



ROBOT PER L'APPRENDIMENTO ROBOT ARM PRO V3

ISTRUZIONI: Modello RA1-PRO V3



Indice

1.Descrizione di ROBOT ARM	8
1.1.Caratteristiche:	8
1.2.Che cosa possiamo fare con Robot Arm?	9
2.Attrezzi richiesti per le modifiche	10
3.Elenco parti	11
4.Costruzione parti meccaniche.....	13
4.1.Costruzione delle principali parti meccaniche	13
4.2.Costruzione del braccio:.....	14
4.3.Assemblaggio ed allineamento della pinza	15
4.4.ASSEMBLAGGIO COMPLETO	16
5.ELETTRONICA.....	17
5.1.Caratteristiche e dati tecnici	17
5.2.Modifiche relative al circuito stampato V3 tipo RA1 PRO e RA2-HOBBY	18
5.3.Assegnazione dei connettori sul PCB.....	19
5.4.LEDS.....	20
5.5.Avviare il Robot	20
6.Installazione del software	22
6.1.Il CD-ROM di Robot Arm	22
6.2.WinAVR - per Windows	23
6.3.AVR-GCC, avr-libc e avr-binutils - per Linux.....	24
6.4.Script per installazione automatica	27
6.5.GCC per AVR.....	28
6.6.AVR Libc.....	28
6.7.Imposta il Path	29
6.8.Installazione manuale.....	30
6.9.Binutils per AVR.....	31
6.10.Java 6	31
6.11.Robot Loader.....	33
7.Programmazione e Loader.....	35
7.1.Software RobotLoader	36
7.2.Collegamento dell'interfaccia USB – Windows	37
7.3.Collegamento dell'interfaccia USB – Linux	40
7.4.Fine dell'installazione del Software	40
7.5.Test interfaccia USB e avvio di RobotLoader.....	41
7.6.Aprire una porta seriale – Linux	42
7.7.SELFTEST	43
7.8.Calibrazione	45
7.9.Posizione dopo la calibrazione.....	46

7.10. Test della pulsantiera	47
8. Software RACS	48
8.1. Manuale di istruzione di RACS.....	49
8.2. RACS - Collegamenti	50
8.3. RACS – Controllo Automatico di Posizione	52
9. Programmazione di Robot Arm.....	54
9.1. Impostazioni dell'editor di testo	54
9.2. Apri e compila un esempio di progetto.	56
10. Per concludere	61

Indice figure

Fig.1 - Braccio robotico RA PRO	9
Fig.2 - Assemblaggio completo	16
Fig.3 - Modulo RP6v2-M256-WIFI	18
Fig.4 - ARX APC220	18
Fig.5 - ARX-BT03	18
Fig.6 - Circuito stampato	19
Fig.7 - Circuito stampato	20
Fig.8 - Jumper ISP/BOOT.....	20
Fig.10 - Adattatore di programmazione USB.....	35
Fig.9 - Software RobotLoader	35
Fig.11 - RobotLoader	36
Fig.12 - Schermo terminale.....	36
Fig.13 - Selezione lingua	41
Fig.14 - Selezione porta seriale	41
Fig.15 - jumper ISP/BOOT	44
Fig.16 - Finestra terminale.....	44
Fig.17 - Selezione calibrazione	45
Fig.18 - Finestra di calibrazione	45
Fig.19 - Braccio robotico calibrato	46
Fig.21 - Circuito stampato Robot	47
Fig.20 - Cavo della pulsantiera	47
Fig.22 - Circuito stampato pulsantiera	47
Fig.23 - Finestra interfaccia utente di RobotLoader.....	48
Fig.24 - Finestra programma RACS	50
Fig.25 - Interfacce seriali disponibili	50
Fig.28 - Abilitazione alimentazione Servo.....	51
Fig.27 - Abilitazione connessione seriale	51
Fig.26 - Selezione interfaccia seriale	51
Fig.29 - Errore di collegamento seriale	51
Fig.30 - Finestra controlli.....	52
Fig.32 - Impostazioni di RACS	53
Fig.31 - Modulo wireless ARX-APC-220	53
Fig.33 - Modulo Bluetooth ARX-BT03	53
Fig.34 - Simbolo Bluetooth	53
Fig.35 - Progetto PN2.....	55
Fig.36 - Aprire un progetto con PN2	56
Fig.37 - Funzione "Make All"	57

Appendice

A. SCHEMA ELETTRICO DI ROBOT ARM RA1-PRO	63
B. SCHEMA ELETTRICO DELL'ALIMENTAZIONE DI RA1-PRO	64
C. SCHEMA ELETTRICO DEI CONNETTORI RA1-PRO.....	65
D. SCHEMA CONTROLLO CORRENTE SERVO V3	66
E. SCHEMA ELETTRICO TASTIERA RA1-PRO.....	67
F. CIRCUITO STAMPATO.....	68

AREXX e ROBOT ARM sono marchi registrati di AREXX Engineering - HOLLAND.

© Traduzione italiana (Giugno 2015): AREXX Engineering (NL).

Questo manuale è protetto dalle leggi del Copyright. Qualsiasi parziale o completa riproduzione del contenuto è vietata senza previa autorizzazione scritta dell'importatore Europeo:

AREXX Engineering - Zwolle (NL).

Costruttore e distributore non possono essere ritenuti responsabili per qualsiasi danno dovuto a maltrattamento, errori di montaggio e/o mancanza di rispetto delle istruzioni contenute nel manuale.

Soggetto a modifiche senza preavviso.



Costruttore:
AREXX Engineering
ZWOLLE The Netherlands

Supporto tecnico di montaggio del robot:

WWW.AREXX.COM
WWW.ROBOTERNETZ.DE

© AREXX Holland and DAGU China

© Traduzione italiana: AREXX - The Netherlands

Diritti legali

©2007 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

Questo manuale è protetto dai diritti del Copyright. E' proibito copiare tutto o parte del contenuto senza previa autorizzazione scritta!

Le specifiche del prodotto e gli accessori forniti sono soggetti a modifiche.

Il manuale è soggetto a modifiche senza preavviso.

Puoi trovare le versioni più recenti di questo manuale in modo gratuito su:

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

"Robot Arm PRO e "Hobby" sono marchi registrati da AREXX Engineering. Tutti gli altri marchi sono di proprietà dei rispettivi proprietari. Noi non siamo responsabili dei contenuti di pagine web esterne che sono menzionate in questo manuale.

Note informative riguardanti la limitazione di garanzia e responsabilità

La garanzia concessa da AREXX Engineering è limitata alla sostituzione o riparazione del Robot Arm e dei suoi accessori entro il periodo legale di garanzia se il difetto è dovuto a errori di produzione, come danni meccanici o mancanti o errato assemblaggio di componenti elettronici eccetto tutti quei componenti che sono collegati mediante spine o prese.

La garanzia non si applica direttamente o indirettamente ai danni dovuti all'uso del robot.

Questo esclude i diritti che rientrano nella prescrizione legale di responsabilità del prodotto.

La garanzia non si applica in caso di modifiche irreversibili (come la saldatura di altri componenti, la perforazione di fori, etc.) del Robot Arm o i suoi accessori o se Robot Arm è stato danneggiato a causa della mancanza di rispetto di questo manuale!

La garanzia non si applica in caso di mancanza di rispetto di questo manuale! Inoltre, AREXX Engineering non è responsabile per danni di qualsiasi tipo derivanti dalla mancanza di rispetto di questo manuale! Si prega di rispettare soprattutto le "Raccomandazioni di sicurezza" nel manuale di Robot Arm.

Si prega di prendere conoscenza dei contratti di licenza che si trovano nel CD-ROM!

IMPORTANTE

Prima di utilizzare questo Robot Arm per la prima volta, leggere attentamente il presente manuale fino alla fine! Il manuale spiega l'uso corretto e ti informa riguardo a potenziali pericoli! Inoltre essi contengono informazioni importanti che potrebbero non essere ovvie per tutti gli utenti.

Importanti raccomandazioni di sicurezza

Questo modulo è dotato di componenti altamente sensibili. I componenti elettronici sono molto sensibili alle scariche elettrostatiche. Prendere il modulo solo per i bordi ed evitare il contatto diretto con i componenti sopra al circuito.

Si prega di non sovraccaricare i servo.

Simboli

Questo manuale fornisce i seguenti simboli:



Il simbolo "Attention!" è usato per attirare l'attenzione su dettagli importanti. La mancanza di rispetto di queste precauzioni può danneggiare o distruggere il robot e/o componenti aggiuntivi e inoltre puoi mettere a rischio la tua salute e quella di altre persone!



Il simbolo "Information" è usato per attirare l'attenzione su utili suggerimenti o informazioni di fondo. In questo caso l'informazione è da considerarsi "utile ma non necessaria".

Raccomadazioni di sicurezza

- Verifica la polarità delle batterie o dell'alimentazione.
- Tieni tutti i componenti asciutti, quando il prodotto si bagna rimuovi immediatamente le batterie o l'alimentazione.
- Rimuovi le batterie o l'alimentazione se non utilizzi il prodotto per lungo tempo.
- Prima della messa in servizio del modulo accertati che i cavi non siano danneggiati.
- Se pensi che il dispositivo non operi in sicurezza scollegalo immediatamente e accertati che non possa essere utilizzato in modo casuale.
- Consulta un esperto se non sei sicuro sul funzionamento, sicurezza o collegamenti del modulo.
- Non usare il modulo in stanze o in condizioni sfavorevoli.
- Non sovraccaricare i servo.
- Questo modulo è costruito con componenti molto sensibili. Le parti elettroniche sono molto sensibili alle scariche elettrostatiche. Prendere il modulo solo per i bordi ed evitare il contatto diretto con i componenti sopra alla scheda elettronica.

Uso normale

Questo prodotto è stato sviluppato per tutte le persone interessate alla robotica. Lo scopo principale è imparare a come puoi programmare il dispositivo in C. Questo prodotto non è un gioco; esso non è adatto a bambini al di sotto dei 14 anni! Questo braccio robotizzato non è un robot industriale, con specifiche e prestazioni industriali.

Il robot può essere usato solo in ambienti chiusi Il prodotto non deve essere umido o bagnato. Porre attenzione quando lo si passa da un ambiente freddo ad uno caldo, prima di usarlo attendere un pò di tempo in modo tale che si adatti alle nuove condizioni ambientali.

Qualsiasi uso al di fuori da quello sopra descritto può danneggiare il prodotto ed aggiungere nuovi rischi come cortocircuiti, incendi, scosse elettriche, ecc...

Per favore leggi le istruzioni di sicurezza di questo manuale.

1. Descrizione di ROBOT ARM

Il grande ROBOT ARM di metallo è adatto per le scuole e progetti educativi per imparare le basi dell'elettronica, della meccanica e della programmazione.

ROBOT ARM è controllato da un potente microcontrollore ATMEGA64 il quale è programmabile in linguaggio C tramite strumenti Open Source.

L'utente può caricare i propri programmi in modo semplice e facile tramite un'interfaccia USB ed un programma di Uploader.

Gli ingressi e le uscite (I/O), insieme ad un flessibile sistema I2C bus, consentono di aggiungere ulteriori moduli in grado di interagire con l'ambiente del robot.

Contenuto dell'imbballaggio:

- 1) Kit completo di costruzione di Robot Arm (meccanica ed elettronica)
- 2) Interfaccia USB con cavo
- 3) CD-ROM contenente manuali e software

1.1. Caratteristiche:

- 1) Processore ATMEGA64
- 2) Diversi ingressi ed uscite I/O disponibili
- 3) Bus I2C
- 4) 6 servomotori (4 pezzi S06NF 2 pezzi S05NF)
- 5) 100% di metallo
- 6) Lunghezza del braccio: 390 mm
- 7) Altezza: 460 mm
- 8) Base diametro: 210 mm
- 9) Alimentazione: 9-14V/3-4A



Attenzione

- * I diritti di reso non si applicano dopo aver aperto le buste di plastica contenenti parti e componenti.
- * Leggere il manuale prima di assemblare l'unità.
- * Poni attenzione quando usi gli attrezzi.
- * Non assemblare il robot in presenza di bambini piccoli. Loro possono farsi del male con gli attrezzi o ingerire piccoli componenti e parti.
- * Fai attenzione al corretto collegamento della batteria.
- * Assicurati che le batterie ed il dispositivo rimangano sempre asciutti. Se il ROBOT ARM dovesse bagnarsi rimuovere immediatamente le batterie ed asciugare quanto prima le parti bagnate.
- * Rimuovere le batterie se ROBOT ARM non verrà utilizzato per più di una settimana.

1.2. Che cosa possiamo fare con Robot Arm?

- Trasferire esempi e nuovi programmi in Robot Arm.
- Controllare il Robot Arm tramite tastiera.
- Controllare e programmare il Robot Arm tramite il software RACS o software Android.
- Espandere il Robot Arm con moduli di estensione pronti all'uso in modo che lui possa ascoltare, sentire e vedere in modo da reagire nel suo ambiente.
- Come i veri robots possono costruire, ad esempio, macchine, questo robot può eseguire alcuni compiti per voi.
- Il Robot Arm può comunicare con il suo ambiente e molte altre unità tramite la sua interfaccia I2C.
- Intelligenza artificiale: il Robot Arm migliora automaticamente il suo software tramite il suo software di autoapprendimento.

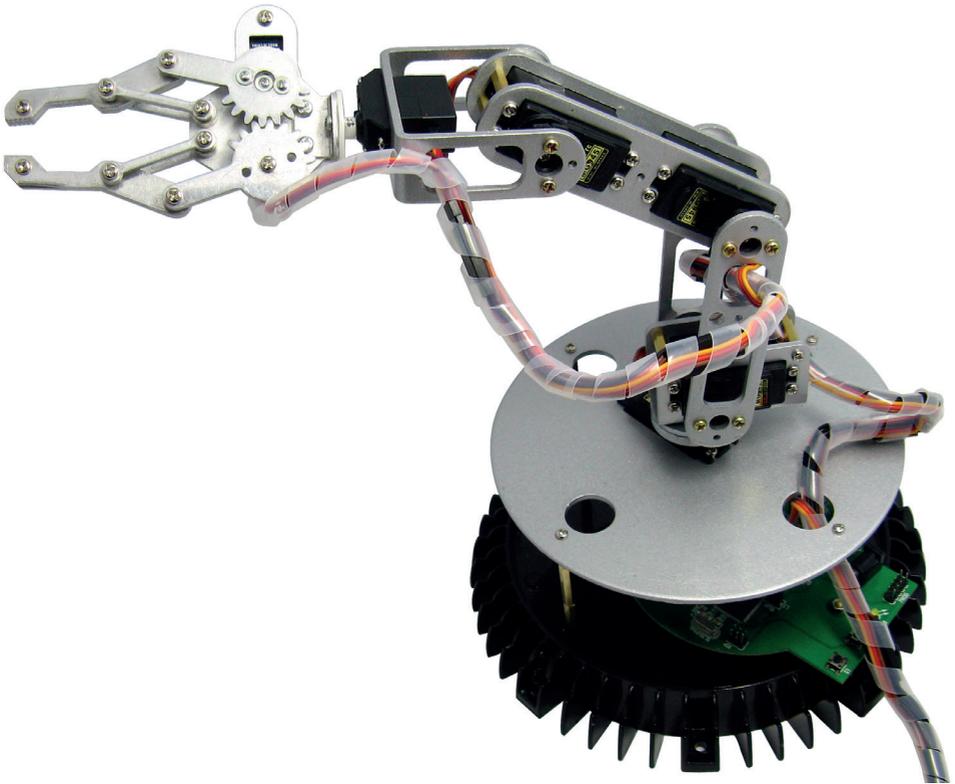
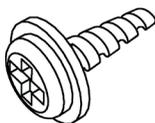


Fig.1 - Braccio robotico RA PRO

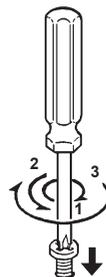
2. Attrezzi richiesti per le modifiche



Viti autofilettanti (Parker)



Le viti autofilettanti si comportano come le viti da legno, per esempio esse, durante la rotazione, realizzano un filetto nel materiale che funziona come un dado. A tal fine, questo tipo di vite ha un filetto più grande ed una punta più tagliente di una vite normale.



Le viti autofilettanti hanno un intaglio in alto che facilita la perforazione del materiale. Il modo migliore per stringere questo tipo di vite è:

- 1 Inserire la vite nel materiale
- 2 Allentare leggermente la vite
- 3 Stringere nuovamente a vite

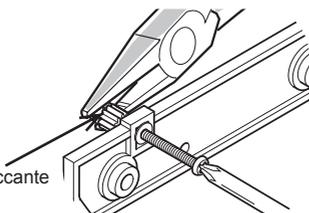
Se le viti si allentano e si stringono troppo spesso volte il foro si allarga gradualmente e la vite non tiene più.

Dado autobloccante

Fissaggio di un dado autobloccante



Dado autobloccante

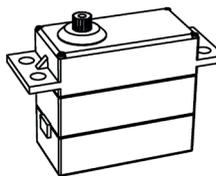


Non forzare le viti altrimenti la plastica può rompersi.

RA1-PRO E' GIA' COMPLETAMENTE ASSEMBLATO

3. Elenco parti

Servomotore



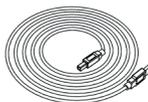
O 6 pezzi.

CD



O 1 pezzo.

Cavo USB



O 1 pezzo.

Tastiera



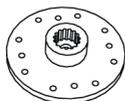
O 1 pezzo.

Disco assiale di metallo



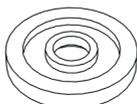
O 3 pezzi.

Disco di plastica per servo



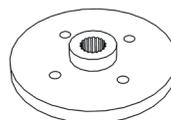
O 1 pezzo.

Disco di metallo per servo Big A



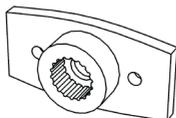
O 1 pezzo.

Disco di metallo per servo Big B



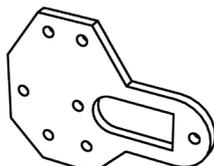
O 1 pezzo.

Leva per servo grande



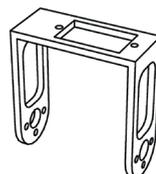
O 3 pezzi.

Supporto per servo



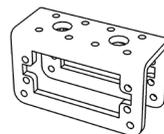
O 1 pezzo.

Supporto braccio



O 1 pezzo.

Squadretta di montaggio



O 1 pezzo.

Distanziale M3x6



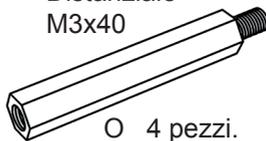
O 4 pezzi.

Distanziale M3x16



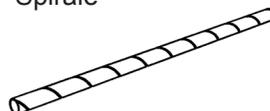
O 2 pezzi..

Distanziale M3x40



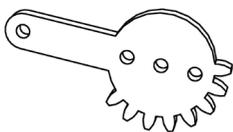
O 4 pezzi.

Spirale



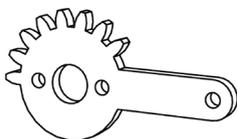
O 1 pezzo.

Dito parte A



○ 1 pezzo.

Dito parte B



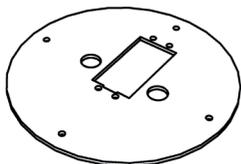
○ 1 pezzo.

Polpastrello



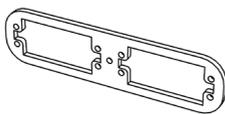
○ 4 pezzi.

Piastra Servo



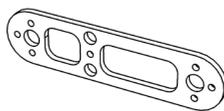
○ 1 pezzo.

Asta di accoppiamento Servo M3 - M4



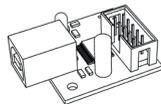
○ 2 pezzi.

Asta di accoppiamento Servo M2 - M3



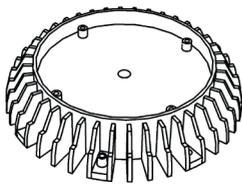
○ 2 pezzi.

Interfaccia di programmazione



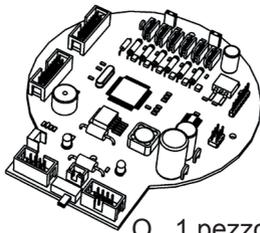
○ 1 pezzo.

Base di Robot Arm



○ 1 pezzo.

Circuito stampato.



○ 1 pezzo.

Cavo di programmazione



○ 1 pezzo.

Prolunga per Servo



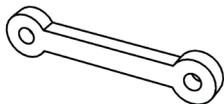
○ 1 pezzo.

Cavo per tastiera



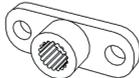
○ 1 pezzo.

Asta di accoppiamento



○ 5 pezzi.

Leva per Servo piccoli



○ 1 pezzo.

Vite testa tonda M3x20



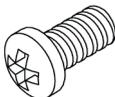
○ 9 pezzi.

Vite autofilettante M3.2x6



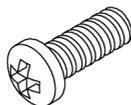
○ 4 pezzi.

Vite test tonda M3x6



○ 2 pezzi.

Vite testa tonda M3x8



○ 42 pezzi.

Vite testa tonda M3x12



○ 9 pezzi.

Dado M3



○ 24 pezzi.

Dado autobloccante



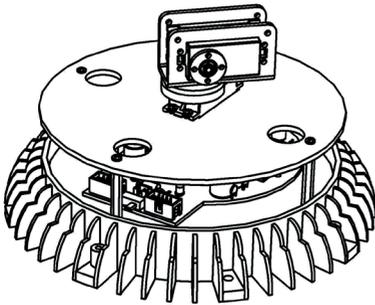
○ 3 pezzi.

4. Costruzione parti meccaniche

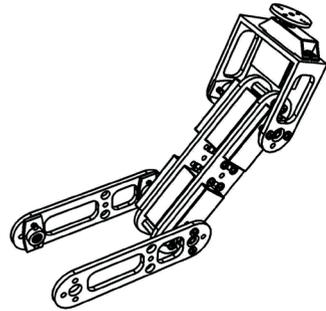
RA1-PRO E' GIA' COMPLETAMENTE ASSEMBLATO

4.1. Costruzione delle principali parti meccaniche

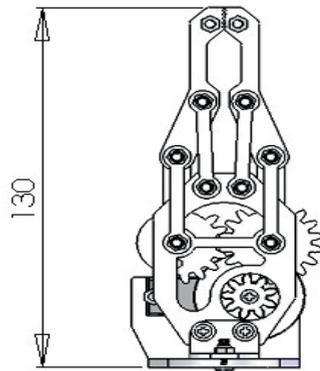
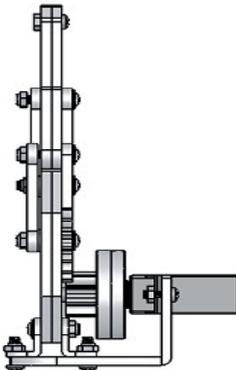
Piastra inferiore



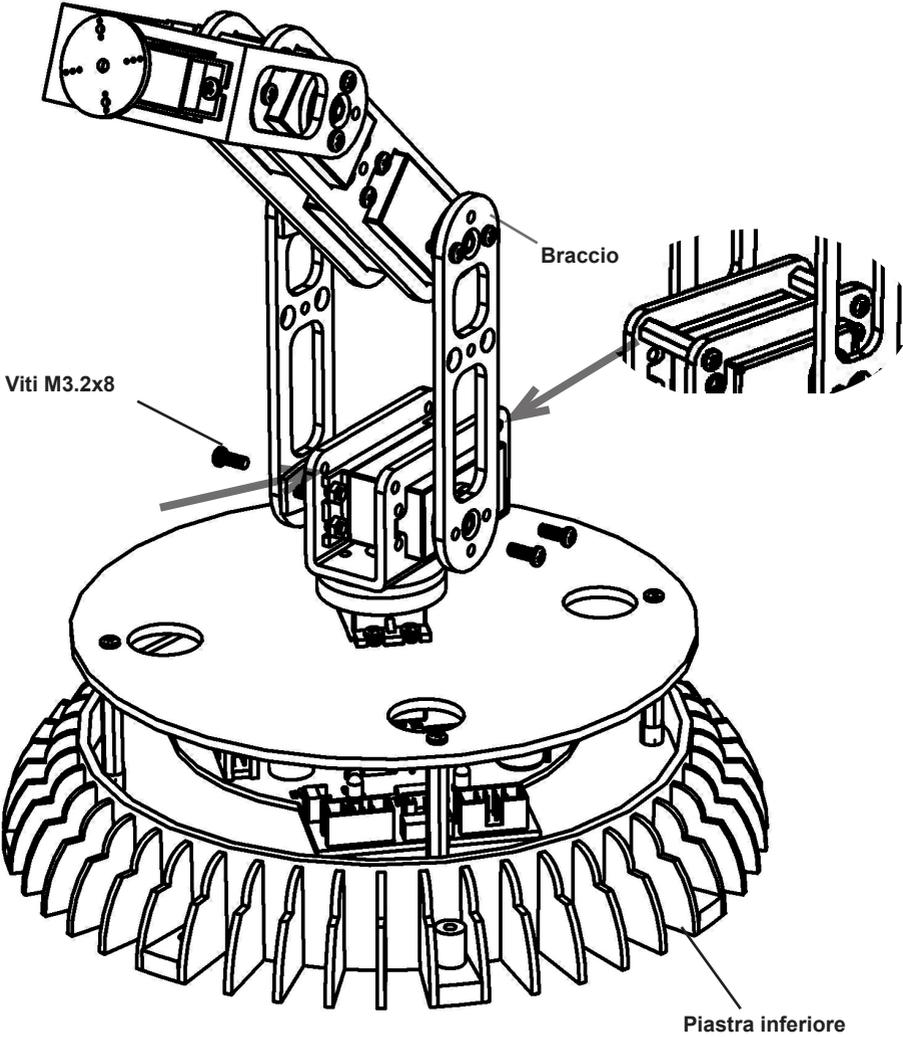
Braccio



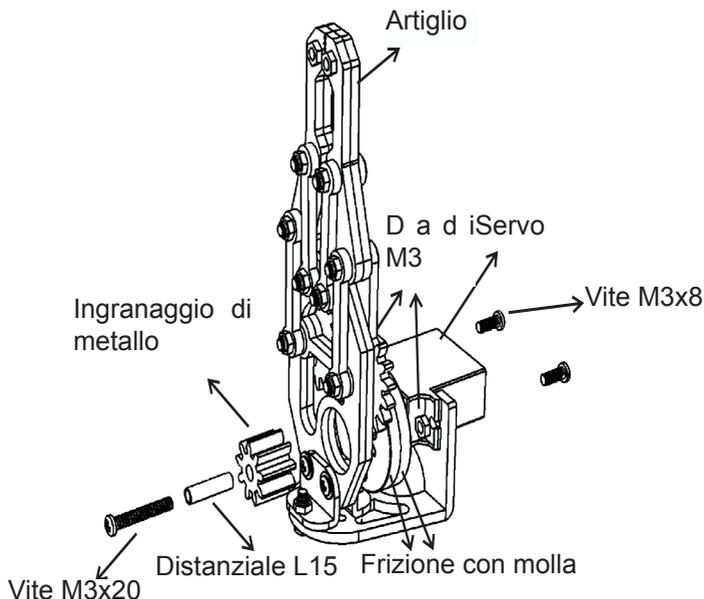
Dito con frizione



4.2. Costruzione del braccio:



4.3. Assemblaggio ed allineamento della pinza



CONSIGLI ED ALLINEAMENTO DELLA COPPIA FRIZIONE

Se la frizione non è propriamente ingrassata ed allineata, questo può causare la rottura del Servo "DITO"!

1. Se la molla nella frizione non si muove facilmente occorre ingrassare leggermente le parti della frizione.
2. Se la molla dell'ingranaggio è troppo stretta, essa blocca tutta la frizione ed il Servo della Falange non si muove.
Per piacere aggiungere un distanziale o allentare la vite.
3. Se la frizione e l'ingranaggio non sono perfettamente allineati occorre cercare di allinearli un pò meglio in modo che il dito completo sia correttamente allineato.

4.4. ASSEMBLAGGIO COMPLETO

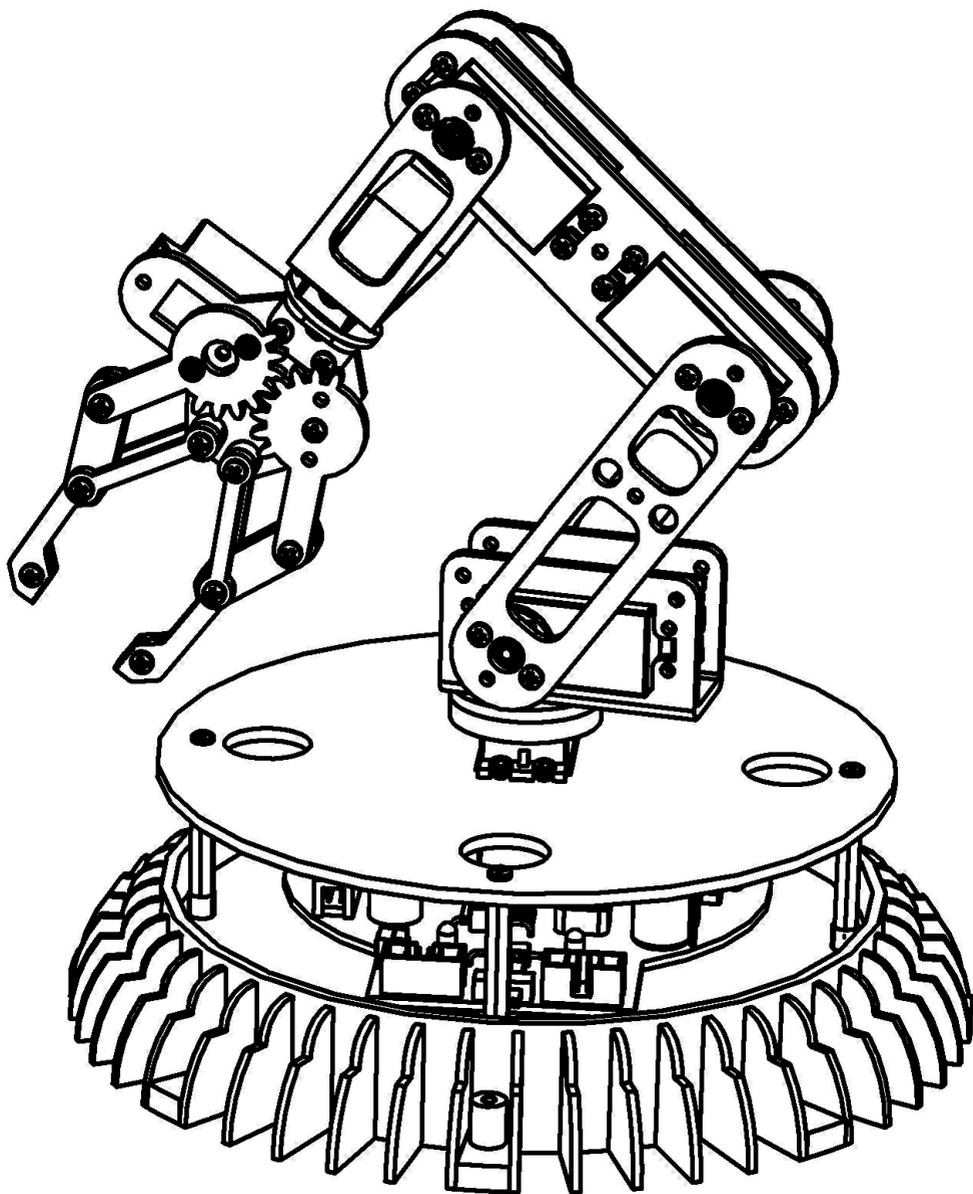


Fig.2 - Assemblaggio completo

5. ELETTRONICA

5.1. Caratteristiche e dati tecnici

Questa sezione fornisce una panoramica delle funzioni del braccio del robot e l'introduzione di alcune parole chiave di base, per farvi conoscere la terminologia utilizzata in questo manuale. La maggior parte di queste parole chiave saranno spiegate nei capitoli successivi. Caratteristiche, componenti e dati tecnici del braccio di ROBOT ARM:

1. Potente microcontrollore Atmel ATMEGA64 8-Bit

- ◇ Velocità 16 MIPS (=16 Milioni di istruzioni al secondo) con frequenza di clock di 16MHz.
- ◇ Memoria: 64KB Flash ROM, 4KB SRAM, 2KB EEPROM.
- ◇ Liberamente programmabile in C (usando WinAVR / avr-gcc)!

2. Sistema di espansione flessibile, basato su I²C-Bus

- ◇ Sono richiesti solo due segnali (TWI -> "Two Wire Interface").
- ◇ Velocità di trasferimento fino a 400kBit/s.
- ◇ Architettura Master->Slave.
- ◇ Possibilità di collegare contemporaneamente al bus fino a 127 Slaves.
- ◇ Bus-system molto conosciuto. Il mercato offre un grande quantità di circuiti integrati standard, sensori e componenti, che possono essere collegati direttamente.

3. Possibilità di montaggio di moduli wireless

- ◇ RP6 WIFI
- ◇ Bluetooth
- ◇ APC-220

4. Interfaccia USB PC per programmare il microcontrollore con PC.

- ◇ Collegamento via cavo per la massima velocità. Il programma di Upload normalmente gira a 500kBaud, riempiendo lo spazio totale della memoria libera (30KB, 2KB sono riservati per il Bootloader) in pochi secondi.
- ◇ L'interfaccia può essere utilizzata per programmare tutti i moduli di espansione disponibili per ROBOT ARM con microcontrollore AVR (modulo RP6-256-WIFI).
- ◇ Essa può essere utilizzata per comunicare tra il robot ed i moduli di espansione. Per esempio, tu puoi usare questo per scopi di debugging trasferendo dati di misura, messaggi di testo, e altri dati al PC.
- ◇ Il driver dell'interfaccia mette a disposizione una porta seriale virtuale (VCP) per tutti i Sistemi Operativi più comuni inclusi Windows 2K/XP/Vista e Linux. La VCP può essere usata in programmi terminali standard e software proprietario

5.2. Modifiche relative al circuito stampato V3 tipo RA1 PRO e RA2-HOBBY

- ◇ Nuovo DC/DC converter, 3500mA, TI TPS54332.
- ◇ Nuovo regolatore di tensione tipo LDO per alimentazione a 5V del microcontrollore.
- ◇ Connettori aggiuntivi di alimentazione per VCC e alimentazione Servo.
- ◇ 6 preamplificatori per la misura di assorbimento di tutti i Servo tramite resistenza di Shunt.
- ◇ Il pulsante di Reset è ora un pulsante di start/stop del programma.
- ◇ Autostart (= il programma parte automaticamente e rapidamente all'accensione). Se lo preferisci può essere configurato tramite il "robotloader"
- ◇ Connettori supplementari di espansione con tutti i pin di I/O liberi provenienti dall' ATMEGA64.
- ◇ Collegamento I2C Bus, compatibile con i moduli di espansione RP6 XBUS, per esempio il modulo RP6-M256 WLAN.
- ◇ Zoccolo per APC220 o modulo Bluetooth.
- ◇ 4 Led di stato di colore blue invece di una coppia di led verde/rossa.
- ◇ Fusibile da 4A .
- ◇ Cristallo da 16.000 MHz invece che 16.384MHz.
- ◇ Le resistenze di pull-up per il Bus I2C ora si trovano sul Circuito Stampato.
- ◇ Il pulsante On/off separa la Logica. L'alimentazione del Servo proveniente dal convertitore DC/DC è controllata dal microprocessore.
- ◇ Alimentazione 9 - 14 Volt (il valore massimo è 18 Volt).

5.2.1. ESTENSIONI PER ROBOT ARM

- ◇ Modulo RP6v2-M256-WIFI per controllare senza fili il braccio robotico all'interno di una rete.
- ◇ ARX-APC220 per controllo wireless RACS 433 MHz.
- ◇ ARX-BT03 per controllo tramite sistemi bluetooth ANDROID.

Vedi anche i manuali di AREXX APC-220, Android, e moduli WIFI.



Fig.3 - Modulo RP6v2-M256-WIFI



**Fig.4 -
ARX APC220**

Fig.5 - ARX-BT03

5.3. Assegnazione dei connettori sul PCB

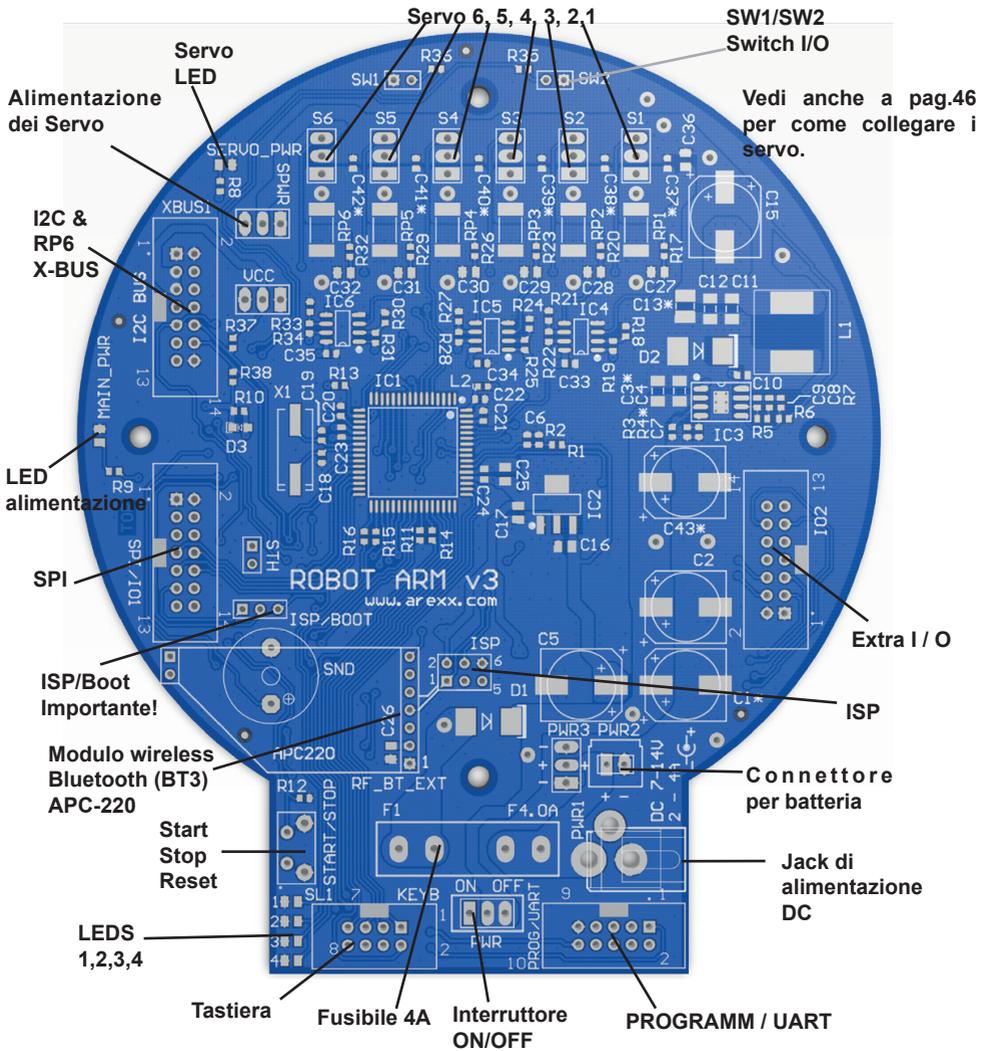


Fig.4 - Circuito stampato

5.4. LEDS

Sul braccio robotico ci sono alcuni LEDs, qui di seguito una breve spiegazione.

- ◇ Il LED "MAIN_PWR" di colore giallo segnala se c'è un'alimentazione esterna e se il robot è acceso.
- ◇ Il LED "SERVO_PWR" di colore rosso si illumina solo quando il convertitore DC/DC per l'alimentazione dei servo è acceso dal microcontrollore. Questa funzione è controllata dal software.
- ◇ Ci sono 4 led di stato LEDs "SL1 - 4" di colore blue che sono controllati direttamente dal microprocessore. Tu puoi usare liberamente questi LEDs all'interno della tua applicazione.

5.5. Avviare il Robot

- 1) Come prima cosa verifica tutto l'assemblaggio meccanico ed elettronico del braccio robotico quindi collega i servo (vedi pag.19 e pag.46).
- 2) Collega l'alimentazione 9 -14V (18 V = Valore massimo).
- 3) Accendi il robot tramite l'interruttore principale On/Off.

5.5.1. Alimentazione

1) Connettore principale

Per alimentare il robot ci sono due opzioni. La soluzione più semplice è quella di collegare un alimentatore con uscita a 9-14V / 3-4 Amps all'ingresso principale DC. In questo modo la tensione è collegata direttamente all'ingresso del regolatore di tensione.

2) Batterie

La seconda soluzione è quella di collegare una batteria al terminale "battery" (9-14V). Se la tensione scende al di sotto di $< 6.7 V$, viene visualizzato un messaggio di attenzione.

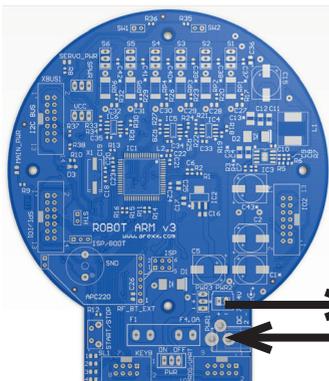


Fig.5 - Circuito stampato

Importante!

Il Jumper "ISP/BOOT", durante l'uso normale, dovrebbe essere tenuto nella posizione di sinistra. vedi Fig.8.



Fig.6 - Jumper ISP/BOOT

Connettore per
batteria
9 to 14 Volt
Terminale DC
9 - 14 Volt

Non appena il Robot Arm viene collegato all'alimentazione i servo si muovono leggermente ed il LED giallo (Power LED) inizia a lampeggiare.

A differenza di quanto poteva sembrare la partenza non è stata difficile, tuttavia il lavoro non è ancora finito .

La vera difficoltà inizia ora...!

Ma... come prima cosa diamo una letta a pag.22 dove si descrive l'installazione del software.

6. Installazione del software

Eseguiamo ora le operazioni di installazione del software.

Un software installato bene è di fondamentale importanza per i capitoli a venire.

Dal momento che occorrono i diritti amministrativi, devi loggarti al sistema come amministratore.

Si consiglia di leggere completamente l'intero capitolo, quindi partire con l'installazione passo per passo.

L'utente deve avere una conoscenza di base dei computer basati su Windows o Linux ed avere familiarità con i programmi attuali così come gestione di files, web browsers, editors di testi, software di compressione file (WinZip, WinRAR, unzip e altri) ed eventualmente la shell di Linux, etc.!

Se la tua conoscenza del computer è molto limitata, dovresti imparare di più sui sistemi prima di utilizzare Robot Arm.

Questo manuale non è inteso come un'introduzione ai computers, occorrerebbe troppo tempo! Esso è finalizzato solo a Robot Arm, alla sua programmazione ed al software specifico richiesto.

6.1. Il CD-ROM di Robot Arm

Probabilmente avrai già inserito il CD-ROM nel lettore del tuo computer, se così non fosse, fallo ora!

In Windows, il menu del CD dovrebbe apparire subito dopo tramite avvio automatico, altrimenti tu puoi aprire il file "start.htm" utilizzando un browser web come, ad esempio, Firefox che lo puoi trovare nella cartella principale del CD usando il file manager.

Nel caso in cui non fosse mai stato installato un browser aggiornato (dovrebbe essere almeno Firefox 1.x o Internet Explorer 6 ...) si possono trovare i files di installazione di Firefox nella cartella del CD:

```
<CD-ROM drive>:\Software\Firefox
```

Dopo aver scelto la lingua nel menù di questo CD, oltre a questo manuale (che tu puoi scaricare anche dalla nostra home page), troverai informazioni, data sheets ed immagini, come pure il menu "Software", il quale contiene tutti gli strumenti software, i drivers USB ed i programmi d'esempio con relativo codice sorgente per il Robot Arm.

In funzioni delle impostazioni di sicurezza del browser, tu puoi iniziare l'installazione del programma direttamente dal CD!

Se le impostazioni di sicurezza del tuo browser non consentono l'installazione diretta da CD-ROM, tu devi prima copiare i files in una cartella del tuo hard disc ed iniziare l'installazione da qui. Per ulteriori dettagli, per piacere, fai riferimento alla pagina software nel menù del CD. Eventualmente tu puoi anche, tramite un file manager, esplorare il CD ed installare il software dallo stesso CD. I nomi delle cartelle sono autospieganti così che tu puoi accedere facilmente ai pacchetti software corrispondenti ed ai sistemi operativi.

6.2. WinAVR - per Windows

Partiamo con l'installazione di WinAVR che, come dice il nome, **è disponibile solo per Windows!**

Gli utenti Linux possono tralasciare la prossima sezione.

WinAVR (pronunciato come la parola "whenever") è una collezione di molti programmi di utilità necessari allo sviluppo di software per microcontrollori AVR utilizzando il linguaggio C.

Oltre a GCC per AVR (conosciuto con il termine "AVR-GCC", maggiori informazioni più avanti) WinAVR include anche il comodo editor "Programmers Notepad 2" che noi useremo durante lo sviluppo di programmi per Robot Arm.

WinAVR è un progetto privato che non è supportato da una ditta.

Questo software è disponibile in internet in modo gratuito. Tu puoi trovare gli aggiornamenti delle versioni e maggiori informazioni a:

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

Nel frattempo il progetto è ufficialmente supportato da ATMEL

AVRGCC è disponibile per AVRStudio, l'ambiente di sviluppo AVR da ATMEL. Tuttavia noi non lo descriveremo in questo manuale in quanto per il nostro uso è più adatto Programmers Notepad.

L'installazione di WinAVR si trova nella cartella del CD in:

<CD-ROM drive>:\Software\AVR-GCC\Windows\WinAVR\

L'installazione di WinAVR è semplice ed autospiegante.

Normalmente non occorre cambiare nessun settaggio così si può cliccare direttamente su

"Continue"!

Se usi Windows Vista o Windows 7 devi installare l'ultima versione di WinAVR! Questa dovrebbe funzionare perfettamente anche con Windows 2K e XP. Se così non fosse prova una versione vecchia che si trova sul CD (prima di eseguire una nuova installazione WinAVR devi disinstallare la versione esistente!). Ufficialmente Win x64 non è supportato ma, se dovessero esserci problemi, il CD contiene una patch per sistemi Win x64. Tu troverai maggiori dettagli nella pagina software del menù del CD.

6.3. AVR-GCC, avr-libc e avr-binutils - per Linux

(Gli utenti Windows possono tralasciare questa sezione!)

Linux potrebbe richiedere maggior impegno. Alcune distribuzioni contengono già i pacchetti necessari ma possono essere versioni obsolete, per questo è necessario compilare ed installare versioni più recenti. E' impossibile descrivere in dettaglio le numerose distribuzioni Linux come SuSE, Ubuntu, RedHat/Fedora, Debian, Gentoo, Slackware, Mandriva ecc... che esistono in molte versioni con le loro particolarità e noi qui ci limiteremo solo alle linee generali.

Lo stesso vale per tutte le altre sezioni Linux di questo capitolo!

La procedura qui descritta potrebbe non essere necessariamente funzionante per te.

Spesso è utile cercare in internet, per esempio,

`"<LinuxDistribution> avr gcc"`

o qualcosa di simile, (prova con diversi linguaggi).

La stessa cosa vale per tutte le altre sezioni Linux, naturalmente con parole di ricerca adatte! Se incontri problemi con l'installazione di AVR-GCC, puoi sempre dare uno sguardo al nostro forum riguardante il robot oppure in uno dei numerosi forums di Linux.

Come prima cosa devi disinstallare le versioni già installate di avr-gcc, le avr-binutils e le avr-libc poichè, come detto, queste sono parecchio obsolete.

Puoi farlo tramite il gestore dei pacchetti della tua distribuzione ricercando "avr" start up e disinstallando i tre pacchetti sopra menzionati, nella misura in cui esistono nel tuo computer.

Puoi scoprire facilmente se avr-gcc è già installato tramite la riga di comando:

```
> which avr-gcc
```

Se viene visualizzato un percorso allora una versione è già installata
Per capire qual'è la versione installata scrivi:

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

e guarda l'uscita. Se la versione visualizzata è precedente alla 3.4.6, tu devi disinstallarla in quanto è una versione obsoleta.

Se il numero di versione è tra la 3.4.6 e la 4.1.0, tu puoi provare a compilare i programmi (vedi il capitolo seguente). Se la compilazione fallisce allora devi installare i nuovi strumenti. In seguito installeremo la versione 4.1.1 (uscita Marzo 2007) insieme ad alcune importanti patches.

Se i pacchetti di cui sopra non vengono visualizzati nel gestore dei pacchetti, anche se sicuramente un avr-gcc è già installato, occorre che tu cancelli manualmente i files binari di rilievo.

Devi, per esempio, cercare in tutte le cartelle /bin, /usr/bin ecc... i files che iniziano con "avr" e cancellarli (naturalmente SOLO questi files e nessun altro!).

Eventualmente le cartelle esistenti come /usr/avr o /usr/local/avr possono essere cancellate.

Importante: devi essere sicuro che i normali strumenti di sviluppo di Linux come GCC, make, binutils, libc, ecc... siano installati prima di compilare ed installare qualsiasi programma! Il modo migliore è farlo utilizzando il gestore pacchetti della tua distribuzione. Qualsiasi distribuzione Linux dovrebbe essere fornita con i pacchetti richiesti, questi dovrebbero trovarsi sul CD di installazione oppure essere disponibili in internet.

Assicurati che il programma "textinfo" sia installato, in caso contrario devi installare il pacchetto in questione altrimenti non funziona!

Dopo che hai fatto quanto sopra puoi partire con l'installazione.

Ora tu hai tre possibilità: o si fa tutto manualmente oppure puoi usare un semplice script di installazione o per ultimo tu puoi usare un pacchetto di installazione tipo .dep già compilato.

Noi suggeriamo di usare per primo uno script di installazione.

Se non dovesse funzionare allora devi installare il compilatore manualmente.

Per scaricare l'ultima versione cerca in:

<http://www.wrightflyer.co.uk/avr-gcc/>

Attenzione: devi avere abbastanza spazio libero sul tuo hard disk! Sono richiesti oltre 400Mb temporanei. Oltre a 300Mb possono essere cancellati dopo l'installazione ma, durante questa, hai bisogno di tutto lo spazio richiesto.

Diversi passi della seguente installazione richiedono i diritti di amministrazione (ROOT RIGHTS), per questo, per piacere, esegui il login con "su" come root o esegui i comandi critici con "sudo" o qualcosa di simile come fai, ad esempio, in Ubuntu (lo script di installazione, mkdir nelle cartelle /usr/local ed il comando "make" richiede i diritti amministrativi).

Per piacere nelle note seguenti fai attenzione all'ESATTA scrittura di tutti i comandi!

Ogni simbolo è importante anche se alcuni comandi sembrano un pò strani, è tutto corretto e non sono errori di battitura!

(<CD-ROM-drive> deve essere sostituito dal percorso dell'unità del CD-ROM!)

La cartella del CD: <CD-ROM drive>:\Software\avr-gcc\Linux contiene tutti i files necessari all'installazione di avr-gcc, avr-libc e binutils.

Come prima cosa tu devi copiare tutti i files di installazione in una cartella del tuo hard disk - questo si applica ad entrambi i metodi di installazione!

Noi useremo la cartella Home (normalmente l'abbreviazione della cartella Home è il simbolo tilde: „~“):

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

Dopo la corretta installazione tu puoi cancellare i files che si trovano nello spazio temporaneo!

6.4. Script per installazione automatica

Dopo aver reso eseguibile lo script tramite il comando "chmod" si può iniziare immediatamente l'installazione:

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

Rispondere "y" alla domanda se si desidera eseguire l'installazione con questa configurazione oppure no.

ATTENZIONE: la compilazione e l'installazione richiedono un po' di tempo che dipende dalla potenza del tuo sistema (per esempio occorrono circa 15 min su un notebook Dual Core a 2 GHz.

Per sistemi più lenti occorre più tempo).

Lo script include anche alcune patches. Queste sono tutti i files tipo .diff presenti nella cartella.

Se l'installazione è andata a buon fine verrà visualizzato il seguente messaggio:

```
(./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
```

Come suggerito puoi eseguire:

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

Questo cancellerà tutti i files temporanei di cui tu non hai più bisogno.

Tu puoi tralasciare il prossimo paragrafo e configurare il percorso per gli strumenti AVR.

Se l'esecuzione dello script dovesse fallire tu devi osservare attentamente i messaggi di errore (se è necessario fai scorrere verso l'alto la console). In molti casi questo è dovuto alla mancanza di programmi che dovrebbero essere stati installati precedentemente (per esempio il file textinfo).

Prima di continuare dopo un errore si raccomanda di cancellare i files appena generati nella cartella standard di installazione "/usr/local/avr", sarebbe preferibile cancellare tutta la cartella.

Se tu non conosci esattamente cosa è sbagliato salva, per piacere, in un file l'uscita di tutta la riga di comando e contatta il supporto tecnico.

Per piacere allega sempre il maggior numero di informazioni, questo può esserti più facilmente di aiuto.

6.5. GCC per AVR

GCC è aggiornato, compilato ed installato come le 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
```

Dopo il carattere “\” premi Enter e continua a scrivere. In questo modo il comando può essere suddiviso su diverse righe, tu però puoi anche tralasciarlo.

6.6. AVR Libc

E ultima, ma non meno importante, la 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
```

Attenzione: nel comando:

```
-build= `./config.guess`
```

accertati di mettere un accento grave `` (à <-- accento grave su a!)
e non un normale apostrofo o virgolette in quanto non funzionerebbe.

6.7. Imposta il Path

Devi accertarti che la cartella

```
/usr/local/avr/bin
```

sia registrata nel path delle variabili di ambiente altrimenti sarà impossibile accedere ad avr-gcc dalla console oppure tramite "Makefile".

Infine devi aggiungere il percorso nel file

```
/etc/profile or /etc/environment
```

o simile (questo varia da una distribuzione all'altra), separato da due punti ":" a partire dalla fine della riga esistente.

Dovrebbe essere simile a questo:

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

Ora scrivi la riga di comando "avr-gcc --version" come descritto precedentemente, se funziona significa che l'installazione è andata a buon fine!

VERSIONI PIU' RECENTI

In internet esistono versioni regolari più recenti, verifica per piacere se ne trovi una più recente rispetto a quella che si trova sul CD.

6.8. Installazione manuale

Se preferisci installare manualmente il compilatore o se l'installazione tramite script non è andata a buon fine allora puoi seguire le seguenti istruzioni.

La descrizione si basa sul presente articolo:

http://www.nongnu.org/avr-libc/user-manual/install_tools.html

che si trova anche nel CD in formato PDF nella documentazione relativa a AVR Libc:

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

La nostra descrizione è molto più corta ma include alcune importanti modifiche senza le quali alcune impostazioni non funzionerebbero bene.

Prima di tutto noi abbiamo creato una cartella nella quale installeremo tutti gli strumenti.

Questa cartella potrebbe essere `/usr/local/avr`.

Per questo entra nella console come ROOT:

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

Può non essere necessariamente questa cartella.

Creiamo quindi la variabile `$PREFIX` per questa cartella:

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

Questo deve essere aggiunto alla variabile d'ambiente `PATH`:

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

VERSIONI PIU' RECENTI

In internet esistono versioni regolari più recenti, verifica per piacere se ne trovi una più recente rispetto a quella che si trova sul CD.

6.9. Binutils per AVR

Ora è necessario decomprimere il codice sorgente delle binutils ed aggiungere alcune modifiche.

Supponiamo che nel nostro esempio tu abbia copiato tutto nella cartella 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
```

Ora esegui lo script di configurazione:

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

Questo script rileva che cosa è disponibile nel tuo sistema e genera gli opportuni makefiles. Ora le binutils possono essere compilate ed installate:

```
> make
> make install
```

In funzione della potenza del tuo computer o tuo sistema quest'operazione può richiedere alcuni minuti.

Questo si applica anche alle prossime due sezioni, specialmente alle GCC!

6.10. Java 6

RobotLoader (vedi le informazioni seguenti) è stato sviluppato per piattaforma Java ed è disponibile per Windows e Linux (teoricamente anche per sistemi operativi tipo OS X ma AREXX Engineering sfortunatamente non è in grado di dare un supporto ufficiale)

Affinchè funzioni occorre installare un aggiornamento di Java Runtime Environment (JRA). Spesse volte JRA è già installato sul computer ma deve essere almeno la versione 1.6 (= Java 6)! Se non hai JRE o JDK installati tu devi installare la JRE 1.6 fornita da SUN Microsystems o alternativamente scaricare una nuova versione dal sito:

<http://www.java.com>

o

<http://java.sun.com>.

6.10.1. Windows

JRE 1.6 per Windows si trova nella seguente cartella:

```
<CD-ROM drive>:\Software\Java\JRE6\Windows\
```

In ambiente Windows l'installazione di Java è molto semplice, tu devi avviare il setup e seguire le istruzioni come da schermo.

Questo è tutto.

Puoi tralasciare il paragrafo seguente.

6.10.2. Linux

In ambiente Linux l'installazione non presenta eccessive difficoltà anche se alcune distribuzioni richiedono un pò di lavoro manuale.

Nella cartella:

```
<CD-ROM drive>:\Software\Java\JRE6\
```

troverai la JRE1.6 così come RPM (SuSE, RedHat etc.) ed anche un archivio autoestraente ".bin". In ambiente Linux è consigliabile cercare i pacchetti Java nel package manager della tua distribuzione (parole come „java“, „sun“, „jre“, „java6“ ...) ed usare i pacchetti della tua distribuzione piuttosto che quella su CD-ROM! Tuttavia accertati di installare Java 6 (=1.6) o una versione più recente ma non una più vecchia!

In ambiente Ubuntu or Debian, l'archivio RPM non funziona direttamente.

Tu devi usare il gestore di pacchetti della tua distribuzione per trovare un pacchetto adatto. RPM dovrebbe funzionare bene con molte altre distribuzioni come RedHat/Fedora e SuSE. Se così non fosse, puoi sempre trovare la soluzione decomprimendo JRE (per esempio in /usr/lib/Java6) dall'archivio autoestraente (.bin) ed impostare manualmente i percorsi per JRE (PATH e JAVA_HOME ecc..).

Per piacere fai riferimento alle istruzioni di installazione da Sun che puoi anche trovare nella sopra menzionata cartella e nel sito di Java (vedi sopra).

Puoi anche verificare se Java è stato installato correttamente dando il comando "java-version" all'interno della console.

L'uscita dovrebbe essere come quella qui di seguito:

```
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)
```

Se l'uscita dovesse essere completamente differente, potresti aver installato una versione errata o sul tuo sistema si trova un'altra Java VM.

6.11. Robot Loader

Robot Loader è stato sviluppato per caricare facilmente nuovi programmi e tutti moduli d'espansione nel Robot Arm (purchè i moduli siano dotati di un bootloader compatibile). Inoltre esso contiene alcune funzioni aggiuntive come, per esempio, un semplice programma terminale.

Non occorre installare RobotLoader, basta infatti copiare il programma in una nuova cartella sul hard disk.

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

Estrarre il programma in una qualche cartella del proprio hard disk, per esempio in una nuova cartella

```
C:\Programme\RobotLoader (or similare).
```

Questa cartella contiene il file: RobotLoader.exe
che può essere avviato con un doppio click..

Robot Loader programma se stesso nell'archivio (JAR) RobotLoader_lib.jar. Alternativamente tu puoi fare partire il programma tramite il comando:

6.11.1. In ambiente Windows:

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

6.11.2. In ambiente Linux:

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

L'opzione -D è necessaria affinché JVM possa trovare tutte le librerie usate.

Dal momento che Windows non lo richiede tu puoi iniziare eseguendo il file .exe.

Linux invece ha bisogno dello script shell "RobotLoader.sh".

Può essere necessario che lo script sia eseguibile:

```
chmod -x ./RobotLoader.sh.
```

In seguito puoi eseguirlo con il comando da console "./RobotLoader.sh".

Si consiglia di creare un collegamento sul desktop o nel menu Start per semplificare l'avvio di RobotLoader.

Per fare questo nell'ambiente Windows occorre premere il pulsante destro del

mouse sul file.exe RobotLoader e quindi selezionare:

“Desktop (Crea collegamento)” nel menu “Invia a”.

6.11.1. Robot Arm Library, Robot Arm CONTROL Library e programmi d’esempio

La Libreria di Robot Arm ed i relativi programmi di esempio si trovano all’interno di un file zip che si trova a sua volta sul CD:

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

Scompatta questo file in una cartella a tua scelta sull’hard disk. E’ consigliabile scompattare i programmi di esempio in una cartella su una partizione dati, ad esempio in una cartella “Robot Arm\Examples\” all’interno della cartella personale “documenti” oppure, se in ambiente Linux, nella cartella Home.

Questo è a tua discrezione.

I singoli programmi d’esempio verranno discussi più tardi nel capitolo: “software”.

7. Programmazione e Loader

Per caricare un programma tipo HEX Robot Arm dal PC all'interno del Robot Arm, noi useremo l'adattatore di programmazione USB e il nostro software RobotLoader.

L'adattatore USB incluso nel pacco deve essere collegato da un lato alla porta USB del computer e dall'altro al connettore Prog/UART presente sul PCB della scheda del Robot Arm.

Quando si carica un nuovo programma nel Robot Arm in automatico viene cancellato quello presente in quel momento.

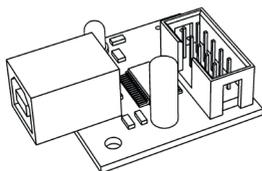


Fig.8 - Adattatore di programmazione USB

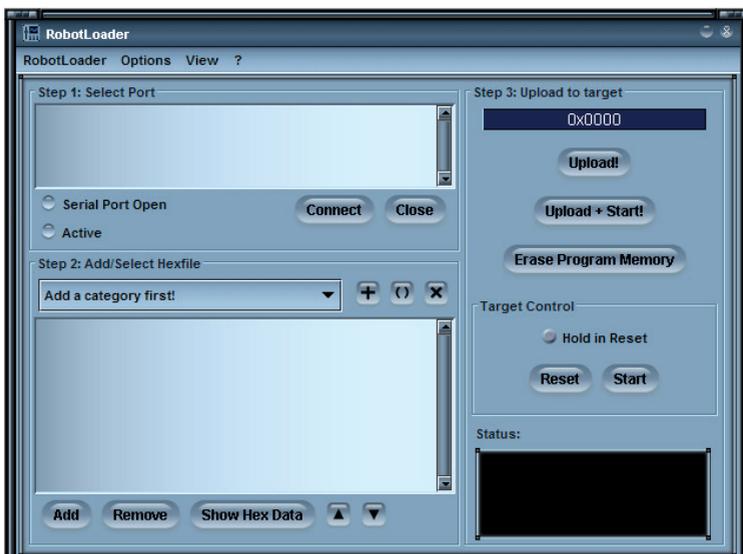


Fig.7 - Software RobotLoader

7.1. Software RobotLoader

Come già detto, il software RobotLoader è stato sviluppato per caricare facilmente i nuovi programmi nel Robot Arm ed in tutti i nostri robots (sempre se contengono un bootloader compatibile).

Se la tensione scende al di sotto di < 6.7 V, viene visualizzato un messaggio di allarme.

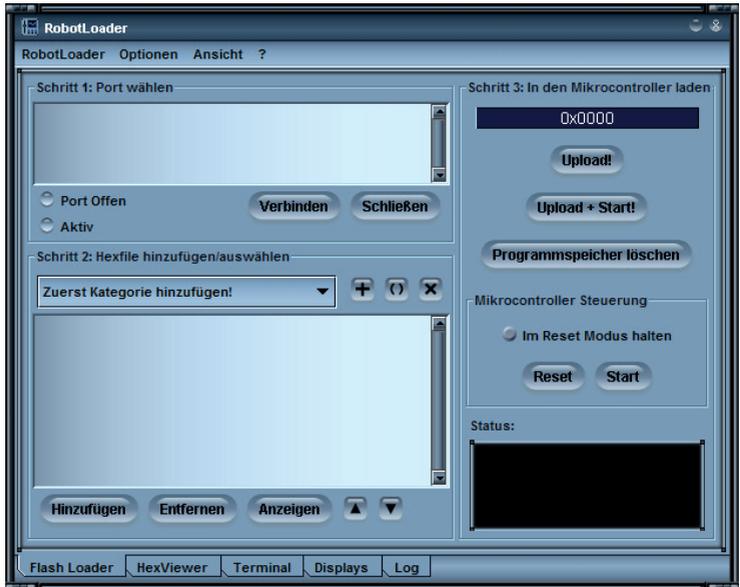


Fig.9 - RobotLoader

Ci sono integrate alcune funzioni aggiuntive utili come un semplice programma terminale.



Fig.10 - Schermo terminale

Il software RobotLoader non deve essere installato, devi semplicemente copiarlo in qualsiasi cartella del tuo harddisk.

7.2. Collegamento dell'interfaccia USB – Windows

Gli utenti Linux possono passare alla prossima sezione!

Esistono diverse opzioni per installare l'interfaccia USB, il modo più semplice è quello di installare il driver PRIMA di collegare l'hardware per la prima volta.

Il CD contiene un programma di installazione dei driver.

Per sistemi Windows 7 a 32 e 64 bit, XP, Vista, Server 2003 e 2000:
<CD-ROM drive>:\Software\USB_DRIVER\Win2k_XP\CDM_Setup.exe

Per i vecchi sistemi, come Win98SE/Me, non esiste un programma semplice di installazione. Su questi sistemi devi installare i driver manualmente dopo aver collegato l'hardware (vedi sotto).

Esegui perciò il programma di setup, alla fine verrà visualizzato un messaggio il quale avverte che i driver sono stati installati in modo corretto.

Ora puoi collegare l'interfaccia USB al PC.

PER PIACERE NON COLLEGARE ORA IL ROBOT!

Connetti il PC tramite il cavo USB.

ATTENZIONE:

- 1) tocca il PCB dell'interfaccia USB solo ai bordi o dal connettore USB o il guscio di plastica del connettore di programmazione (vedi le note di sicurezza riguardanti le cariche elettrostatiche)!
- 2) Non toccare alcun componente presente sul PCB, alcun punto di saldatura o i contatti del connettore IDE se non se strettamente necessario onde evitare cariche elettrostatiche!

Il driver precedentemente installato verrà usato automaticamente dal dispositivo senza bisogno del tuo aiuto. In ambiente Windows XP/2k appariranno piccoli avvisi sulla task bar. L'ultimo messaggio dovrebbe essere:

“The device has been successfully installed and is ready for use!”.

Non ha molta importanza se hai collegato l'interfaccia USB prima dell'installazione (o usi Win98/Me) Windows ti chiederà il driver.

Anche questo metodo di installazione è possibile.

Il driver si trova sul CD in formato compresso.

Se, in ambiente Windows, la situazione è quella precedente allora comparirà un messaggio con richiesta di installare il nuovo driver.

Devi allora indicare al sistema dove si trova il driver.

In Windows 2k/XP devi prima selezionare l'installazione manuale e non cercare i driver tramite il servizio web. Sul nostro CD il driver si trova nella cartella sotto citata.

E' sufficiente indicare la cartella per la tua versione di Windows ed eventualmente quella dei pochi altri files che il sistema non è in grado di trovare automaticamente (si trovano tutti nella cartella sotto citata!)

Sotto Windows XP e versioni successive esce spesso un messaggio che dice che i drivers FTDI non sono firmati o verificati da Microsoft (normalmente non in questo caso essendo i drivers FTDI firmati).

Questo messaggio non è comunque importante e puoi confermare senza problemi l'installazione.

7.2.1. Attivazione

Per **sistemi Windows 7 a 32 e 64 Bit, XP, Vista, Server 2003 e 2000** :

<CD-ROM drive>:\Software\USB_DRIVER\Win2k_XP\FTDI_CDM2\

Per vecchi **sistemi di Windows 98SE/Me**:

<CD-ROM drive>:\Software\USB_DRIVER\Win98SE_ME\FTDI_D2XX\

Dopo l'installazione del driver con vecchie versioni come Win98SE può essere necessario il riavvio del computer! **ATTENZIONE:** con Win98/Me solo uno dei due drivers è funzionante: Virtual Comport o il driver D2XX di FTDI! Sfortunatamente non esistono drivers che offrono entrambe le funzioni. Normalmente non esiste una porta COM virtuale disponibile dato che RP6Loader sotto Windows usa come standard i drivers D2XX (tu puoi cambiarlo - contatta eventualmente il nostro supporto tecnico!).

7.2.1. Verifica il collegamento del Dispositivo

Per verificare se il dispositivo è stato installato in modo corretto, sotto Windows XP, 2003 and 2000, puoi usare il comando "gestione periferiche" in alternativa a RobotLoader: fai clic con il tasto destro del mouse su:

Risorse del Computer--> Proprietà --> Hardware --> Gestione dispositivi

O altrimenti:

Start --> Impostazioni --> Pannello di controllo --> Prestazioni e manutenzione --> Sistema --> Hardware --> Gestione dispositivi e verifica che nella vista ad albero sotto

"Porte(COM e LPT)" esista "USB-Serial Port (COMX)", la X sostituisce il numero della porta, o guarda sotto "USB serial bus controller" per un "USB Serial Converter" !

7.2.1. Rimozione dei drivers

Se poi desideri rimuovere il driver (no, non ora, questo è un suggerimento nel caso in cui tu voglia farlo dopo alcuni giorni): se hai usato il programma di installazione del CD ROM, puoi disinstallarlo direttamente tramite

Start --> Impostazioni -->Pannello di controllo --> Software.

Nell'elenco visualizzato troverai una voce "FTDI USB Serial Converter Drivers", selezionala e fai clic su "uninstall".

Se hai installato il driver manualmente, puoi eseguire il programma "FTUNIN.exe" che si trova nella cartella dedicata al driver USB relativo al tuo sistema!

Attenzione: altri adattatori USB-->RS232 con il chip FTDI usano questo driver! Fai attenzione prima di disinstallarlo.

7.3. Collegamento dell'interfaccia USB – Linux

Gli utenti Windows possono tralasciare questa sezione!

I sistemi Linux con kernel 2.4.20 o maggiore includono già il driver richiesto (almeno per il modello precedente FT232BM compatibile con il chip sulla nostra interfaccia USB, il FT232R). L'hardware è automaticamente riconosciuto e tu devi fare niente.

In caso di problemi, puoi scaricare i drivers per Linux (ed il supporto per i drivers più recenti) direttamente da FTDI:

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

Tuttora che l'hardware è stato collegato, tramite il comando di Linux:

```
cat /proc/tty/driver/usbserial
```

puoi verificare se la porta seriale USB è stata installata in modo corretto. Questo è quanto devi fare.

Vale la pena ricordare che RobotLoader, sotto Windows, usa i drivers D2XX e che i descrittori completi USB vengono visualizzati nell'elenco delle porte (esempio "USB0 | Robot USB Interface | serialNumber")

Mentre in Linux i descrittori delle porte virtuali vengono visualizzati come /dev/ttyUSB0, /dev/ttyUSB1 ecc... Le porte COM normali sono visualizzate come "dev/ttyS0" ecc...

In questo caso devi verificare qual'è la porta corretta!

Purtroppo Linux non dispone di un driver che fa entrambe le cose, quindi ha più senso usare i drivers della Virtual Comport che sono presenti nel Kernel. L'installazione del driver D2XX richiederebbe parecchio lavoro manuale...

7.4. Fine dell'installazione del Software

L'installazione del software e dell'interfaccia USB è terminata! Ora copia i files più importanti dal CD all'hard disk (in modo particolare la cartella completa "Documentation" e, se non è già stato fatto, i programmi d'esempio), questo evita di cercare continuamente i files di cui hai bisogno sul CD. Le cartelle sul CD sono nominate in modo tale che tu puoi trovare facilmente i pacchetti software o la documentazione!

Se un giorno "perdi" il CD puoi scaricare i files più importanti, come questo manuale, RobotLoader e i programmi d'esempio dalla home page di AREXX. Qui troverai anche i collegamenti agli altri pacchetti software di cui hai bisogno.

7.5. Test interfaccia USB e avvio di RobotLoader

Il prossimo passo è fare il test del programma di upload tramite interfaccia USB. Collega l'interfaccia USB al PC (per primo collega sempre il PC!) e l'altro capo del cavo piatto a 10 pin al connettore su Robot Arm "PROG/UART". (Robot Arm DEVE ESSERE SPENTO!).

Il cavo piatto a 10 pin è meccanicamente protetto contro l'inversione di polarità. Tranne che non sia forzato non può essere collegato in modo errato.



Fig.11 - Selezione lingua

Avvia RobotLoader.

Il menù, in base alla lingua che hai scelto, può contenere vocaboli diversi. Lo screen shots mostra la versione Inglese.

Tramite la voce di menù

"Options->Preferences" sotto

"Language /Sprache" puoi scegliere il linguaggio desiderato (Inglese o Tedesco).

Quindi fai click su OK.

Tuttora che hai scelto la lingua devi riavviare RobotLoader per rendere effettive le modifiche!

Aprire una porta seriale - Windows



Fig.12 - Selezione porta seriale

Seleziona la porta USB. Fino a quando non ci saranno altri adattatori USB>Serial con controller FTDI collegati al PC, vedrai solo un singolo collegamento che dovrai scegliere.

Se esiste più di una porta dovrai trovare quella con il nome "Robot USB Interface" (o „FT232R USB UART“).

Il numero di serie programmato apparirà dietro al nome della porta.

Se nessuna porta è visualizzata, puoi aggiornare l'elenco delle porte tramite la voce di menù "RobotLoader-->Refresh Portlist" !



ATTENZIONE!

Se la tensione dovesse scendere al di sotto di < 6.7 V, verrà visualizzato un messaggio di allarme. verrà visualizzato un messaggio di attenzione.

7.6. Aprire una porta seriale – Linux

Linux gestisce l'adattatore seriale USB come una normale porta COM. L'installazione del driver D2XX di FTDI sotto Linux potrebbe non essere così semplice ed i drivers della normale porta seriale virtuale (VCP) sono comunque inclusi negli attuali Kernel di Linux. Funziona quasi come sotto Windows. Devi conoscere il nome dell'interfaccia USB di Robot Arm ed essere certo che la porta USB non venga scollegata dal PC fino a quando la connessione è aperta (altrimenti dovrai riavviare RobotLoader per riconnetterti). Sotto Linux i nomi delle porte seriali virtuali sono `"/dev/ttyUSBx"`, dove x rappresenta il numero della porta, per esempio `"/dev/ttyUSB0"` o `"/dev/ttyUSB1"`. I nomi delle porte seriali normali sotto Linux sono `"/dev/ttyS0"`, `"/dev/tty- S1"` ecc... Nel momento in cui sono attive vengono mostrate nell'elenco delle porte.

RobotLoader, nel caso in cui esistano diverse porte, si ricorda quale è stata usata l'ultima volta e la sceglie automaticamente quando avvii il programma (normalmente vengono ricordati i vari settaggi e selezioni).

Ora puoi fare click sul pulsante "Connect"! RobotLoader aprirà la porta e verificherà se la comunicazione con il bootloader, che si trova sul robot, è attiva.

Il campo nero "Status" nella parte bassa dovrebbe visualizzare il messaggio "Connected to: Robot Arm ..." o qualcosa di simile insieme ad altre informazioni come il valore di tensione misurato. Se così non fosse, riprova! Se non funziona ancora allora c'è un errore!

Spegni immediatamente il robot e prova a cercare l'errore.

Se la tensione è troppo bassa viene visualizzato un messaggio di attenzione. Devi perciò caricare immediatamente le batterie (preferibilmente anche prima che la tensione scenda al di sotto di 6,70V)!

ATTENZIONE!



Se la tensione dovesse scendere al di sotto di < 6.7 V, verrà visualizzato un messaggio di allarme.

7.7. SELFTEST

Il LED giallo, tensione, lampeggia quando Robot Arm è acceso.

Il LED status lampeggia quando viene caricato un file HEX.

Se l'avvio è andato a buon fine puoi eseguire un piccolo programma di test per verificare il funzionamento di tutti i sistemi del robot.

Fai click sul pulsante "Add" che si trova nella parte bassa della finestra del Robot Loader e quindi, nella cartella "example" di RobotArmExamples [PRO], seleziona il file

„Example_11_Selftest\RobotArm_Selftest.hex“.

Questo file contiene il programma di test in formato esadecimale, per questo motivo questo tipo di file di programma è chiamato "hex file". Il file appena selezionato apparirà successivamente nell'elenco. In questo modo puoi aggiungere altri files esadecimali dai tuoi programmi e dai programmi d'esempio (guarda lo screen shot dove sono stati aggiunti alcuni files .hex).

Il RobotLoader è in grado di gestire diverse categorie di files .hex.

Questo permette di ordinare in modo più chiaro i files nel caso in cui, ad esempio, moduli con diverse estensioni programmabili siano montate sul robot o siano utilizzate diverse versioni di programma.

L'elenco è salvato in modo automatico alla fine dell'esecuzione del programma. Naturalmente sono salvati solo i percorsi relativi ai files .hex e non i files .hex completi. Se stai lavorando su un programma, devi aggiungere e scegliere il file .hex solo una volta. Puoi caricare il nuovo programma nel microcontrollore dopo ogni nuova compilazione dello stesso (puoi anche usare la combinazione di tasti [STRG+D] o [STRG+Y], per avviare direttamente il programma dopo il trasferimento). A seconda del sistema operativo usato i nomi dei percorsi variano notevolmente. Tuttavia è possibile utilizzare RobotLoader sia in Windows che in Linux, senza alcuna modifica, in quanto ci sono due liste separate, una per Windows ed una per Linux.

In entrambi i casi puoi continuare ora con altri programmi di esempio (Examples) del Robot Arm oppure iniziare con la tua programmazione software.

ATTENZIONE!



Il programma di test della pulsantiera è già preinstallato, in questo modo puoi partire con le funzioni di verifica della pulsantiera.

Vedi pag.56 al paragrafo "Test della pulsantiera"

Seleziona il file "RobotArm_Selftest.hex" nell'elenco e fai click sul pulsante "Upload!" che si trova nella parte alta destra della finestra.

Il programma verrà trasferito nel processore MEGA64 su Robot Arm. Questa operazione dovrebbe impiegare non più di pochi secondi (max. 5 seconds per il programma di autotest).

Seleziona il tab "Terminal" (nella parte bassa della finestra)!

Puoi aprire la finestra del terminale Fig.16 tramite l'opzione del menù: "View -->Terminal" oppure utilizzando il tasto "F4".



Importante!

Il jumper "ISP/BOOT" durante l'uso normale dovrebbe essere nella posizione di sinistra, vedi Fig.15.



Fig.13 - jumper ISP/BOOT

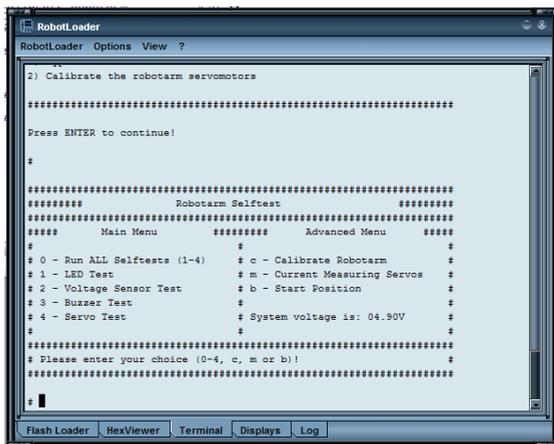


Fig.14 - Finestra terminale

Ora puoi eseguire il selftest e la calibrazione del Robot Arm.

Premi il pulsante Start/Stop Reset presente su Robot Arm per avviare il programma.

In seguito potrai fare questo tramite il comando di RobotLoader menu --> Start o la combinazione dei pulsanti [STRG]+[S].

Tuttavia ora puoi verificare se il pulsante funziona in modo corretto.

Se durante la fase di test si verifica un errore, spegni immediatamente il robot e prova ad individuare l'errore.

SI RACCOMANDA DI INIZIARE CON LA CALIBRAZIONE DEL ROBOT ARM VEDI pag.54.

7.8. Calibrazione

Avvia il programma di calibrazione per calibrare il robot.

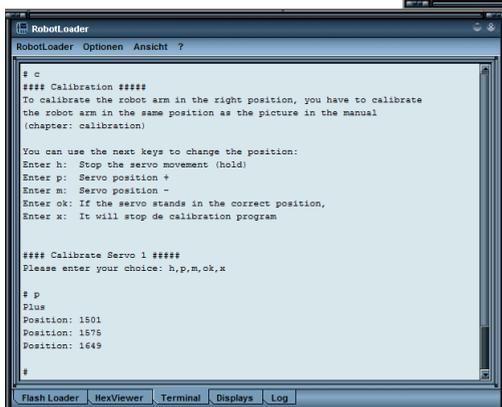
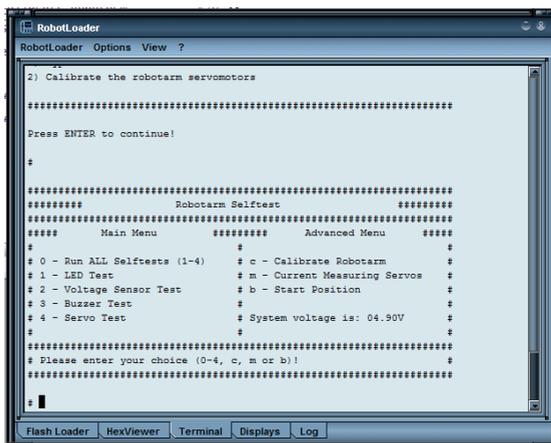
Per questo scopo devi fare click sul pulsante "Add" che si trova in basso a sinistra della finestra di RobotLoader e seleziona il file:

RobotArmExamples [MINI],
"Example_11_Selftest\RobotArm_Selftest.hex"
che si trova nella cartella esempi.

Questo file contiene il programma di test in formato esadecimale. Il file appena selezionato apparirà in sequenza nell'elenco. (vedi screenshot).

Nella finestra Fig.17 del terminale seleziona C
(C - Calibrate)
per iniziare la fase di calibrazione.

Fig.15 - Selezione calibrazione



Porta tutti i servomotori in posizione centrale in modo che il robot si trovi come a pag.46

I servomotori 2-6 sono approssimativamente in posizione centrale ed il dito (servo 1) è aperto o chiuso (dipende dalla versione).

Fig.16 - Finestra di calibrazione

Nel momento in cui la calibrazione Fig.18 (C - Calibrate) è completata il robot può eseguire il successivo selftest.

Il risultato della calibrazione è salvato nella eeprom di ATMEGA.

7.9. Posizione dopo la calibrazione

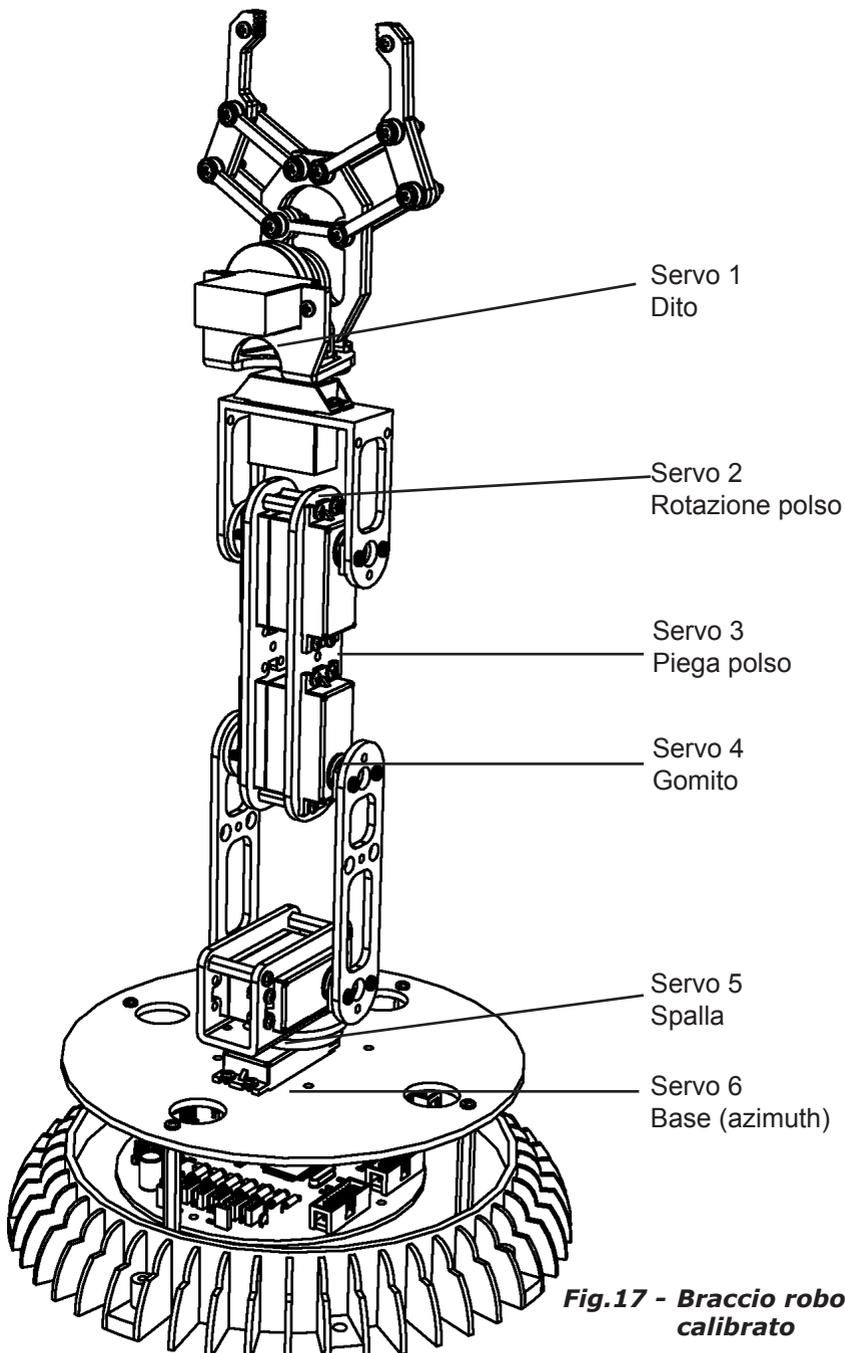


Fig.17 - Braccio robotico calibrato

7.10. Test della pulsantiera

Il set è fornito con una pulsantiera che può essere collegata al Robot Arm.

Questa è una buona opzione per semplici dimostrazioni e per permetterci di fare pratica con il controllo di un braccio robotico usando una pulsantiera.

Sulla pulsantiera si trovano 6(x2) pulsanti di controllo e 4 pulsanti speciali per estensioni future.

Se vogliamo fare il test di Robot Arm tramite pulsantiera, dobbiamo trasferire l'opportuno programma hex nel microprocessore del robot.

Fai click sul pulsante "Add" che si trova in basso a sinistra della finestra di RobotLoader e seleziona il file RobotArmExamples, "RobotArm_Key_Board.hex" nella cartella esempi.

Seleziona il file "RobotArm_Key_Board.hex" nell'elenco e successivamente premi il pulsante "Upload!" nella parte alta a destra.

Dopo che hai fatto questo, puoi controllare Robot Arm semplicemente tramite i pulsanti presenti sulla pulsantiera.

UTILIZZA LA VERSIONE CORRETTA V3 PRO O MINI!

Puoi eseguire un test della pulsantiera con il SELFTEST SOFTWARE.

Normalmente il software di test della pulsantiera è installato dal costruttore. In questo modo puoi eseguire il test non appena accendi il

r Fig.19 - Circuito stampato Robot

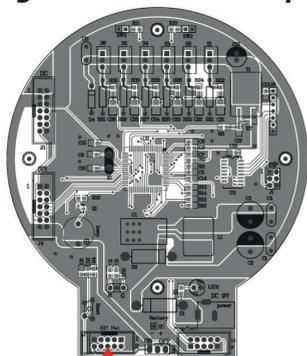
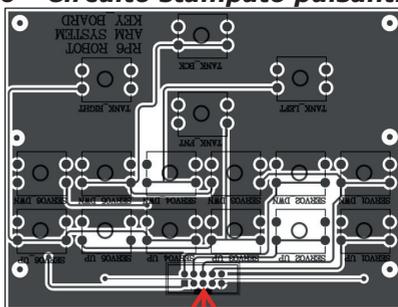


Fig.18 - Cavo della pulsantiera

Fig.20 - Circuito stampato pulsantiera



8. Software RACS

RACS (Robot Arm Control Software) è il modo più semplice per controllare e programmare Robot Arm. La programmazione tramite il metodo RACS richiede il software RobotLoader e l'adattatore di programmazione USB.

Prima di usare il robot, devi caricare il file .hex "RACS-PRO.hex" nella memoria Flash del processore.

Collega il cavo di programmazione/controllo alla porta USB sul tuo computer ed avvia il programma di Loader, verrà visualizzata la seguente interfaccia utente:

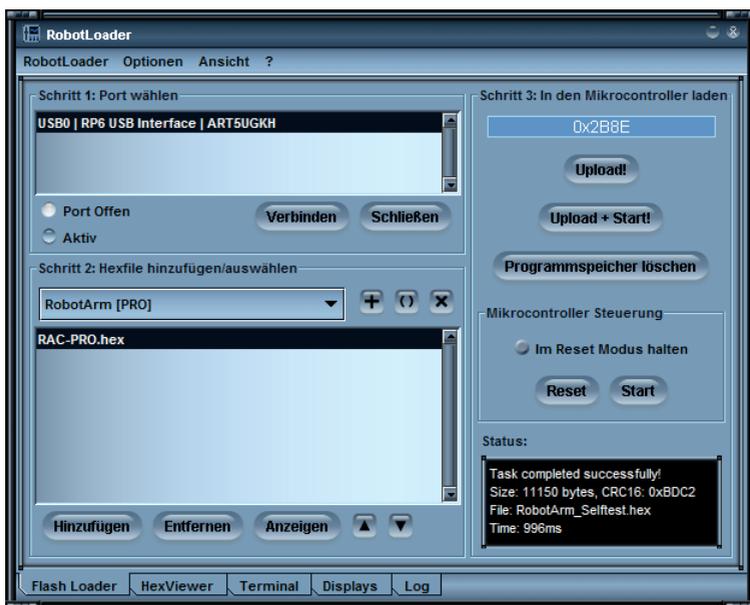


Fig.21 - Finestra interfaccia utente di RobotLoader

Nel caso in cui nell'elenco "Step 1: Select a port" non appaia nessuna porta USB, accertati che il cavo sia collegato e che siano installati i drivers di programmazione. Puoi richiamare l'elenco delle porte tramite il menù:

RobotLoader -> Refresh port list.

Seleziona la porta e fai click su "Connect".

In "Step 2:Add/Select Hexfile" scegli il file .hex opportuno

Fai click su "Add": **RACV3-PRO.HEX**



Utilizza la corretta versione V3PRO oppure MINI SOFTWARE!

In "Step 3:Upload to target" fai click sul pulsante "Upload" per importare il file.

Se vuoi azionare Robot Arm, devi disconnettere il RobotLoader allo "Step 1" facendo click sul pulsante "Close".

Se chiudi il programma il collegamento sarà interrotto in modo automatico.

Per piacere accertati che non siano in esecuzione contemporaneamente il software Loader ed il Robot Arm, altrimenti il robot non può essere controllato tramite il software RACS.

8.1. Manuale di istruzione di RACS

Robot Arm può essere facilmente controllato tramite il software RACS.

Un collegamento è stabilito tra il cavo di programmazione /controllo e quindi i motori di Robot Arm reagiscono alla posizione del cursore impostata tramite mouse. La posizione corrente può essere salvata, modificata e cancellata nella list box che si trova nella parte bassa dell'interfaccia utente. Questo genera un elenco contenente le posizioni singole che possono essere salvate come file sul computer cliccando sul pulsante "Save". Questo elenco può essere caricato in qualsiasi istante.



Importante !!!

Il robot controlla la corrente del motore e quella di ogni singolo servomotore. Se la soglia di corrente di un servo è eccessiva, esempio durante una collisione o sovraccarico, il testo nel software RACS inizia a lampeggiare.

In questo caso il robot deve essere riportato nella sua ultima posizione il più presto possibile oppure l'alimentazione dei servo nel software RACS deve essere disabilitata.

8.2. RACS - Collegamenti

- 1) Avviare con un doppio click il software di controllo di Robot Arm.
Si aprirà la finestra di Fig.24:

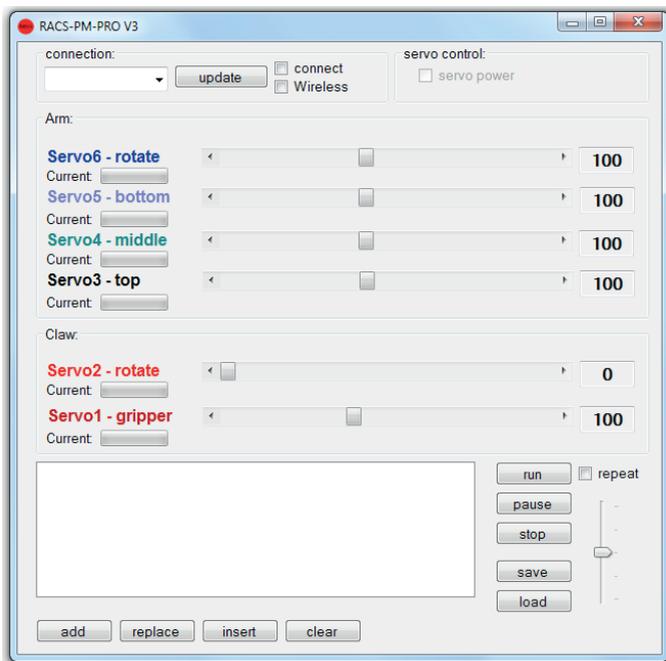


Fig.22 - Finestra programma RACS

- 2) Nel menù a tendina verranno visualizzate tutte le interfacce seriali disponibili come in Fig.25.

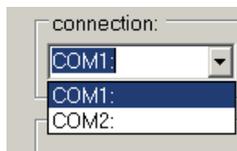


Fig.23 - Interfacce seriali disponibili

- 3) Collega il programmatore USB.
- 4) Fai Click sul pulsante "Update". Se riguardi il menù a tendina noterai una ulteriore interfaccia. Questa interfaccia è stata aggiunta inserendo il programmatore USB.

Attenzione:

I nomi delle interfacce possono essere diversi da un computer

5) Seleziona la nuova interfaccia



Fig.24 - Selezione interfaccia seriale

6) Abilita l'opzione "Connect"



Fig.25 - Abilitazione connessione seriale

7) Abilita l'opzione "servo power"



Fig.26 - Abilitazione alimentazione Servo

7) Muovi i cursori per controllare i servo. Se durante il collegamento si dovesse verificare un errore, apparirà la finestra di Fig.29. In questo caso deve essere ristabilito il collegamento ed occorre ripetere i passi da 2 a 6 e ricontrollare l'interfaccia.



Fig.27 - Errore di collegamento seriale

UTILIZZA IL CORRETTO SOFTWARE V3 PER PRO OR MINI

8.3. RACS – Controllo Automatico di Posizione

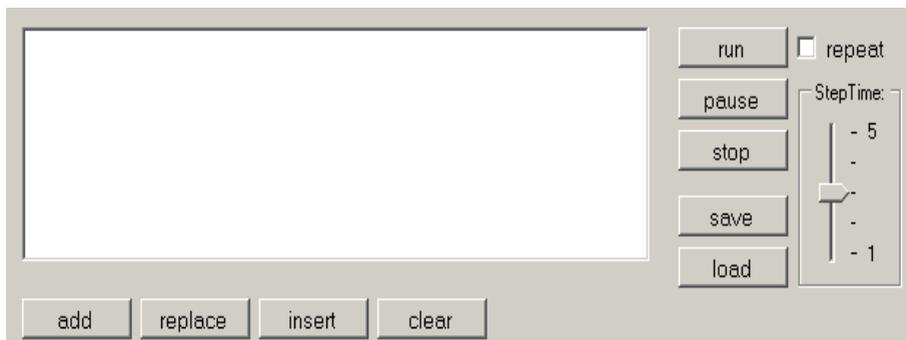


Fig.28 - Finestra controlli

Nella finestra di Fig.30 sono disponibili i seguenti controlli:

- Add:** questo pulsante aggiunge all'elenco la posizione corrente.
- Replace:** la voce selezionata nell'elenco sarà sostituita dalla posizione corrente del cursore.
- Insert:** la posizione corrente del cursore verrà inserita nella posizione superiore della voce selezionata.
- Clear:** la voce selezionata nell'elenco verrà cancellata.
- Save:** le voci dell'elenco corrente verranno salvate in un file.
- Load:** le voci dell'elenco verranno caricate da un file (**Attenzione**, le voci attuali dell'elenco saranno cancellate!).
- Run:** Le voci dell'elenco verranno eseguite in sequenza partendo dall'alto. Se è stata selezionata l'opzione "Repeat" le voci dell'elenco saranno eseguite continuamente.
- Step Time:** tempo che definisce il ritardo di esecuzione tra un comando e l'altro dell'elenco. Se ci sono comandi di breve durata questo ritardo può essere breve altrimenti per comandi di esecuzione lunga, come la rotazione di 180° del servo, occorre aumentare il tempo.
- Pause:** Il comando verrà messo in pausa.
- Stop:** Il comando sarà arrestato.

8.3.1. RACS – WIRELESS

Con il software RACS ed il set di schede ARX-APC-220 di AREXX puoi controllare Robot Arm anche senza fili.

Di seguito viene descritto passo dopo passo come fare:

- 1) collega al PC il programmatore RP6v2 e la scheda APC-220 esattamente com'è descritto sul CD;
- 2) collega l'altro APC al PCB di Robot Arm (vedi Fig.6);
- 3) carica nel processore di Robot Arm il file Wireless Racs HEX;
- 4) seleziona la porta seriale COM corretta nel software RACS.
 - * **Premi il pulsante di START (Robot Arm PCB)**
 - * Abilita --> **Wireless - Connect e Servo Power**
- 5) Ora sei in grado di controllare Robot Arm in modalità senza fili.

Fig.29 - Modulo wireless ARX-APC-220

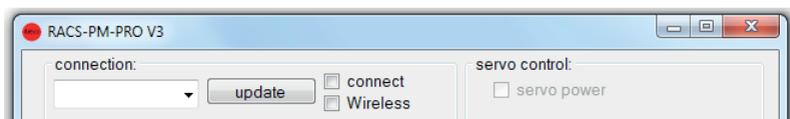


Fig.30 - Impostazioni di RACS

8.3.1. PROGRAMMA ANDROID

Il nostro modulo ARX-BT03 ti permette di controllare Robot Arm tramite Bluetooth utilizzando un' applicazione ANDROID.

Segui le istruzioni sotto:

- 1) collega il modulo Bluetooth al PCB di Robot Arm (vedi Fig.6;
- 2) carica l'applicazione ANDROID HEX nel processore di Robot Arm;
- 3) carica l'applicazione ANDROID APK nel tuo smartphone o tablet. Questa applicazione la puoi trovare sul CD oppure in Google Play Store.



Fig.32 - Simbolo Bluetooth

Fig.31 - Modulo Bluetooth ARX-BT03

9. Programmazione di Robot Arm

Iniziamo ora, in modo graduale, la programmazione del robot.

9.1. Impostazioni dell'editor di testo

Come prima cosa dobbiamo impostare un piccolo ambiente di sviluppo per creare il codice sorgente (detto anche "sourcecode") per il nostro programma in linguaggio C da salvare in un modo o nell'altro nel nostro computer!

Per questo scopo non possiamo utilizzare programmi di editor come OpenOffice o Word! Questo va sottolineato dal momento che tutti non ne sono a conoscenza. Questi programmi sono adatti per scrivere manuali ma totalmente inadatti per scopi di programmazione.

Un testo sorgente è un testo che non ha caratteri di formattazione, al compilatore infatti non interessano le dimensioni dei font utilizzati oppure i loro colori...

Naturalmente ad una persona è molto più chiaro se alcune parole o tipi di testo vengono evidenziati a colori. Queste funzioni ed altre ancora sono contenute in "Programmers Notepad 2" (abbreviato con "PN2") che è l'editor che noi utilizzeremo per scrivere il codice sorgente (ATTENZIONE: in ambiente Linux devi usare un editor con caratteristiche simili a PN2). In Linux normalmente sono già installati alcuni editor come kate, gedit, exmacs o similari.

Oltre all'evidenziazione di parole chiave ed altre funzioni (chiamate "evidenziazione di sintassi") PN2 offre anche una semplice IDE. Questo permette di organizzare nei progetti diversi files sorgenti di testo e di visualizzare un elenco di files relativi al progetto stesso. Inoltre in PN2 puoi recuperare facilmente programmi simili ad AVR-GCC e ottenere i programmi opportunamente compilati tramite una voce di menù.

Normalmente AVR-GCC è una pura riga di comando che non ha interfaccia grafica...

Puoi trovare versioni aggiornate di Progr



Notepad alla homepage del progetto: <http://www.pnotepad.org/>

Le versioni più recenti di WINAVR non richiedono la creazione di ulteriori voci di menù!

Prendi nota:

In questa sezione non descriviamo come devi impostare le voci di menù di PN2 in quanto queste impostazioni sono già state predisposte per te nelle versioni più recenti di WINAVR!

UTILIZZA IL CORRETTO SOFTWARE V3 PER PRO OR MINI

Guarda a pag.56 come puoi aprire un esempio di progetto!

Se tu hai aperto un esempio di progetto, sullo schermo di PN2 vedrai qualcosa di simile:

File “Robot ArmExamples.ppg“ .

Questo è un gruppo di progetti per PN2 che carica tutti gli esempi ed in più la libreria di Robot Arm Library in un unico progetto (“Projects“).

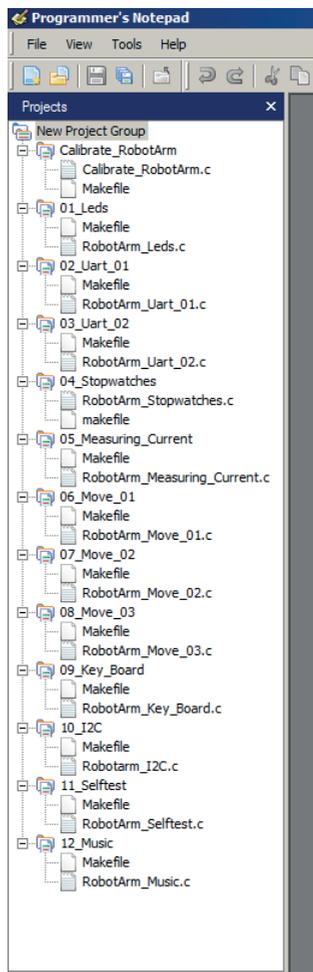
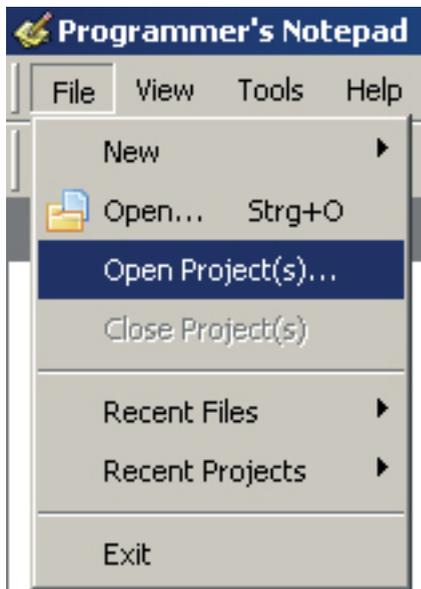


Fig.33 - Progetto PN2

Sul lato sinistro sono visualizzati tutti gli esempi, sul lato destro l’editor di testo dei files sorgenti (con la sintassi evidenziata come detto) e sul fondo l’uscita strumenti (in questo caso l’uscita del compilatore).

Con PN2 puoi fare molte altre cose in quanto PN2 offre diverse funzioni.

9.2. Apri e compila un esempio di progetto.



Se tutto funziona in modo corretto è ora possibile aprire un progetto d'esempio.

In "File" seleziona la voce di menù

"Open Project(s)".

Verrà visualizzata una finestra di dialogo vedi Fig.36. Nella cartella dove hai salvato i files d'esempio cerca un'ulteriore cartella:

"Robot Arm_Examples [PRO]"

Apri il file

"Robot ArmExamples.ppg".

Questo è un gruppo di progetti per PN2 che carica tutti i programmi d'esempio ed anche la libreria di Robot Arm Library nell'elenco del progetto ("Projects") vedi Fig.35

Fig.34 - Aprire un progetto con PN2

Ora tutti gli esempi sono disponibili in modo che puoi fare loro riferimento o guardare le funzioni di libreria di Robot Arm ecc...

Apri il primo esempio nella parte alta dell'elenco ("01_Leds" e seleziona il file "01_Leds") che appare nella parte sinistra della finestra programmi! Ora fai doppio click su "01_Leds.c"!

In una finestra all'interno del programma verrà visualizzato l'editor di testo del file sorgente.

Nella parte inferiore della finestra di programma di PN2 dovrebbe essere visualizzata una zona per i messaggi d'uscita. Se così non fosse tu devi abilitare questa zona tramite il menù

"View" --> "Enable output",

o, se l'area è troppo piccola, aumentare le dimensioni tirando i bordi con il mouse (il cursore del mouse cambia in una doppia freccia nel bordo superiore della zona grigia contrassegnata "output" nella parte inferiore della finestra del programma...).

Puoi dare una rapida occhiata al programma che hai appena aperto con l'editor di testo ma non occorre che tu capisca ora cosa succede esattamente. Tuttavia come prima informazione: i testi in verde sono commenti che non fanno parte del programma stesso. Essi sono utilizzati solo per scopi di descrizione / documentazione.

Torneremo più dettagliatamente in seguito (c'è anche una versione di questo programma SENZA commenti in modo che si può vedere quanto in realtà sia davvero corto. I commenti rendono di parecchio più grande un programma ma sono necessari per la sua comprensione. La versione non documentata è utile anche per copiare il codice nei vostri programmi!).

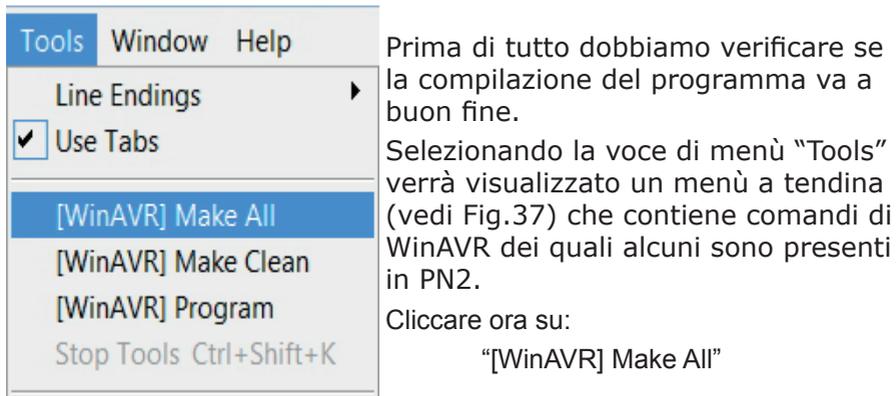


Fig.35 - Funzione "Make All"

PN2 recupera ora il file batch "make_all.bat" il quale eseguirà il programma "make". Maggiori informazioni su "make" verranno date in seguito.

Il programma di esempio viene ora compilato. Alla fine sarà generato un file hex che contiene il programma convertito in un formato adatto per essere caricato nel microcontrollore ed, in seguito, eseguito.

Il processo di compilazione genera un certo numero di files temporanei (suffissi come ".o", ".lss", ".map", ".sym", ".elf", ".dep").

Questi files vanno ignorati e potranno essere cancellati tutti, tranne il file HEX, tramite il comando "[WinAVR] Make Clean".

Il file HEX non va cancellato essendo l'unico che ci interessa.

Dopo l'esecuzione di menu MAKE ALL dovrebbe essere visualizzata la schermata di uscita (di seguito in una versione notevolmente accorciata. Alcune linee possono essere leggermente diverse!):

```
> "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
```

Il codice d'uscita "Process Exit Code: 0" è molto importante. Questo significa che, durante la compilazione, non si è verificato alcun errore. Nel caso in cui venga visualizzato un altro codice significa che il codice sorgente contiene un qualche errore che deve essere corretto prima di poter eseguire il programma. In questo caso il compilatore visualizzerà diversi messaggi d'errore i quali ci daranno maggiori informazioni.

Prendi nota che il messaggio "Process Exit Code: 0" non è garanzia di un programma senza errori. Il compilatore non conosce cosa vuole fare il tuo programma e per questo non può impedire che il robot vada a sbattere contro un muro ;-)

IMPORTANTE: puoi trovare dei messaggi di "warnings" ed altri messaggi ancora. Questi messaggi sono spesso molto utili ed indicano problemi importanti. Ecco perchè debbono essere sempre risolti. PN2 evidenzia i messaggi di "warnings" e di errori con colori diversi per rendere l'identificazione più semplice. Viene poi indicato anche il numero di riga dove il compilatore ha trovato l'errore. Se tu fai click sul messaggio d'errore colorato, PN2 salterà direttamente alla riga interessata.

Anche l'indicazione finale "AVR Memory Usage" è sempre d'aiuto.

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)

Questo significa che il nostro programma all'interno del processore Atmega64 occupa 3074 bytes di programma e che 68 bytes di RAM sono riservati per le variabili statiche (tu devi aggiungere a questo il range dinamico per heap e stack ma questa è un'altra cosa... tieni sempre un qualche centinaio di bytes di memoria libera).

Disponiamo di 64kb (65536 bytes) of Flash ROM e 2kb (2028 bytes) di RAM. Dei 64kb, 2k sono occupati dal bootloader - così noi possiamo usare solo 62kb.

Accertati sempre che il programma stia dentro allo spazio di memoria disponibile (RobotLoader non trasferisce il programma se è troppo grande!)

Ciò significa che i programmi di esempio sopra lasciano 60414 bytes di spazio libero.

L'esempio "Example_01_Leds.c", relativamente corto, è così grande solamente perché è inclusa la libreria ArmBaseLibrary!

Quindi, non preoccupatevi, vi è spazio sufficiente per i tuoi programmi e programmi così piccoli, di solito, non hanno bisogno di tanto spazio in memoria.

La funzione di libreria ha bisogno di diversi KB di memoria Flash, ma rende il lavoro molto più facile e di conseguenza i propri programmi saranno generalmente più piccoli rispetto al Robot ArmBaseLibrary.

Il programma appena compilato può essere caricato tramite il RobotLoader nel robot. Per fare questo, è necessario aggiungere il file hex appena generato nella lista nel RobotLoader tramite il pulsante "Add", selezionarlo e fai click sul pulsante "Upload" esattamente come hai fatto con il programma di autotest. Dopo di che è possibile passare al terminale e osservare l'output del programma.

Naturalmente è necessario avviare l'esecuzione del programma. Il modo più semplice per farlo da terminale è quello di premere la combinazione dei tasti [STRG] + [Y] sulla tastiera o utilizzare il menu (o semplicemente una "s" - dopo un reset si deve aspettare un pò fino a quando il "[READY]" non viene visualizzato nel terminale!).

La combinazione di tasti [STRG] + [Y] è anche molto conveniente in quanto il programma attualmente selezionato viene caricato nel Robot ARM e subito avviato. Questo evita di fare clic sulla scheda "Flash Loader" nel terminale oppure di utilizzare il menù.

Il programma di esempio è molto semplice ed è composto solo da un piccolo LED che lampeggia e qualche messaggio di testo.

10. Per concludere

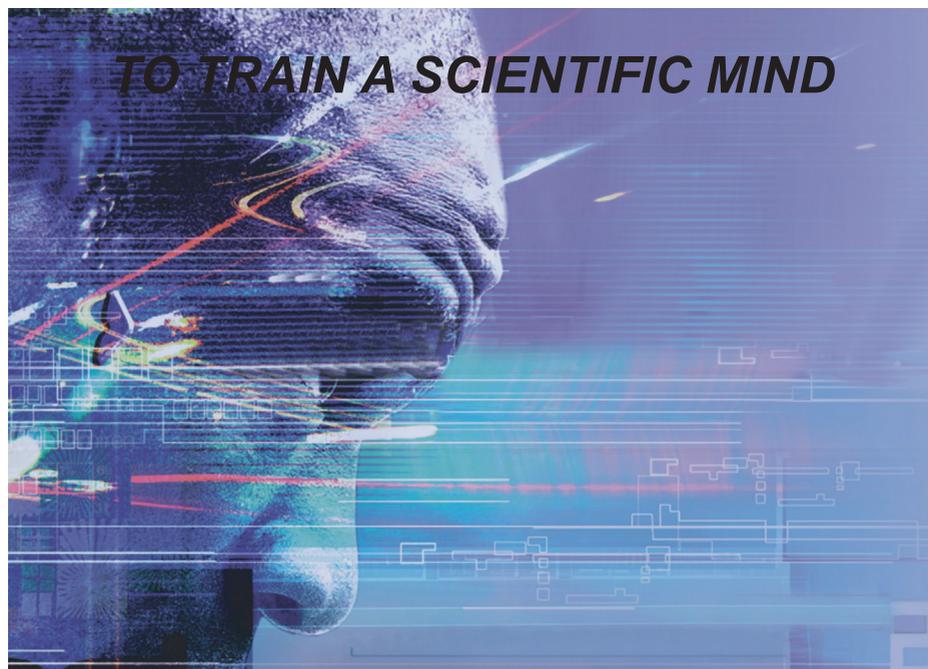
Speriamo che i nostri robot vi abbiano guidato sulla strada del mondo dei robot.

Condividiamo la convinzione dei nostri amici giapponesi che i robot diventeranno la prossima rivoluzione tecnologica dopo computer e telefoni cellulari.

Questa rivoluzione innescherà nuovi impulsi economici.

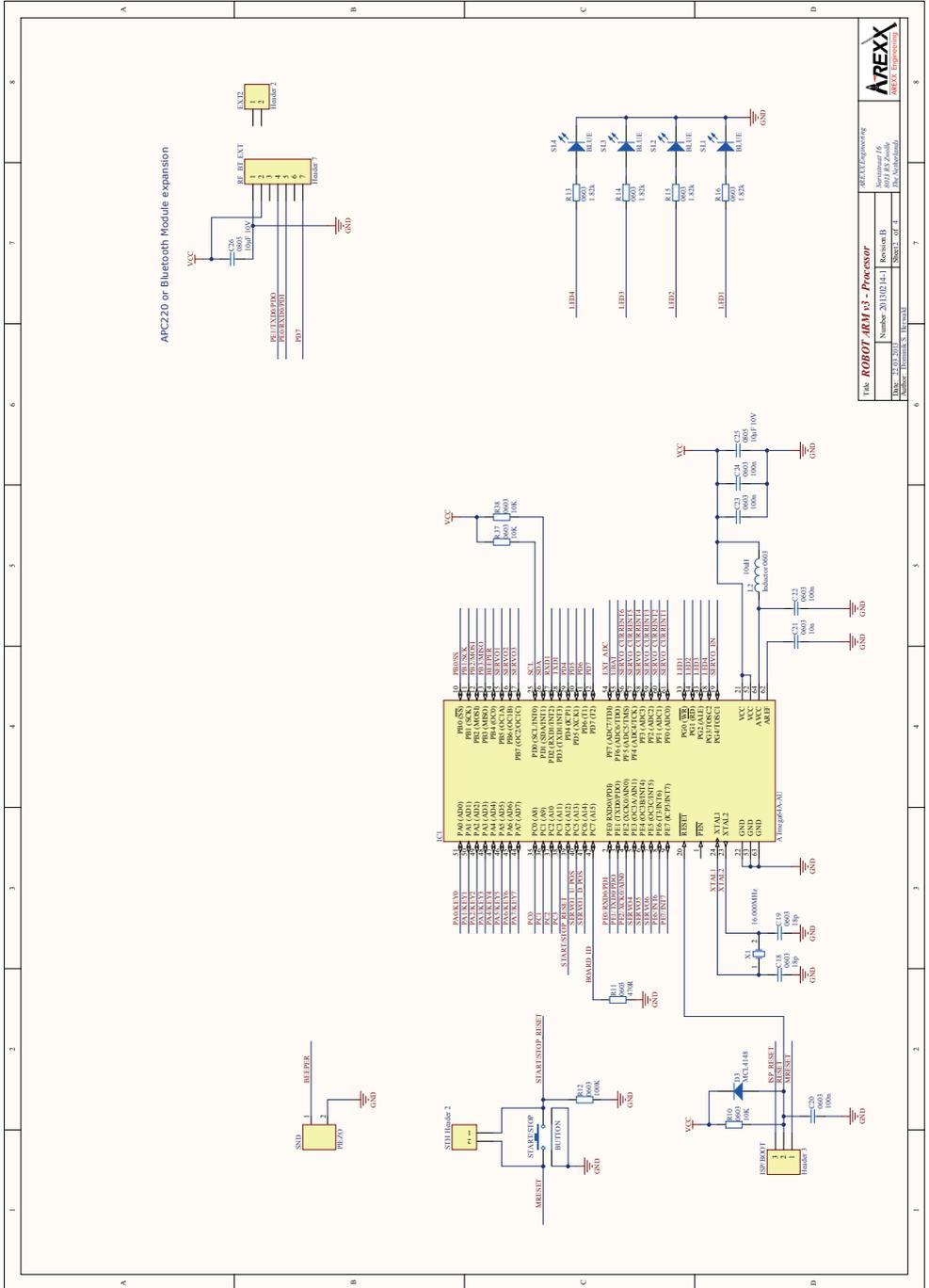
Purtroppo il Giappone, altri paesi dell'Estremo Oriente e anche gli Stati Uniti hanno ampiamente superato l'Europa in questo campo. A differenza dell'Europa, in Estremo Oriente corsi tecnici iniziano già nella scuola primaria e sono una parte importante della formazione.

Il nostro obiettivo nello sviluppo dei nostri robot ASURO, YETI, Caterpillar e Robot Arm è quindi:



APPENDICE

A. SCHEMA ELETTRICO DI ROBOT ARM RA1-PRO

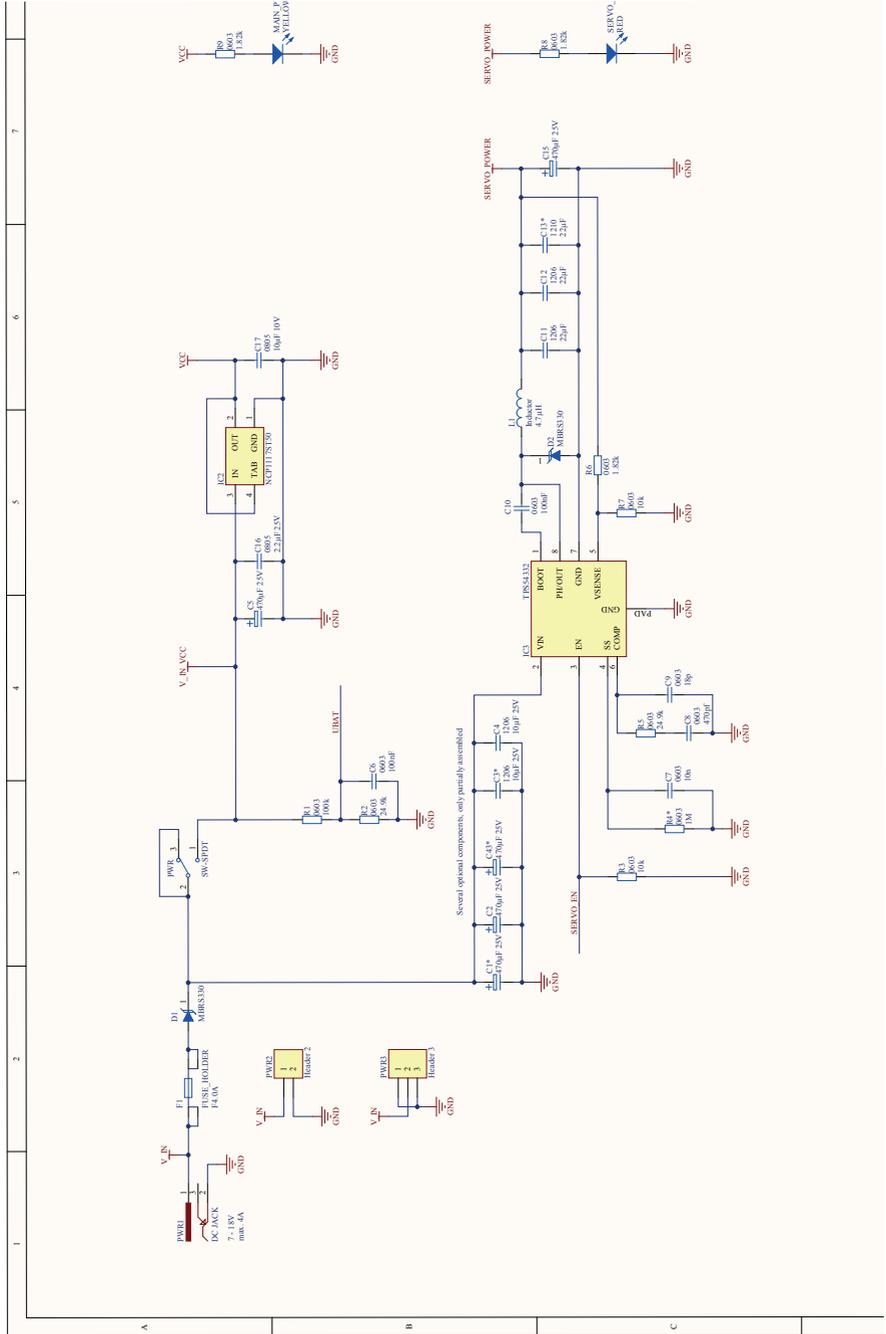


AREXX Engineering
 Number 20170104 | Revision B
 Date: 27.03.2017
 Author: Giovanni S. Iannace

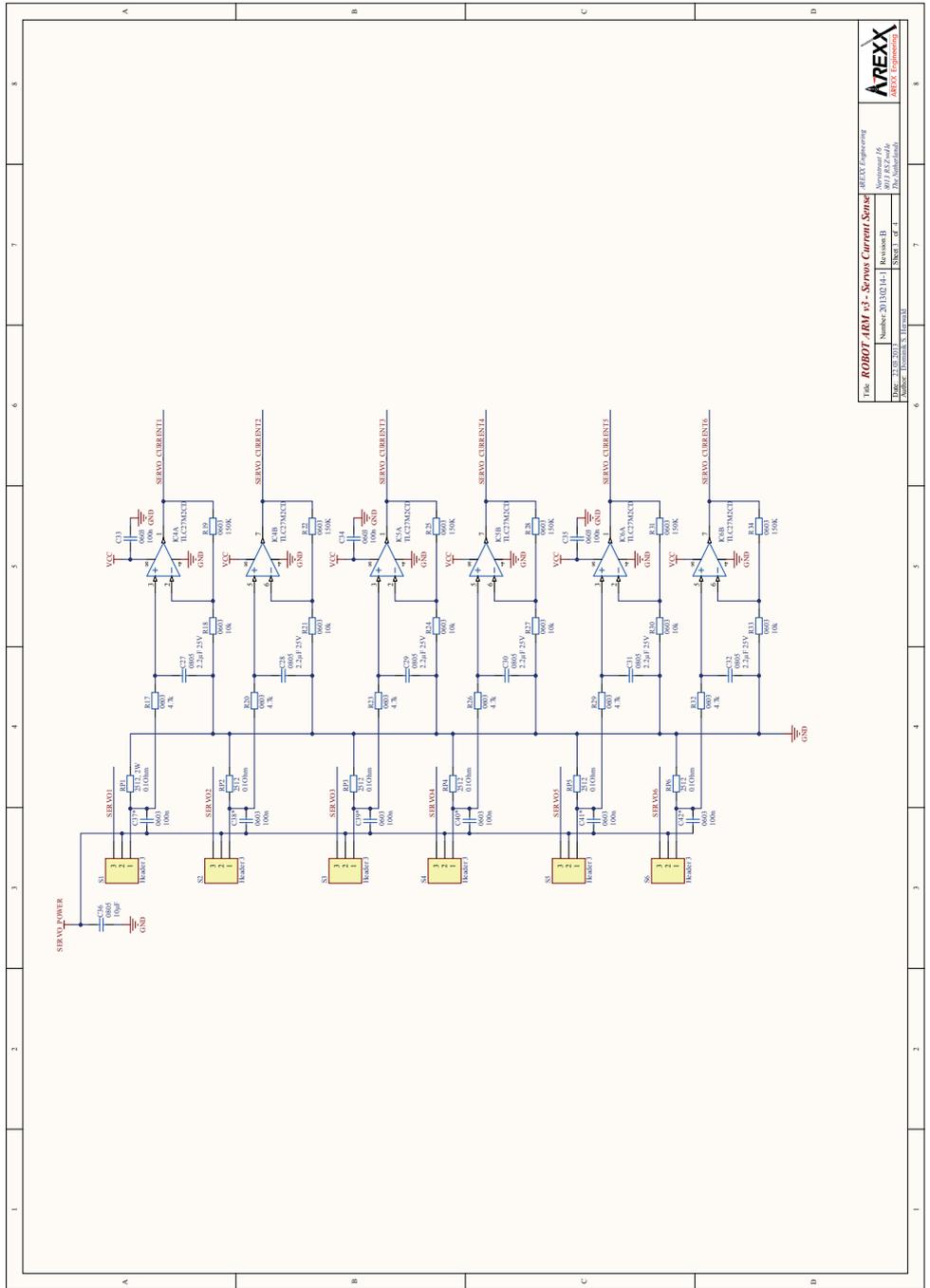
File: **ROBOT_ARM v3 - Processor**
 Number: 20170104 | Revision: B
 Date: 27.03.2017
 Author: Giovanni S. Iannace

AREXX Engineering

B. SCHEMA ELETTRICO DELL'ALIMENTAZIONE DI RA1-PRO



D. SCHEMA CONTROLLO CORRENTE SERVO V3





 AREXX Engineering

 Via S. Felice 16

 31044 Montebelluna

 TV (Treviso) - Italy

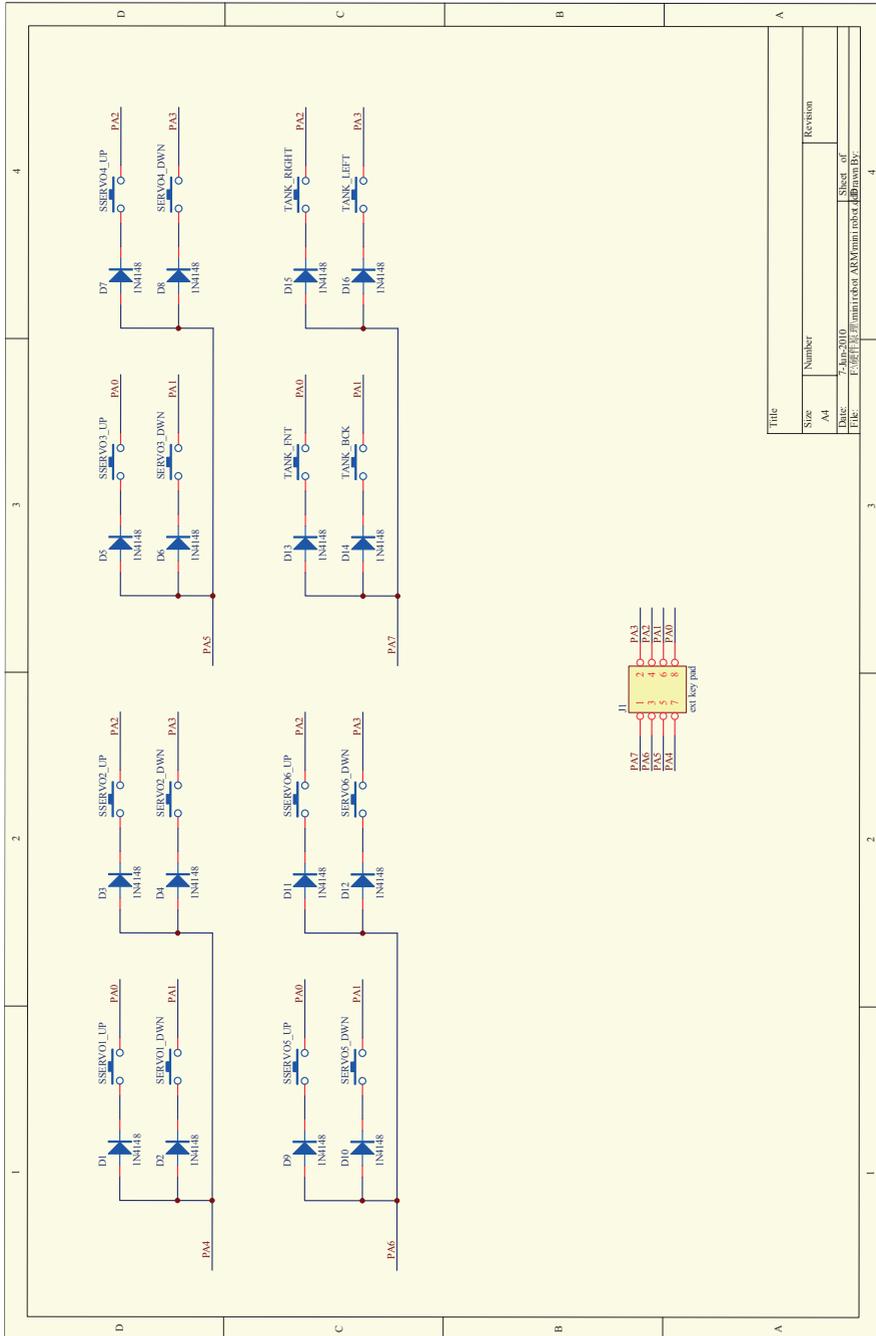
 Tel. +39 0422 740111

 Fax +39 0422 740112

 Email: info@arexx.it

 Web: www.arexx.it

E. SCHEMA ELETTRICO TASTIERA RA1-PRO



Title		Revision	
Size	Number		
A4			
Date:	2-Jun-2010	Sheet of	
File:	F:\proj\ra1\amin\ra1\ra1main\ra1_keyboard.dwg	Drawn By:	

F. CIRCUITO STAMPATO

