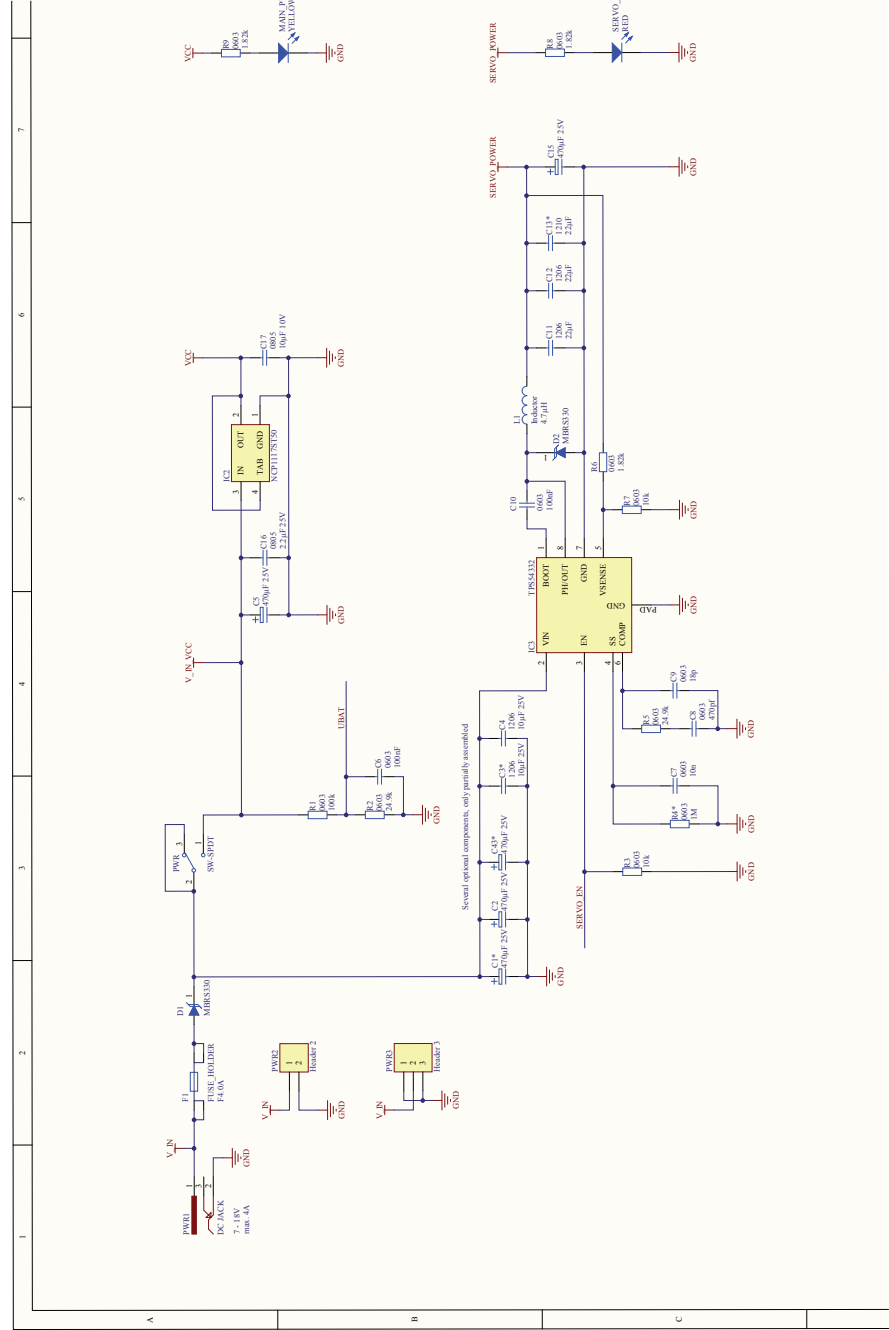
# ***APPENDICE***

# A. SCHEMA TECHNIQUE DU ROBOT ARM RA1-PRO

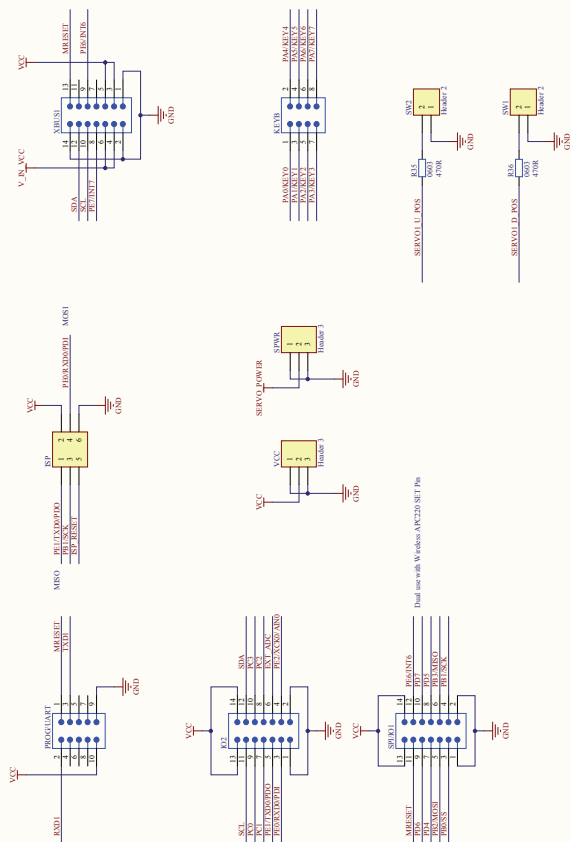
APC220 or Bluetooth Module expansion



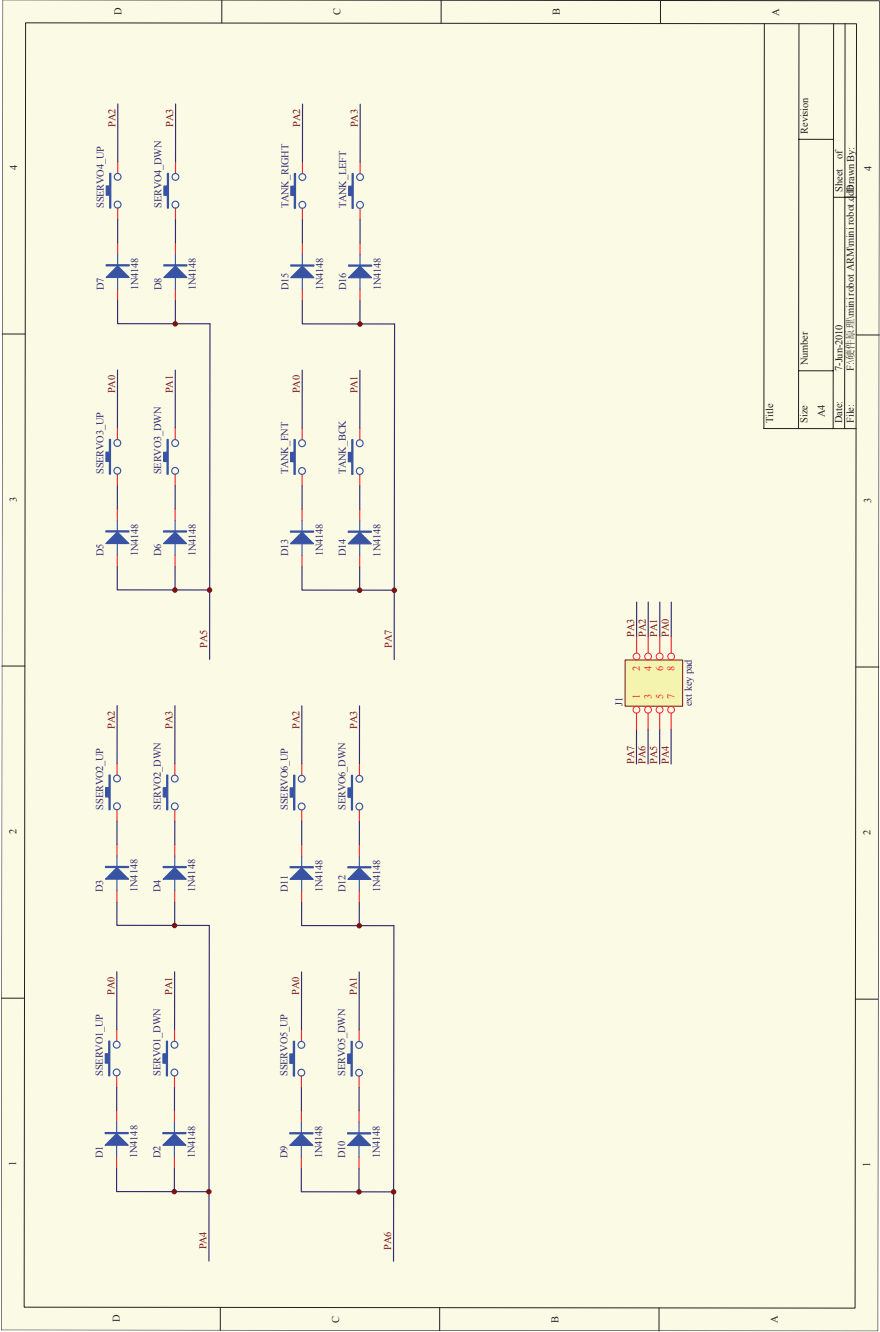
# B. SCHEMA TECHNIQUE DU POWER SUPPLY RA1-PRO



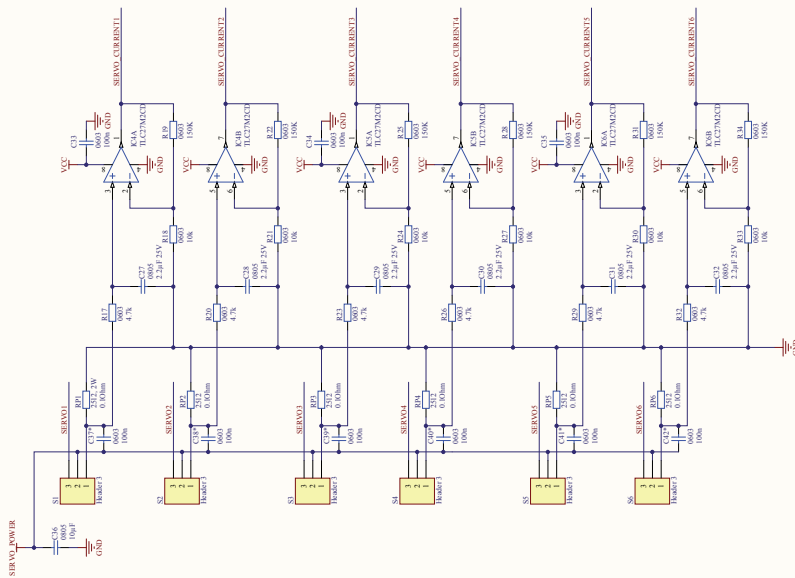
# C. SCHEMA TECHNIQUE DU CONNECTORS RA1-PRO



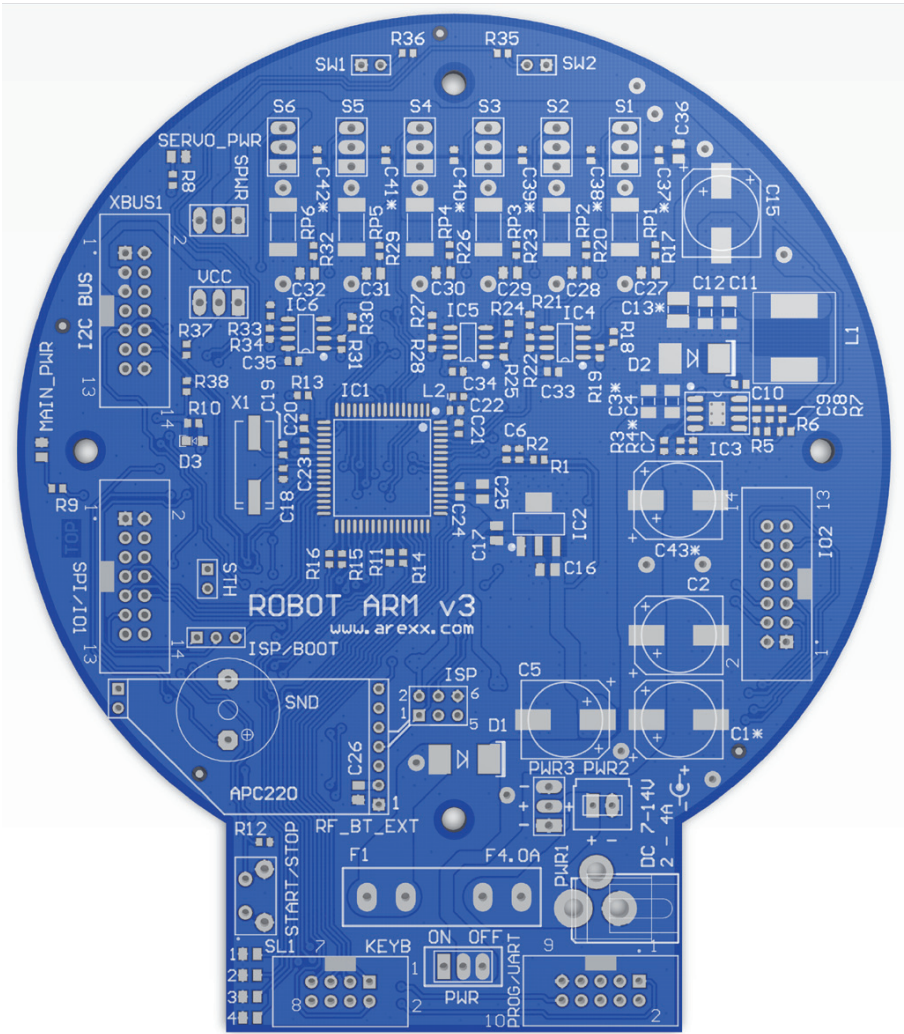
D. SCHEMA TECHNIQUE DU CLAVIER RA1-PRO



# E. SERVO MOTUER COURANT V3



F. PLATINE ROBOT ARM RA1-PRO





# G. ADAPTEUR DE PROGRAMMATION V3

