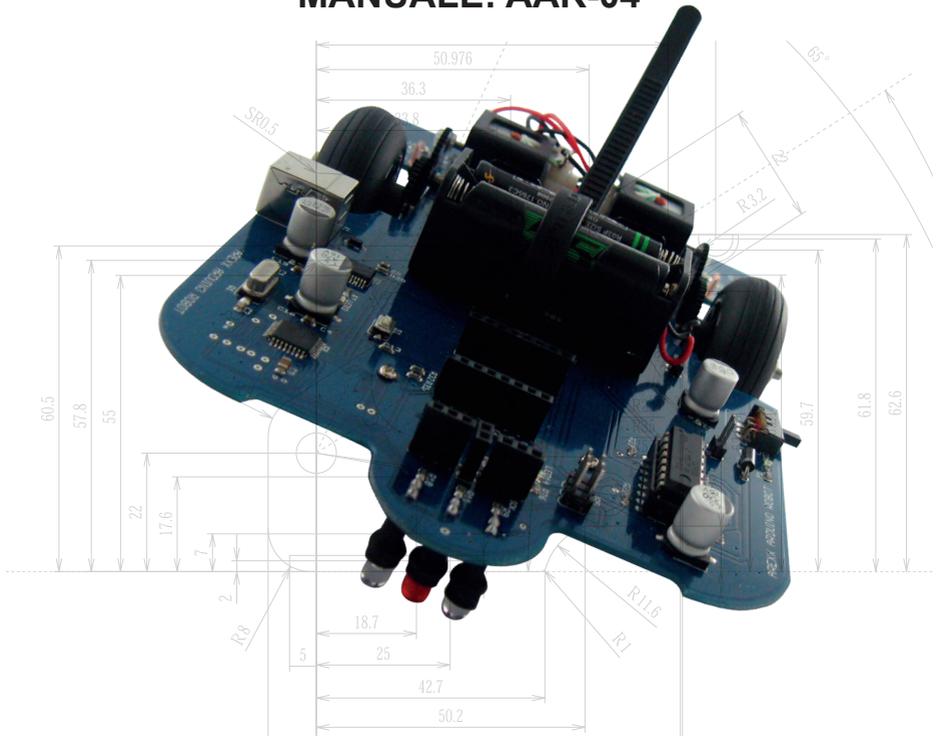




ARDUINO ROBOT AAR

MANUALE: AAR-04



Indice

1.Descrizione del Robot AAR	5
1.1.La Famiglia ARDUINO Robotics	5
1.2.Caratteristiche:	5
1.3.Precauzioni	6
2.ARDUINO Descrizione Generale.....	7
2.1.Chi è e cosa fa ARDUINO?.....	7
2.2.Microcontrollori!.....	8
2.3.Consumo di energia e velocità	9
2.4.Programmi per il Microcontrollore.....	9
3.ARDUINO ROBOT.....	11
3.1.LEDS DI ARDUINO ROBOT	11
3.2.Circuito stampato di ARDUINO ROBOT	12
3.3.Informazioni di base per AAR	13
3.4.Informazioni di base per Software Arduino	14
4. Si parte!	15
4.1.Scaricare ed installare il software di Arduino	15
4.2.Linguaggio di Arduino.....	15
4.3.Installazione del driver USB.....	15
4.4.Installazione delle batterie	16
4.5.Programmazione del Robot con Programmi Arduino.	17
4.6.Selezionare un programma Arduino.....	18
4.7.Selezionare la scheda Arduino	20
4.8.Selezione della Porta Seriale	21
4.9.Compilare lo Sketch	22
4.10.Caricamento dello Sketch nel robot	23
4.11.Risoluzione dei problemi	25
4.12.Autotest della scheda.	27
5.Kits di espansione per scheda AAR.....	29
5.1.Kits ASURO di espansione per Arduino Robot.	29
6.Configurazione Kits di Espansione di ASURO.	30
7.Programma ANDROID	32
7.1.Ponte H alimentato a 3 Volt.....	33
7.2.Un ponte H a 4,5 Volt.....	35
8.Odometria	36
8.1.Ricercatori di linea, amanti e nemici dei colori.....	36
8.2.Paure ed amori per la musica	36
8.3.Ricercatori di linea complessi.....	37
9.Programmare un Bootloader	39

Indice figure

Fig.1 - AAR PCB	11
Fig.2 - Collegamento del porta batteria.....	16
Fig.3 - IDE di Arduino	17
Fig.4 - Apertura Sketch "Blink"	18
Fig.5 - Listato programma "Blink"	19
Fig.6 - Selezione scheda Arduino per robot AAR.....	20
Fig.7 - Selezione porta COM.....	21
Fig.8 - Operazione di verifica dello Sketch.....	22
Fig.10 - Pulsante di invio dello sketch nel robot	23
Fig.11 - Fase di trasferimento dello sketch	23
Fig.12 - Caricamento dello sketch completato.....	24
Fig.13 - Gestione dispositivi	25
Fig.14 - Carica lo sketch nel robot.....	27
Fig.15 - Finestra terminale di Arduino.....	28
Fig.16 - Connettori per le espansioni	29
Fig.17 - RP6v2 Programmatore USB con APC 220	31
Fig.18 - Simulazione di un ponte H in un robot Hyper Peppy.	33
Fig.19 - Impostazioni DC per un ponte H di tipo Peppy Robot	34
Fig.20 - Circuito del ponte H relativo all'integrato L293D.....	35
Fig.21 - Connettore ISP.....	39

AVVERTENZA!

AAR è un marchio registrato!

Tutti i diritti sono riservati.

Qualsiasi parziale o completa riproduzione del contenuto è vietata senza previa autorizzazione scritta dell'importatore Europeo:

AREXX Engineering - Zwolle (NL).

Le specifiche, la forma e il contenuto di questo prodotto sono soggette a modifiche senza preavviso.

Costruttore:

AREXX Engineering
Zwolle, The Netherlands

© 2015 AREXX Engineering

© Traduzione italiana:

AREXX - The Netherlands



Supporto tecnico:

www.arexx.com

www.roboternetz.de

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 ARDUINO. Questo prodotto non è un gioco; esso non è adatto a bambini al di sotto dei 14 anni!
Questo robot 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 del Robot AAR

1.1. La Famiglia ARDUINO Robotics

Arduino è una piattaforma open-source per lo sviluppo di prototipi elettronici. Questa piattaforma ci fornisce un microcontrollore con incluse le periferiche ed il software necessario. alla sua programmazione.

La piattaforma "Arduino" è stata progettata per imparare nel modo più semplice possibile la moderna elettronica per la robotica, la programmazione per il controllo di attuatori e l'uso di sensori.

Come successore del robot ASURO, che può essere programmato in linguaggio C, ora abbiamo progettato il robot Arduino.

Il nuovo robot assomiglia al suo predecessore ASURO, ma in combinazione con un linguaggio di programmazione "open source", la programmazione del sistema Arduino sarà molto più facile.

1.2. Caratteristiche:

Motori:	2 motori in Corrente Continua (3 Volt)
Processore tipo:	ATmega328P-AU
Linguaggio di programmazione:	ARDUINO
Alimentazione:	4 batterie tipo AAA 4,8 - 5,5 Volts
Corrente assorbita:	Min. 10 mA Max. 600 mA
Porta di comunicazione:	Cavo USB
Espansioni:	Espansioni compatibili ASURO
Altezza:	40 mm
Larghezza:	120 mm
Profondità:	180 mm



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 dovesse bagnarsi rimuovere immediatamente le batterie ed asciugare quanto prima le parti bagnate.
- * Rimuovere le batterie se ROBOT non verrà utilizzato per più di una settimana.

1.3. Precauzioni

1. Attenzione!

Prima di dare tensione a qualsiasi terminale devi leggere questo manuale. Collegamenti errati possono danneggiare l'hardware.

2. Attenzione!

Controlla con attenzione le funzioni di ogni connettore dello schema.

Controlla con attenzione il cablaggio del circuito elettrico.

Collegamenti errati possono danneggiare i moduli.

Rispetta la corretta polarità dei cavi d'alimentazione. Invertire i cavi di alimentazione danneggia l'hardware.

3. Attenzione!

Non usare un'alimentazione con voltaggio diverso da quello richiesto.

Usa un alimentatore stabilizzato e filtrato per non aver sovratensioni o disturbi.

4. Attenzione!

La scheda non fornisce alcuna protezione di impermeabilità o umidità.

Per piacere usa il sistema in ambiente asciutto.

5. Attenzione!

Evita cortocircuiti con superfici metalliche e non stressare la scheda elettronica o le spine con forza eccessiva o pesi.

6. Attenzione!

Fai attenzione alle scariche elettrostatiche ESD (leggi misure di prevenzione, precauzioni e descrizioni cercando l'argomento "Scariche elettrostatiche" su Wikipedia).

2. ARDUINO Descrizione Generale

2.1. Chi è e cosa fa ARDUINO?

Arduino è una scheda a microcontrollore tipo "open source", la quale fornisce un approccio facile alla programmazione, ai microcontrollori e ai progetti interattivi per artisti, disegnatori, hobbisti ed altri.

La piattaforma di Arduino è basata sui microcontrollori ATmega168 o ATmega328 di Atmel. Il sistema fornisce agli utenti porte di I/O digitali e canali di ingresso analogici i quali consentono al sistema Arduino di interagire con l'ambiente esterno.

Sul mercato possiamo trovare diverse schede Arduino così come "Arduino Uno", "Arduino LilyPad" e "Arduino Mega 2560". Ogni scheda Arduino è stata progettata per scopi specifici. Ovviamente gli utenti possono scegliere il tipo di scheda Arduino che più si adatta alle loro esigenze.

I segnali di ingresso possono essere forniti da interruttori, fotocellule, sensori di velocità e accelerazione, sensori di prossimità e di temperatura. Ulteriori comandi possono essere inviati anche tramite web-server. I segnali di uscita possono essere utilizzati per comandare motori, pompe e display grafici.

Il sistema è fornito di un compilatore per un linguaggio di programmazione standard e di un bootloader. Il linguaggio di programmazione è basato su "Wiring" il quale è simile al C++.

Arduino è un progetto nato ad Ivrea, in Italia, nell'anno 2005. Lo scopo principale è quello di supportare gli studenti in progetti per i quali la prototipizzazione dovrebbe essere poco costosa e molto efficiente a differenza della maggior parte dei metodi standard.

Il gruppo di sviluppatori, guidati da Massimo Banzi e David Cuartielles, ha deciso di chiamare il progetto "Arduino", nome del bar in cui il progetto è nato ma che a sua volta richiama il nome di di uno storico re "Arduin of Ivrea". "Arduino" significa anche "strong friend" o "grande amico".

Il nome della versione inglese è "Hardwin".

2.2. Microcontrollori!

2.2.1 Applicazioni

Un microcontrollore (a volte abbreviato in μC , uC o MCU) è un piccolo computer integrato in un piccolo chip il quale contiene il processore, la memoria e le periferiche programmabili di I/O.

Sul chip risiedono la memoria programma e una piccola quantità di memoria dati (RAM).

I microcontrollori sono utilizzati in prodotti e dispositivi controllati automaticamente, come sistemi di controllo motore delle automobili, dispositivi medici impiantabili, controlli remoti, macchine per ufficio, elettrodomestici, utensili elettrici e giochi. Riducendo le dimensioni ed il costo, rispetto ad un sistema a microprocessore con memoria e dispositivi di I/O separati, i microcontrollori rendono economico controllare digitalmente più dispositivi o processi.

Una casa tipica in un quartiere moderno può avere 4 microprocessori general-purpose e 3 dozzine di microcontrollori. Un'automobile di tipo medio ha almeno 30 o più microcontrollori.

I microcontrollori si trovano anche in lavatrici, forni a microonde e telefoni.

2.3. Consumo di energia e velocità

Alcuni microcontrollori, per bassi consumi energetici (milliwatts o microwatts), possono funzionare a frequenze di clock più basse di 4 KHz. Essi normalmente hanno la capacità di mantenere le loro funzionalità in attesa di un evento tipo la pressione di un pulsante o altro interrupt; il consumo di energia, durante la fase di sleeping (clock della CPU e diverse periferiche spente), può essere di qualche nanowatts, rendendo molti di questi microcontrollori adatti per programmi dove è importante il risparmio energetico per avere una lunga durata delle batterie. Altri microcontrollori sono utili in applicazioni dove le prestazioni sono elevate, ad esempio quando vengono utilizzati come processori di segnali digitali (DSP), e quindi alta velocità di clock ed elevato consumo.

Il sistema Arduino utilizza un potente single-chip ATmega328P di ATMEL, microcontrollore a 8-bit, 16 MHz di clock con 32Kbytes ed una flash programmabile in-system. La tensione di alimentazione è stata progettata in modo che possa variare nella gamma tra 7V e 12V DC, offrendo condizioni di funzionamento stabilizzate e protette per il chip, inoltre linee di potenza relative al circuito comando motore supportano correnti fino a 2 A.

2.4. Programmi per il Microcontrollore

Dal momento che una memoria programma esterna sarebbe troppo costosa, il microcontrollore ne possiede una internamente nella quale risiede il programma utile al suo funzionamento.

Compilatori ed assembler opportuni sono utilizzati per convertire un linguaggio ad alto livello ed un linguaggio assembler in un codice macchina compatto ed adatto per essere salvato nella memoria del microcontrollore.

In funzione del tipo di dispositivo utilizzato, la memoria programma può essere permanente o a sola lettura e perciò programmata dal costruttore, oppure memoria tipo flash che può essere modificata o memoria di sola lettura cancellabile.

In origine i microcontrollori erano programmati in linguaggio assembler, ora invece sono di uso comune linguaggi di programmazione ad alto livello per il microcontrollore in uso.

Questi linguaggi possono essere stati progettati per uno scopo specifico, oppure possono essere stati progettati per un uso più generico come il linguaggio di programmazione C.

I fornitori di microcontrollori spesso offrono tools di sviluppo di tipo gratuito in modo da utilizzare facilmente il loro hardware.

Il sistema Arduino fornisce circa 32Kbytes di memoria flash per i suoi programmi "sketches", i quali possono essere programmati in linguaggio C.

2.5. Architettura d'interfacciamento

Normalmente i microcontrollori contengono decine di ingressi e uscite di uso generico (GPIO). I pins di tipo GPIO possono essere configurati tramite software sia come ingressi che come uscite.

Quando i pins GPIO sono configurati come ingressi tramite essi è possibile leggere segnali provenienti da sensori o da segnali esterni. Configurati invece come uscite i pins GPIO possono pilotare dispositivi d'uscita come LED o motori.

Molti sistemi embedded hanno bisogno di leggere sensori che producono segnali analogici. Questo è lo scopo principale di un convertitore analogico/digitale (ADC). Dal momento che i processori sono costruiti per interpretare e processare dati digitali, ad esempio 1 (uno) o 0 (zero), essi non sono in grado di interpretare segnali che gli arrivano da dispositivi analogici come, ad esempio, sensori di temperatura o misura di tensioni variabili.

Affinchè il processore possa riconoscere i segnali appena detti vengono utilizzati allo scopo convertitori analogici/digitali.

Una funzione meno comune in alcuni microcontrollori è un convertitore digitale/analogico (DAC) che permette al processore di mettere sul pin d'uscita un segnale analogico o livelli di tensione variabili.

Oltre ai convertitori molti microcontrollori aggiungono alle loro funzioni anche un certo numero di timers. Uno dei timers più tipici è il Timer Programmabile (Programmable Interval Timer - PIT).

Un PIT decrementa il suo valore fino a zero. Appena raggiunge il valore di "zero" il timer manda un interrupt al processore per indicare che il conteggio è terminato. Questa funzione è utile per dispositivi come i termostati, i quali, ad intervalli regolari, verificano la temperatura ambiente per accertarsi se occorre accendere o spegnere l'aria condizionata, oppure se accendere o spegnere il riscaldamento.

La Universal Asynchronous Receiver/Transmitter (UART) permette di ricevere e trasmettere dati su una linea seriale senza sovraccaricare la CPU.

Altro hardware specifico, come la I2C o la SPI (Serial Peripheral Interface), offre la possibilità di comunicare con altri dispositivi (chips) in formato digitale.

Arduino è provvisto di 14 segnali di I/O e 7 segnali analogici.

3. ARDUINO ROBOT

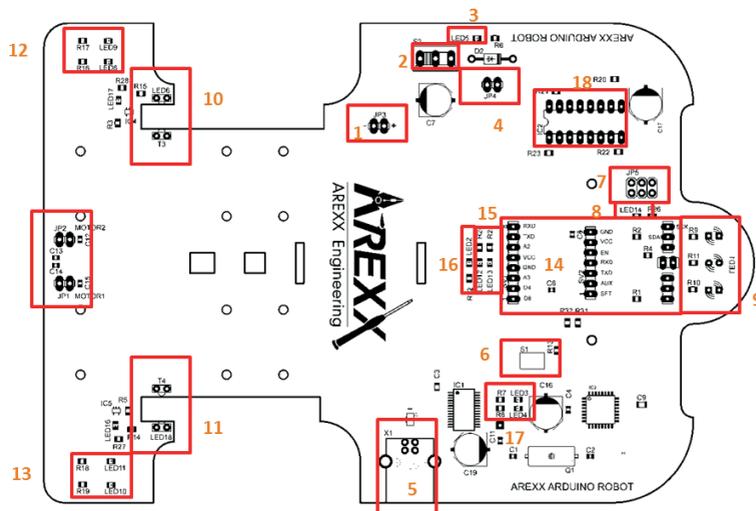


Fig.1 - AAR PCB

3.1. LEDS DI ARDUINO ROBOT

LED 1.	Red	Proiettore per controllo linea (sotto)
LED 2.	Blue	Liberamente programmabile
LED 3.	Red	USB DATA
LED 4.	Green	USB DATA
LED 5.	Blue	LED segnalazione alimentazione
LED 6.	IR	Proiettore infrarosso encoder ruota sinistra
LED 7.	NA	NA
LED 8.	Red	Rotazione avanti sinistra motore
LED 9.	Red	Rotazione indietro sinistra motore
LED 10.	Red	Rotazione indietro destra motore
LED 11.	Red	Rotazione avanti destra motore
LED 12.	Green	TX seriale
LED 13.	Red	RX seriale
LED 14.	Blue	Liberamente programmabile
LED 15.	NA	NA
LED 16.	Red	Encoder ruota Destra
LED 17.	Red	Encoder ruota sinistra
LED 18.	IR	Proiettore infrarosso encoder ruota destra

LED 14 Lampeggia durante il bootloader

LED 14 è OFF senza bootloader

LED 14 lampeggia lentamente quando si accende nel momento in cui un programma è caricato

3.2. Circuito stampato di ARDUINO ROBOT

Vedi le corrispondenze in fig.: AAR PCB

- 1) Connettore di collegamento del porta batteria.
(Fai attenzione alla corretta polarità!)
- 2) Interruttore di On/Off del robot.
- 3) LED di stato: segnala che il robot è alimentato da tensione esterna.
- 4) Nel caso in cui tu stia utilizzando delle batterie ricaricabili devi chiudere questo ponticello, in questo modo il robot sarà alimentato con un valore di tensione corretto.
- 5) Connettore USB: serve per programmare il robot utilizzando il software ARDUINO.
- 6) Pulsante di Reset: serve per resettare manualmente il robot.
- 7) Connettore ISP: permette di installare un altro programma bootloader.
- 8) LED 14: può essere liberamente usato dal programma utente, lampeggia durante la fase di bootloader.
- 9) Inseguitore di linea: questo modulo è liberamente programmabile e permette al robot di seguire una linea.
- 10) Sensore ruota sinistra: questo modulo genera una serie di impulsi proporzionali alla velocità di rotazione della ruota sinistra.
- 11) Sensore ruota destra: questo modulo genera una serie di impulsi proporzionali alla velocità di rotazione della ruota destra.
- 12) Leds di stato per motore lato sinistro: questi leds indicano rispettivamente la rotazione oraria ed antioraria del motore.
- 13) Leds di stato per motore lato destro: questi leds indicano rispettivamente la rotazione oraria ed antioraria del motore.
- 14) Connettore per scheda d'espansione: in questo connettore possono essere installati, per esempio, un modulo wireless APC220 oppure un modulo SnakeVision per poi essere collegati al sistema Arduino.
- 15) Led di stato per l'interfaccia di comunicazione RS232.
- 16) Led di stato LED 2: accessibile per la programmazione.
- 17) Leds di stato per porta di comunicazione for USB. Controllo motore.

Sul CD si trova il software di test di AAR AAR_SelftestV3.

Con questo programma puoi fare il test di tutte le funzioni base del robot AAR .

Usa il monitor seriale disponibile nel software di Arduino, non dimencarti di installare le librerie, vedi a pag. 22.

3.3. Informazioni di base per AAR

Nella parte frontale si trova l'interfaccia USB realizzata con un integrato tipo FT232. Quest'integrato converte il segnale USB in un segnale tipo RS232 il quale può essere elaborato dal processore ATMEGA328 (si trova nella parte destra della posizione frontale).

Sulla parte opposta si trovano l'interruttore ON/OFF, il connettore di alimentazione JP3 e l'integrato IC2 , controllo motore. Nella parte posteriore del circuito stampato (PCB) sono stati posizionati entrambi i motori ed i sensori di controllo delle ruote. I sensori delle ruote funzionano come fotocellule. Nelle ruote dentate sono stati eseguiti 4 fori disposti a 90° tra di loro. Non appena la luce attraversa un foro ed illumina un sensore, questo invierà un impulso di trigger al processore corrispondente alla sua ruota. Inoltre i circuiti elettronici accendono rispettivamente i led LED16 e LED17. Gli impulsi di trigger ci permettono di avere una visione accurata della velocità del motore per ciascuna ruota posteriore.

Nella parte frontale sono stati disposti i connettori per le schede d'espansione, mentre nella parte inferiore del PCB si trovano i sensori di inseguimento della linea.

La linea d'inseguimento utilizza un LED per inviare un fascio luminoso nell'area sottostante. Allineati al LED sono stati posizionati due sensori infrarosso per monitorare la luce riflessa dal pavimento. Inoltre sul PCB si trovano altri componenti (leds, resistori e capacità, per completare la linea d'inseguimento in un modulo funzionante.

Il robot usa una scheda Arduino la quale può essere paragonata alla scheda Arduino Duemilanove.

Il cuore del sistema è un microcontrollore ATMEGA328P, il quale fornisce 14 linee digitali di I/O, di queste 6 sono configurabili come canali di uscita tipo Pulse-Width-Modulated (PWM). Inoltre il robot è stato configurato con 6 canali analogici, un quarzo a 16 Mhz ed un connettore USB per la programmazione ed il controllo.

L'elenco può essere completato con un connettore ISP il quale permette ad hobbyisti esperti di programmare il loro programma personale di bootloader.

Il robot è stato disegnato per una tensione di alimentazione di 5 Volt la quale può essere prelevata anche dal connettore USB. Questa opzione è molto comoda durante la fase di test del programma.

Il robot è stato personalizzato con alcuni connettori molto comodi tramite i quali è possibile inserire moduli personali o altri della serie ASURO.

3.4. Informazioni di base per Software Arduino

Il software Arduino appartiene alla categoria Open Source ed è disponibile universalmente per tutti, inclusi i codici sorgenti della programmazione della piattaforma.

La piattaforma di programmazione di Arduino è stata fornita di un editor di testo, una finestra messaggi ed una console per il codice.

La piattaforma di programmazione può comunicare direttamente con AAR permettendo un facile trasferimento del programma all'interno del processore.

I programmi che sono stati scritti in linguaggio Arduino sono chiamati "sketches". Un normale editor di testo è utilizzato per sviluppare e scrivere questi programmi. Il files "sketches" saranno memorizzati nell'HardDisk del tuo PC. Il nome file degli Sketches è identificabile dall'estensione ".ino"

Le operazioni di salvataggio dei files "sketches" vengono visualizzate nella finestra dei messaggi. Questi messaggi possono riportare anche gli eventuali errori di stesura del codice sorgente.

La parte bassa di destra della finestra visualizza il tipo di scheda Arduino utilizzata e la porta seriale in uso.

L'architettura base di Arduino ci fornisce anche librerie contenenti funzioni extra. Una libreria è un insieme di funzioni predefinite le quali possono essere riutilizzate in parti ricorrenti di un programma senza aumentare i costi di sviluppo.

Un programma di Arduino può essere strutturato in tre sezioni:

- struttura base,
- definizioni (per le rispettive variabili),
- funzioni.

La struttura base di Arduino consiste di una funzione di "setup" e di una di "loop".

La funzione di "setup" è utilizzata per inizializzare le variabili, la definizione dei pins („Pin-Modes") e la definizioni delle librerie.

La funzione di "loop" rappresenta un ciclo senza fine nel quale il programma cicla infinitamente fino a quando non viene spento il sistema.

Il programma usa "variabili modificabili" all'interno delle quali vengono memorizzati valori che possono essere modificati dal programma oppure "variabili costanti" il cui valore non cambia, ad esempio la definizione dei pin (ingresso od uscita) o il livello di tensione presente sui pin di collegamento.

4. Si parte!

4.1. Scaricare ed installare il software di Arduino

Installa il software di Arduino (versione 1) presente sul CD, siamo sicuri che funziona. In seguito potrai andare sul sito di Arduino e scaricare l'ultima versione disponibile.



IMPORTANTE:

Usando versioni diverse di Arduino o di applicazioni si potrebbero avere problemi.

Alcune volte, dopo aver aggiornato il software di Arduino, occorre modificare il proprio applicativo altrimenti potrebbe non funzionare!

4.2. Linguaggio di Arduino

La sintassi del linguaggio utilizzato con Arduino è documentata sul suo sito ufficiale. Cerca di capire le peculiarità del linguaggio in funzione di ciò che ti serve.

4.3. Installazione del driver USB

Quando colleghi per la prima volta la scheda, Windows dovrebbe iniziare il processo di installazione dei driver.

In sistemi da Windows Vista e più recenti i driver dovrebbero essere scaricati ed installati in modo automatico.

Dalla barra menù scegli "Strumenti -> Porta" per impostare la porta di comunicazione seriale. Questa potrebbe essere la COM3 o porte con valore più alto (la COM1 e la COM2 sono di fatto riservate alle porte seriali hardware).

Per scoprire la porta corretta puoi scollegare la tua scheda e riaprire il menù; la voce che scompare dal menù è quella relativa alla scheda Arduino.

Ricollega la scheda e scegli la porta seriale.

4.4. Installazione delle batterie

Il robot è stato disegnato per essere alimentato con 4 batterie da 1.5 Volt tipo AAA. Se preferisci utilizzare batterie ricaricabili devi chiudere il jumper JP4. Le batterie ricaricabili hanno infatti una tensione inferiore rispetto a quelle alcaline (1.2 Volt)(vedi fig. 1, numero 4).



ATTENZIONE!

L'inserzione del jumper JP4 elimina il diodo che previene l'inversione di polarità dell'alimentatore.

Quando è inserito il jumper JP4 errori di collegamento dell'alimentazione, come l'inversione di polarità, danneggiano il robot..

Collega il portabatteria come mostrato in figura Fig.2

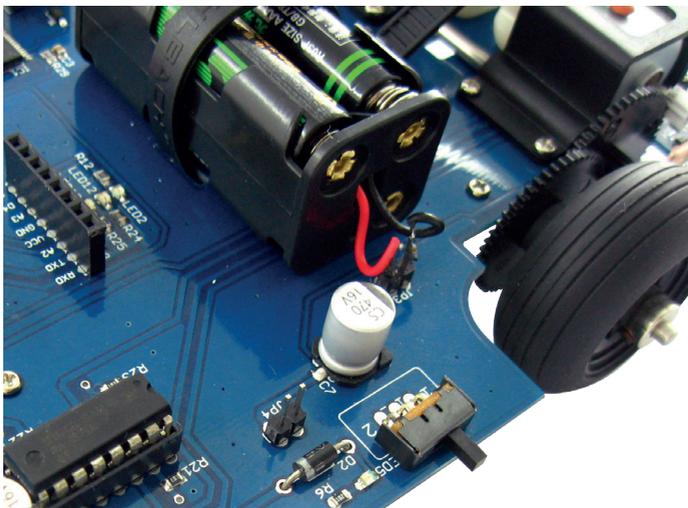


Fig.2 - Collegamento del porta batteria

Ora puoi accendere il robot tramite l'interruttore ON/OFF.

Il led (LED5), posto vicino all'interruttore, che indica la presenza di tensione, si accenderà.

4.5. Programmazione del Robot con Programmi Arduino.

Collega il robot al PC tramite il cavo USB.

Collegando il robot ad una porta USB il sistema Arduino non ha bisogno di una batteria extra o di un alimentatore

La tensione richiesta verrà fornita dal connettore USB del PC.



ATTENZIONE:

Il robot si attiva sempre nel momento in cui viene collegato al PC. L'interruttore di alimentazione ON/OFF ed il LED5 funzionano solo nel caso in cui siano utilizzate le batterie.

Ora puoi eseguire il software di Arduino (vedi fig. 3).

Compila il codice e verifica che sia corretto.

Invia il file compilato al robot

Nuovo sketch

Apri sketch

Salva sketch

Finestra di editor

Area messaggi
Console testi.
In questa finestra vengono visualizzati messaggi di errore o di stato

Area di visualizzazione scheda e COM utilizzate

```
sketch_jul01a | Arduino 1.6.5
File Modifica Sketch Strumenti Aiuto
sketch_jul01a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino Duemilanove or Diecimila, ATmega328 on COM10

Fig.3 - IDE di Arduino

4.6. Selezionare un programma Arduino

Partiamo ora con il caricare nel robot un semplice programma chiamato "Blink".

Questo programma farà lampeggiare continuamente il led LED14 del robot.

Carica il programma cercandolo nel software Arduino tramite il comando di menù:

File>Esempi>01.Basics>Blink

vedi Fig.4

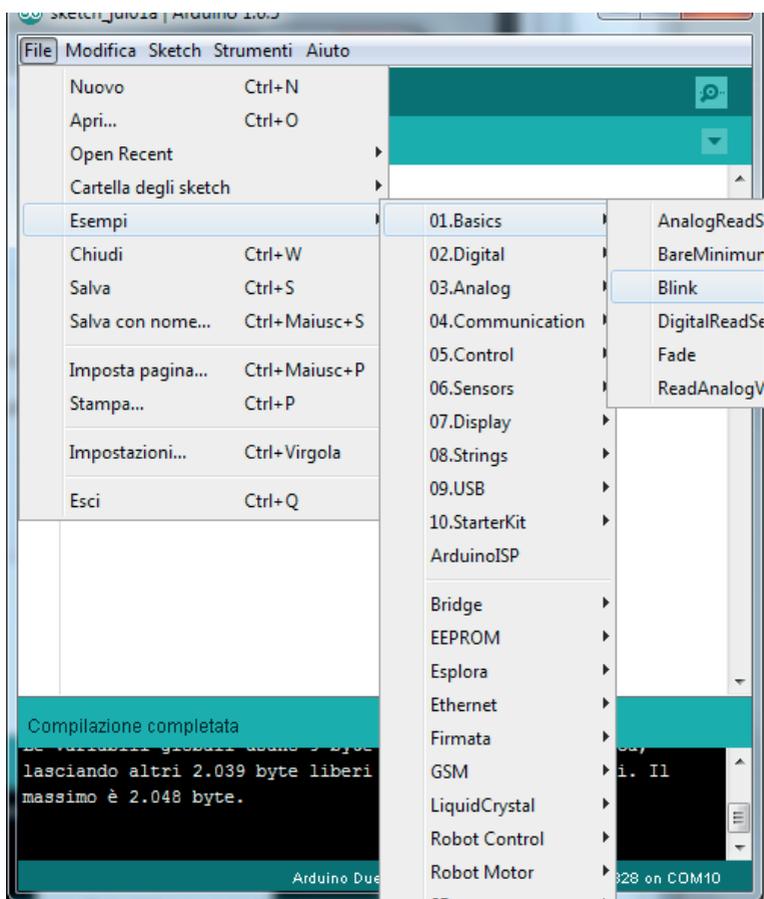
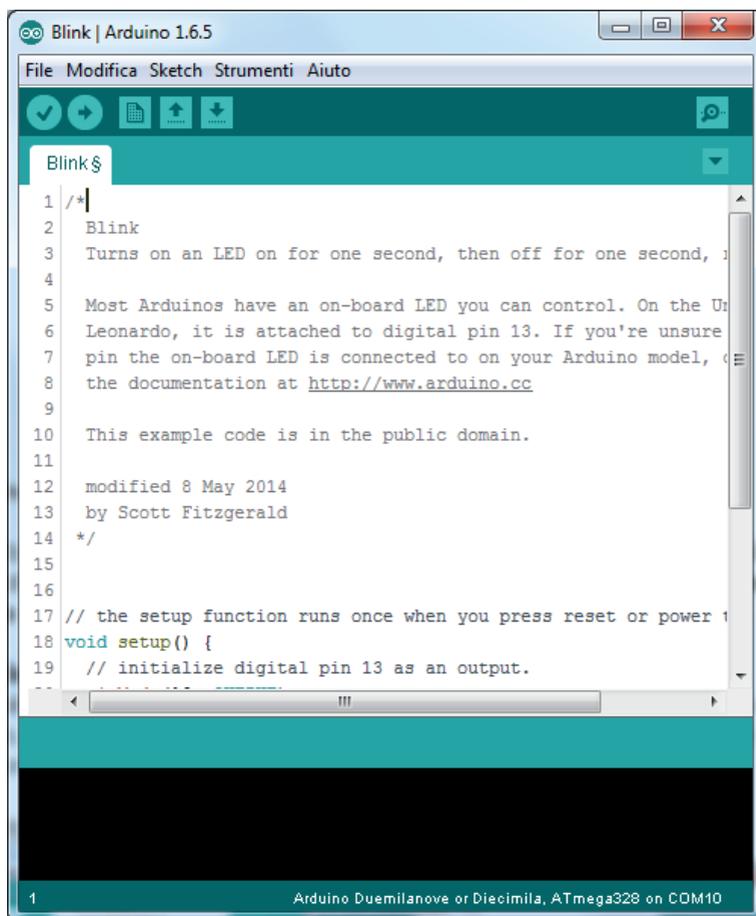


Fig.4 - Apertura Sketch "Blink"

si aprirà una finestra simile a quella di Fig.5



The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.6.5". The menu bar includes "File", "Modifica", "Sketch", "Strumenti", and "Aiuto". The toolbar contains icons for saving, undo, redo, and other functions. The main text area displays the following code:

```
1  /*  
2   Blink  
3   Turns on an LED on for one second, then off for one second, ...  
4  
5   Most Arduinos have an on-board LED you can control. On the Uno  
6   Leonardo, it is attached to digital pin 13. If you're unsure  
7   pin the on-board LED is connected to on your Arduino model, check  
8   the documentation at http://www.arduino.cc  
9  
10  This example code is in the public domain.  
11  
12  modified 8 May 2014  
13  by Scott Fitzgerald  
14  */  
15  
16  
17  // the setup function runs once when you press reset or power up  
18  void setup() {  
19    // initialize digital pin 13 as an output.  
20    pinMode(13, OUTPUT);  
21  }  
22  
23  // the loop function runs over and over again forever  
24  void loop() {  
25    digitalWrite(13, HIGH);   // turn the LED on (HIGH is the positive voltage)  
26    delay(1000);              // wait for a second  
27    digitalWrite(13, LOW);    // turn the LED off by making the pin LOW (no voltage)  
28    delay(1000);              // wait for a second  
29  }  
30
```

The status bar at the bottom indicates "1 Arduino Duemilanove or Diecimila, ATmega328 on COM10".

Fig.5 - Listato programma "Blink"

In questa pagina ora sarà possibile editare o modificare lo sketch appena caricato.

4.7. Selezionare la scheda Arduino

A questo punto occorre selezionare il tipo di scheda della serie Arduino corrispondente a quella che si trova sul robot.

Sul robot si trova una scheda simile a:

"Arduino Duemilanove con processore ATmega328"

oppure

"Nanow/Atmega328"

Dalla barra strumenti selezionare:

Strumenti-->Scheda->Arduino Duemilanove

come in Fig.6

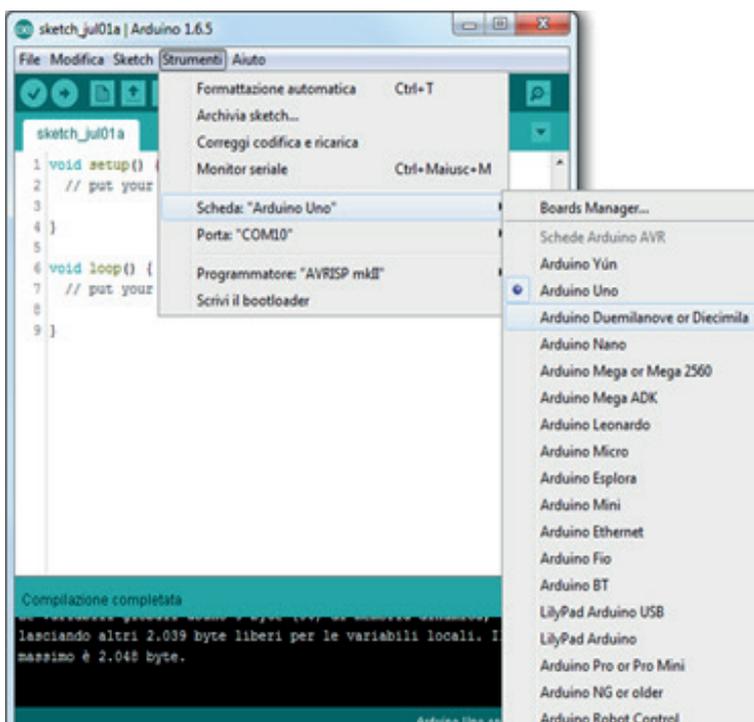


Fig.6 - Selezione scheda Arduino per robot AAR

4.8. Selezione della Porta Seriale

A questo punto occorre definire la porta COM corretta per la comunicazione tra PC ed il robot.

L'interfaccia COM corretta (o porta COM) normalmente è maggiore di COM3 (COM1 e COM2 sono già in uso).

Per selezionare l'interfaccia COM apri il menù alla voce:

Strumenti>Porta>COMx

(vedi fig. 7)

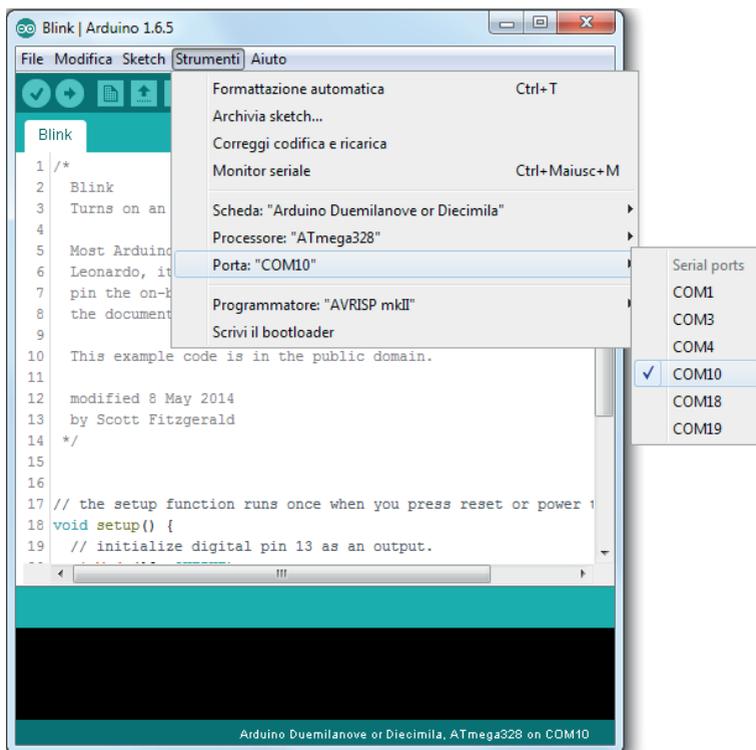


Fig.7 - Selezione porta COM

In questo caso la porta com utilizzata è la COM10.

4.9. Compilare lo Sketch

Dopo aver terminato di scrivere il programma occorre verificare se non ci sono errori di compilazione o di sintassi.

Occorre ricordare che il compilatore non corregge errori comportamentali o malfuzionamento.

Per eseguire la verifica occorre fare click sul pulsante di verifica oppure dalla barra strumenti scegliere:

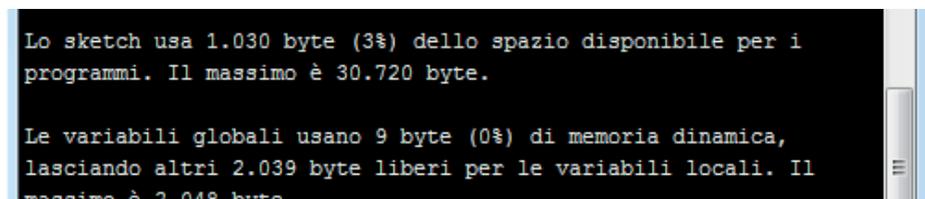
Sketch -> Verifica e Compila

vedi Fig.8



Fig.8 - Operazione di verifica dello Sketch

Se la verifica termina senza errori verrà visualizzato un messaggio come in Fig.9



4.10. Caricamento dello Sketch nel robot

Se la verifica è andata a buon fine (Fig.9) allora si può trasferire lo sketch al robot facendo click con il pulsante del mouse sul pulsante di invio oppure dalla barra strumenti occorre scegliere:

Sketch ->Carica

vedi Fig.10

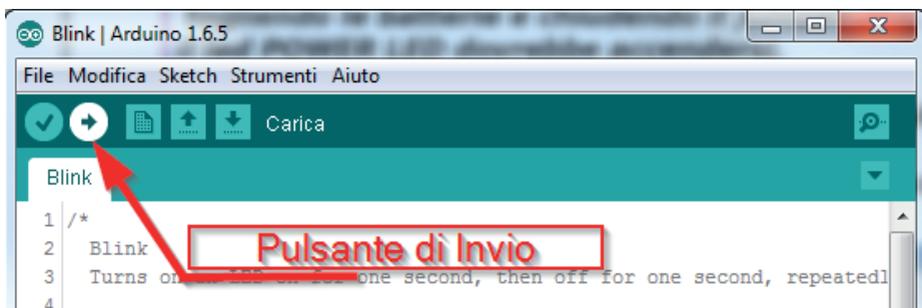


Fig.10 - Pulsante di invio dello sketch nel robot

Nel momento in cui inizierà il trasferimento dello sketch dal PC al Robot, nella finestra messaggi, Fig.11, verranno visualizzate alcune informazioni e verrà visualizzata una barra di colore verde che indica il susseguirsi delle fasi di caricamento.



Fig.11 - Fase di trasferimento dello sketch

Terminato il trasferimento dello Sketch, nella console testi verrà visualizzato un messaggio di trasferimento completato ed andato a buon fine, Fig.12.

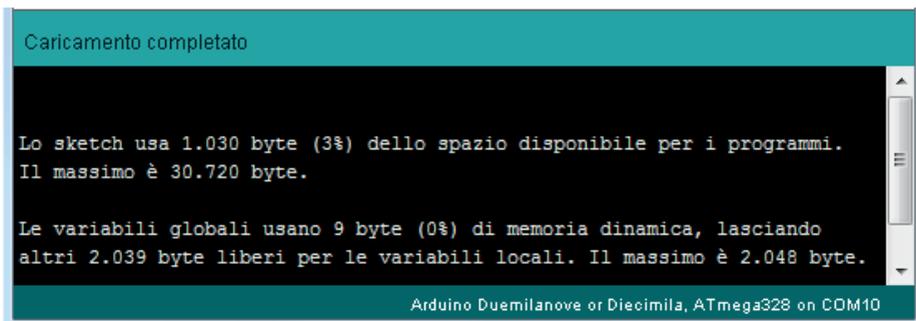


Fig.12 - Caricamento dello sketch completato

A questo punto il LED14 inizierà a lampeggiare lentamente.

Ora è possibile scollegare il robot dal PC, rimuovere il cavo USB, collegare il porta batteria, facendo attenzione alla corretta polarizzazione del connettore, ed avviare il robot.

Per ulteriori informazioni e per scaricare gli aggiornamenti ti consigliamo di visitare uno dei forum all'indirizzo internet:

www.arexxx.com
www.roboternetz.de

4.11. Risoluzione dei problemi

In caso di problemi di mal funzionamento occorre effettuare alcune verifiche.

- 1) Verificare che la tensione di alimentazione proveniente dalla USB sia sufficiente per alimentare il robot. Questo test può essere eseguito togliendo le batterie e chiudendo il jumper JP4, il led POWER LED dovrebbe accendersi.



Fig.13 - Gestione dispositivi

Problema: nessun collegamento con il Robot.

Verificare che i leds LED3 e LED4 lampeggino lentamente quando si collega il robot con il PC tramite USB

Tramite il pannello di controllo (Fig. 13) verifica che il robot sia collegato con il PC.

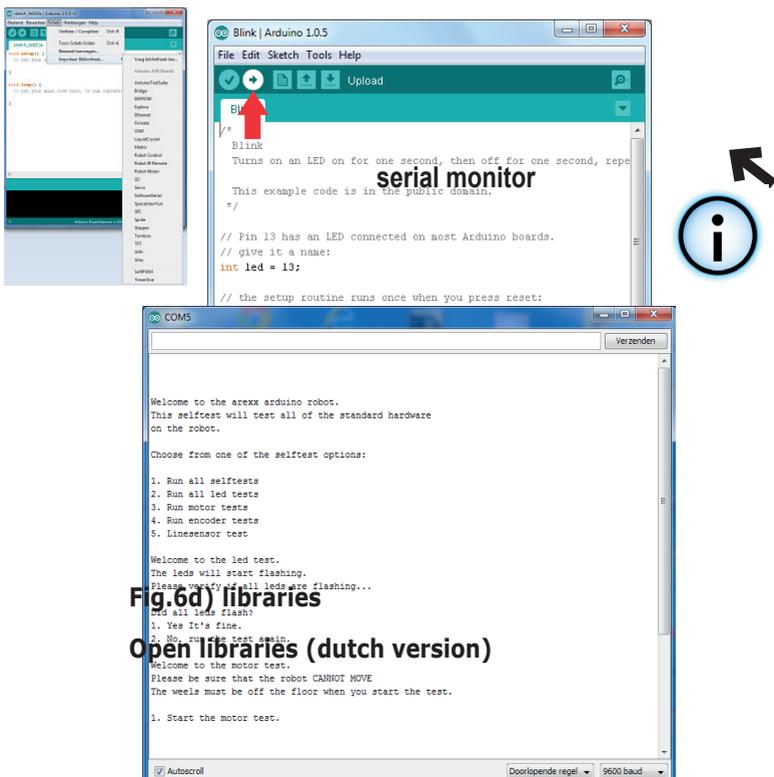
Il collegamento è effettuato tramite convertitore usb-seriale della FTDI.

Accertati che siano stati installati i driver corretti. Questa connessione, nel pannello di controllo, è sotto la voce "**USB SERIAL PORT**". Essendo questo uno dei problemi principali, scarica dal sito della FTDI i drivers più recenti.

Problema: non ci si riesce a caricare nel robot il file .INO

Verifica di aver scelto la COM corretta, normalmente questa è diversa dalla COM1 e dalla COM3. Verifica anche di aver scelto la corretta scheda Arduino ed il corretto processore (ATmega328).

Verifica di aver installato per il tuo sketch le librerie corrette, dalla barra menù scegli "**Sketch-->Include Library -->**".



4.12. Autotest della scheda.

Dal menu comandi scegli: **"File-->Apri"**

e seleziona lo sketch che si trova su CD:

"Examples--> Selftestv1_0-->Selftestv1_0.ino"

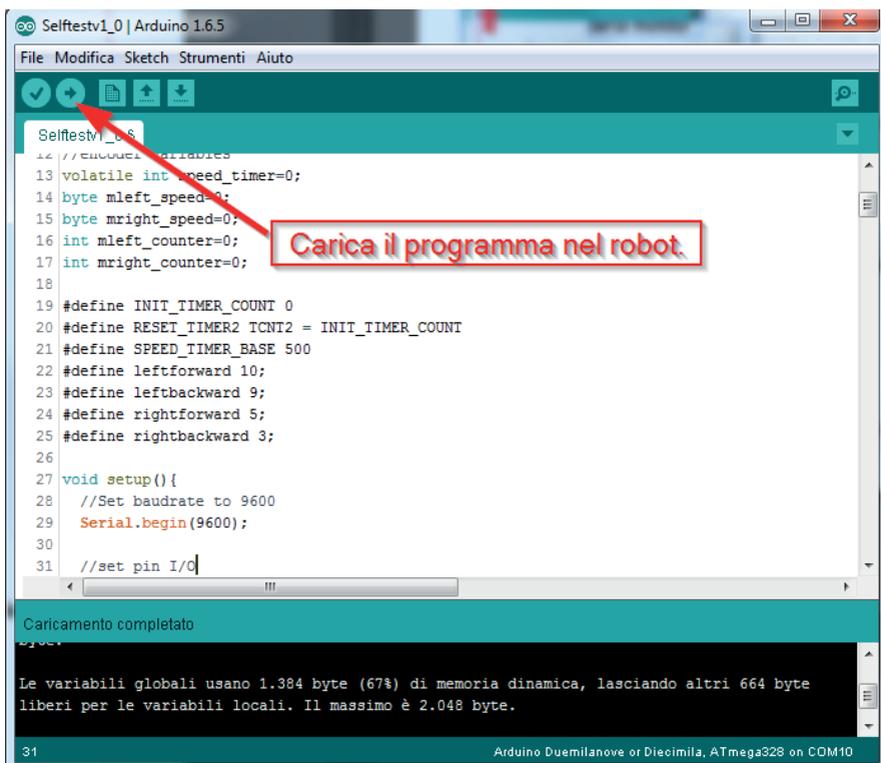


Fig.14 - Carica lo sketch nel robot

Fai click sul pulsante "carica" indicato dalla freccia in Fig.14 oppure nel menù comandi seleziona:

"Sketch --> Carica"

In questo modo, se non ci sono errori di compilazione e se il robot è collegato al pc, lo sketch verrà caricato nella memoria del robot.

Nella barra dei menù seleziona: "**Strumenti --> Monitor seriale**"
si aprirà la seguente pagina:

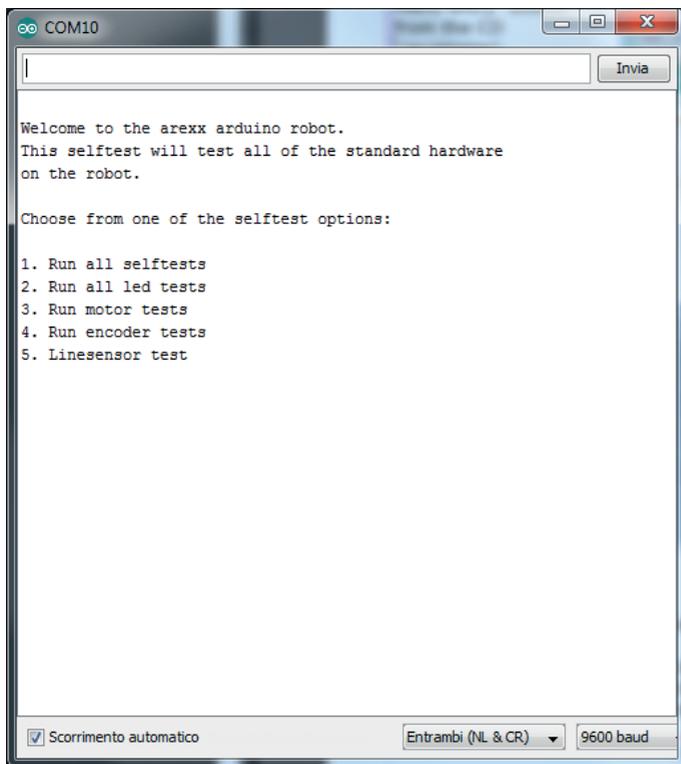


Fig.15 - Finestra terminale di Arduino

La finestra di Fig.15 rappresenta il terminale di Arduino.

Da questa finestra è possibile inviare stringhe seriali ad Arduino e dallo stesso riceverne.

In questo caso all'interno della finestra sono visibili 5 comandi di test che possono essere eseguiti per verificare l'hardware del robot.

Selezionando un comando alla volta è possibile eseguire il test della funzione descritta nella riga di comando.

5. Kits di espansione per scheda AAR

5.1. Kits ASURO di espansione per Arduino Robot.

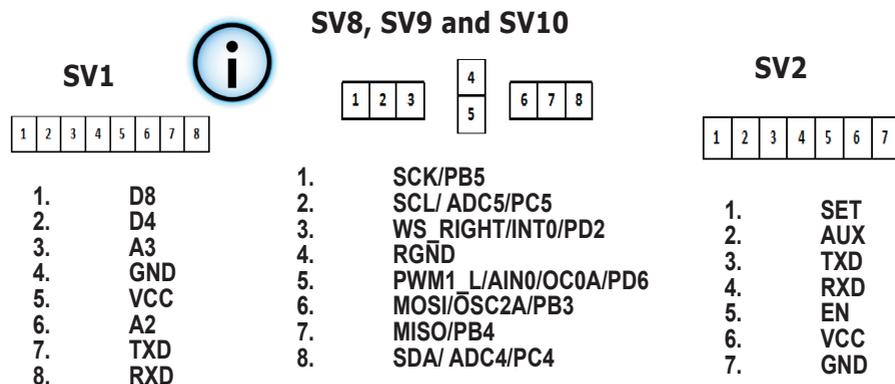
Sulla scheda AAR si trovano alcuni connettori per moduli di espansione. Questi connettori sono compatibili con quelli del robot ASURO di AREXX (un kit robotico che può essere programmato in linguaggio C). Questo significa che è possibile utilizzare per AAR tutti i kits di AREXX ASURO.

Sul CD si trovano alcuni programmi di esempio per Arduino, in questo modo è possibile utilizzare direttamente i kit di espansione.

Il robot ASURO ha anche una scheda generica per esperimenti (ARX-EXP2) che può essere utilizzata con le tue schede Arduino.

Qui sotto si trova il layout dei pin dei connettori per le espansioni per AAR.

Questi sono i connettori sulla scheda denominati SV8, SV9 e SV10.



ARX-EXP2

Scheda per esperimenti di AREXX.

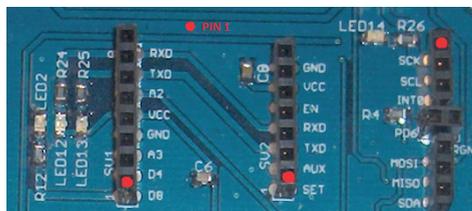
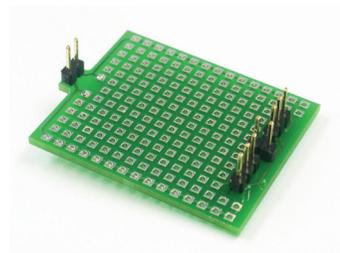


Fig.16 - Connettori per le espansioni

Il punto rosso è il pin 1

6. Configurazione Kits di Espansione di ASURO.

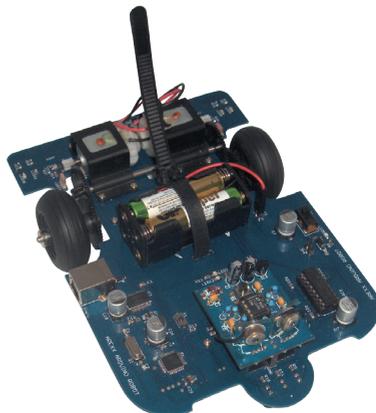
ARX-ULT10

Kit ultrasuoni AREXX

Utilizza il programma di esempio che si trova sul CD:

Examples\AAR_ULTRA\AAR_ULTRA\ AAR_ULTRA.ino

Con questo programma di esempio il robot arretra quando incontra un ostacolo.



ARX-SNK20

AREXX SNAKE VISION (PYRO) KIT

Utilizza il programma di esempio che si trova sul CD:

Examples\AAR_SNAKE\AAR_SNAKE\AAR_SNAKE.ino

Con quest'esempio il robot insegue corpi caldi.

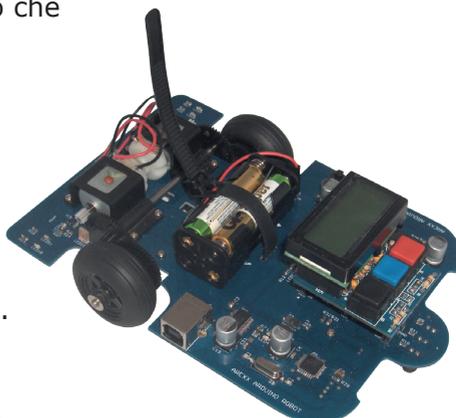
ARX-DSP30

AREXX DISPLAY KIT

Utilizza il programma di esempio che si trova sul CD:

Examples\AAR_DISPLAY\AAR_DISPLAY.ino

Quest'esempio visualizza messaggi sul display, i pulsanti sono liberamente programmabili.



ARX-MNSP55

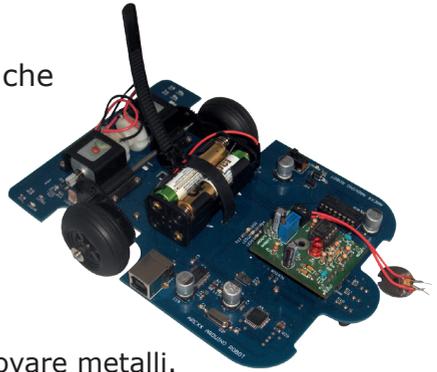
Kit cerca metalli AREXX

Utilizza i programmi di esempio che si trovano su CD:

**Examples\AAR_MINE\
AAR_MINE1\AAR_MINE1.ino**
oppure

**Examples\AAR_MINE\
AAR_MINE2\AAR_MINE2.ino**

Con questo esempio il robot può trovare metalli.



ARX-APC220

AREXX WIRELESS KIT

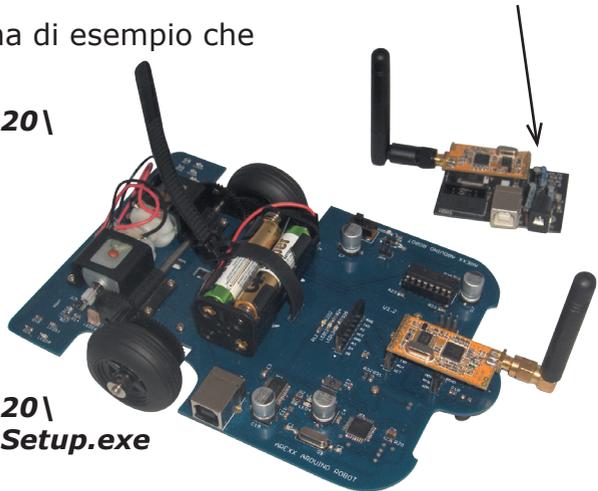
Utilizza il programma di esempio che si trova su CD:

**Examples\AAR_APC220\
AAR_INO_SRC\
AAR_INO_SRC.ino**

Sul PC devi installare il programma Visual Basic:

**Examples\AAR_APC220\
AAR_VB_Software\Setup.exe**

Fig.17 - RP6v2 Programmatore USB con APC 220



Con il set AREXX ARX-APC-220 è possibile controllare il robot AAR tramite connessione senza fili (wireless).

Di seguito la descrizione passo passo del funzionamento:

- 1)Collega al PC il programmatore USBR P6v2 ed il modulo APC-220.
- 2)Collega l'altro modulo APC ad AAR (connettore di mezzo).
- 3)Carica il programma AAR_SRC nel processore di AAR.
- 4)Installa il software AAR_VB sul tuo PC.
- 5)- Seleziona la porta COM corretta nel software AAR-VB.

7. Programma ANDROID

ARX-BT3

AREXX BLUETOOTH KIT

Il nostro modulo ARX-BT03 permette di controllare il robot AAR tramite Bluetooth utilizzando un'applicazione ANDROID.

Dal CD AREXX_AAR_0214 apri la cartella:

Android_App\AAR_APP

ed installa nel tuo dispositivo il programma:

BLUETOOTH_AAR.APK

subito dopo installa sul robot il programma:

AAR_SRC_Bluetooth\AAR_SRC_Bluetooth.ino

Segui passo dopo passo le istruzioni seguenti:

- 1) collega il modulo Bluetooth sullo stampato del robot;
- 2) carica il programma ANDROID Bluetooth nel processore del robot;
- 3) carica l'applicazione ANDROID APK nel tuo smartphone o tablet.



Bluetooth®



Puoi trovare le applicazioni ANDROID della AREXX sul CD nella cartella:

Android_App\AAR_APP

oppure su

GOOGLE PLAY.



Circuiti elettronici tipo Ponte H

Un Ponte H è un circuito elettronico che permette di invertire la polarità di un dispositivo (come, ad esempio, un motore in Corrente Continua) tramite quattro interruttori.

Questi Ponti-H vengono spesso utilizzati nei robots per controllare i due sensi di rotazione del motore.

I sistemi moderni, per il controllo dei motori, utilizzano circuiti integrati, tuttavia per imparare le basi di funzionamento e le problematiche di dimensionamento dell'alimentazione occorre prima capire il circuito base di controllo di un motore.

7.1. Ponte H alimentato a 3 Volt

Il circuito elettrico per pilotare i robots tipo Hyper Peppy è composto da 2 transistor PNP (TR7 e TR8) e da altri 2 transistor tipo NPN (TR9 e TR10). In questo circuito solo 2 transistor possono funzionare simultaneamente per alimentare il motore M, per questo motivo la corrente può circolare contemporaneamente:

o tra TR7 e TR10 oppure tra TR8 e TR9.

La versione Free del software di simulazione Microcap ci permette di simulare il circuito e leggere i risultati nello schema della finestra sotto:

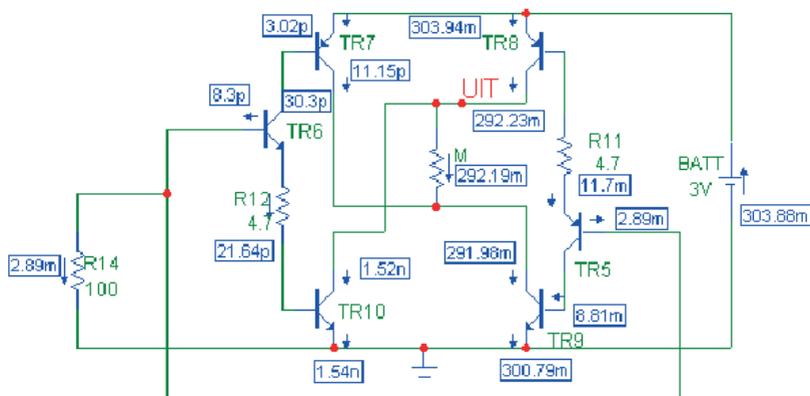


Fig.18 - Simulazione di un ponte H in un robot Hyper Peppy.

Nello stadio pilota è possibile identificare il motore in corrente continua M. Il circuito di preamplificazione del driver è simulato dalla resistenza R14. Questa resistenza mantiene a 0V le basi dei transistor TR6 e TR5, e questa è la condizione per la quale solo la parte di destra del circuito conduce corrente.

In questa situazione solo i transistor TR8, TR5 e TR9 conducono corrente mentre tutti gli altri sono interdetti. Non appena la resistenza R14 viene collegata ad una tensione positiva allora si metterà a condurre il ramo di sinistra mentre il ramo di destra verrà interdetto ed il motore invertirà la sua rotazione. Il programma di simulazione Microcap permette di simulare e calcolare le correnti di tutti i componenti e di leggere i valori sullo schema elettrico. Con una tensione di alimentazione di 3 Volt l'assorbimento sarà di circa 300mA.

La possibilità di utilizzare una bassa tensione di alimentazione è dovuta all'impiego di transistor al silicio di tipo NPN e PNP. Questi transistor possono lavorare con cadute di tensione di 0,7 Volt. Il motore però è stato inserito tra due collettori i quali, in saturazione, hanno una caduta di soli 0.3V. Questi interruttori forniscono al motore M una rispettabile tensione di 1.5V. I valori calcolati da Microcap possono essere letti dalla Fig. 18.

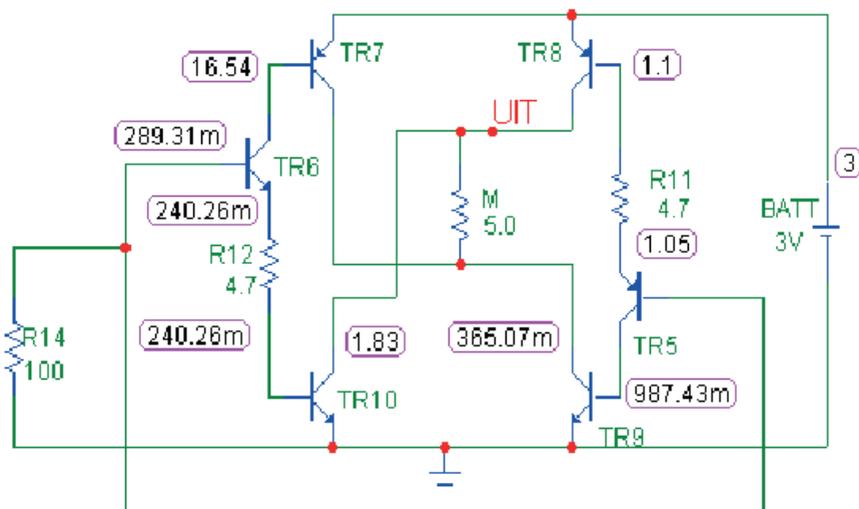


Fig.19 - Impostazioni DC per un ponte H di tipo Peppy Robot

I 3 volt di alimentazione sono la condizione ideale per un robot alimentato da un pacco batteria di sole due celle. I transistor PNP purtroppo non possono essere integrati in un chip come il L293D. Un integrato ha altri vantaggi come l'affidabilità, la protezione contro un cablaggio fatto male ed inoltre riduce le dimensioni ed il peso.

Per questa ragione abbiamo deciso di usare l'integrato L293D il quale racchiude al suo interno un doppio ponte H in grado di controllare contemporaneamente due motori DC.

7.2. Un ponte H a 4,5 Volt

L'integrato L293D (see fig. 20) ci permette di controllare correnti di uscita fino a 600mA per canale (massimo: 1.2A di corrente di picco)

Dal momento che la tensione di alimentazione del driver (VCC2) può variare tra 4.5V e 36V, quest'ora si che l'integrato L293D possa essere il favorito in circuiti di controllo per motori in DC.

La tensione minima di alimentazione (VCC2) tuttavia è stata scelta essere 4.5V, obbligandoci a scegliere un minimo di 4 batterie ricaricabili da usare come alimentatore. Questa scelta però aumenta il peso del robot, ma questo è il prezzo che si paga nei moderni circuiti elettronici.

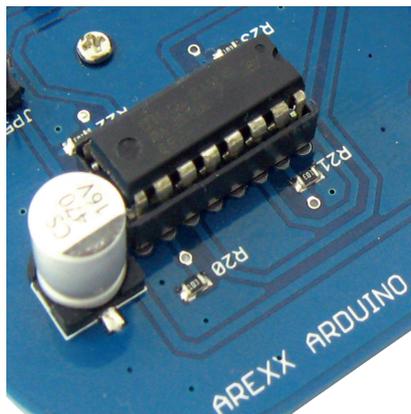


Fig.20 - Circuito del ponte H relativo all'integrato L293D

8. Odometria

Questo capitolo è stato destinato ad interessanti contetti applicativi del robot AAR. Le idee si riferiscono a studi ed a progetti specifici.

Probabilmente lo sviluppo di tale software ci appassionerà alla programmazione dei microcontrollori.

8.1. Ricercatori di linea, amanti e nemici dei colori

Sensori sensibili alla luce ci permetteranno di programmare i robots come ricercatori di linea, amanti o nemici dei colori.

Nel primo di questi esempi il robot si aspetta di seguire una linea in una curva ad 8 la quale, in un ciclo infinito, forza il robot a girare intorno eternamente.

Il secondo ed il terzo esempio insegnano al robot ad evitare le fonti di luce rossa ma essere, nello stesso tempo, attratto da una luce verde. Questi tipi di modelli comportamentali sono già considerati come pratiche di sopravvivenza per semplici organismi viventi come i vermi.

8.2. Paure ed amori per la musica

Molto interessante è anche un modello che dipende dal rumore ambientale. Ad un robot dotato di microfono si può insegnare ad evitare suoni con toni bassi ma nello stesso tempo gli si può insegnare ad essere attratto dalla musica di un flauto la quale contiene toni alti.

Essere attratto dalle alte frequenze può permettere di dimenticare la paura per le frequenze più basse. In questo modo il robot può essere costretto ad essere attratto dalla sorgente delle frequenze alte di un flauto nonostante sia circondato da musica heavy-metal.

Comportamenti che dipendono da frequenze alte e basse o luce e colori hanno bisogno di alcuni sensori, due filtri di frequenza ed una coppia di sensori di luce dipendente dotati di filtri colorati.

8.3. Ricercatori di linea complessi

I robots ricercatori ed inseguitori di linea normalmente hanno bisogno di una sorgente luminosa come un LED e di una o più coppie di sensori fotocellule. Questi dispositivi permettono al sistema di identificare una linea e di seguirne la traccia. Inizialmente il robot abbisogna di una speciale routine di ricerca per trovare la linea.

L'algoritmo può consistere nel seguire un percorso a spirale con un raggio che aumenta progressivamente rispetto al punto di partenza.

Questo algoritmo viene eseguito fino a quando il sensore non rileva una linea particolare ed inizia a seguirla.

Questo tipo di programma, che risolve in modo adeguato il problema di ricerca di qualsiasi tipo di linea, fa già parte di diversi algoritmi di programmi sofisticati.

8.3.1 Algoritmi complessi (esempio di programmazione)

Il progetto lo si può complicare specificando la strategia di ricerca per lavorare in un'area piena di linee colorate e ricercare la linea rossa che guiderà il robot verso un sicuro e buio "garage".

Non appena questo disturbo sparisce il robot può uscire dal "garage" per cercare una linea verde la quale guiderà il robot verso un altro "garage" con luce brillante e verde nel quale si troverà a proprio agio - anche se circondato da musica assordante.

Non appena la musica sarà composta da alte frequenze di flauto il robot si sentirà a disagio. A questo punto abbandonerà il suo posto auto coperto alla ricerca delle linee rosse che lo guideranno verso il suo buio "garage".

I programmatori esperti sono consapevoli della complessità e dei requisiti che richiede un programma che deve risolvere i problemi di un robot che insegue una linea che dipende da colori e suoni.

I programmatori dovranno allora scrivere un programma di tipo gerarchico dipendente da numerose funzioni.

Il concetto di programmazione modulare e strutturata ci permette di scrivere un software complesso, ma affidabile, in grado di risolvere l'attività richiesta.

La complessità del software possa ispirare i programmatori a provare ammirazione per i minuscoli organismi viventi, i quali sono la combinazione di questi modelli di comportamento della ricerca di cibo e della strategia di successo per la procreazione.

E' il grande sforzo della natura quello di cercare di migliorare continuamente i modelli comportamentali per tenere in vita la vita...



9. Programmare un Bootloader



ATTENZIONE!

Per eseguire in modo corretto quanto descritto da questo capitolo occorre esperienza di programmazione!

Sarai in grado di trasferire un Arduino-bootloader nel microcontrollore, per esempio, con l'aiuto della scheda STK500.

Per poter trasferire qualsiasi programma, scritto in ambiente di linguaggio Arduino, nel microcontrollore Atmega questo deve essere stato precedentemente programmato con un programma speciale chiamato "Bootloader".

Questo Bootloader è necessario per caricare nella memoria di Atmega il programma appena compilato.



Per programmare il Bootloader occorrono i seguenti componenti:

- * una scheda di programmazione AVR (per esempio la STK500)
- * un alimentatore a 12 Volt (per la STK500)
- * il robot AAR con a bordo un connettore ISP (fig. 21)
- * un PC dotato di una porta seriale fisica. Preferibilmente non un convertitore USB-RS232 che spesso danno errori di timeout.
- * Installa la versione corrente (ed eventualmente l'aggiornamento) dell'ambiente di Arduino che si trova all'indirizzo www.arduino.cc. Probabilmente il file scaricato sarà compresso (.zip o .rar).
- * Scompatta questo file ed il contenuto salvalo sull' hard disc. Utilizza, per esempio, WINAVR per trasferire il bootloader al robot.

Attenzione!

Il software di Arduino appartiene alla categoria di software freeware e, a volte, può succedere che l'ambiente Arduino ed il bootloader non siano tra loro compatibili!



Fig.21 - Connettore ISP

In caso di problemi simili sei invitato a visitare il sito ed il forum di Arduino!

APPENDICE

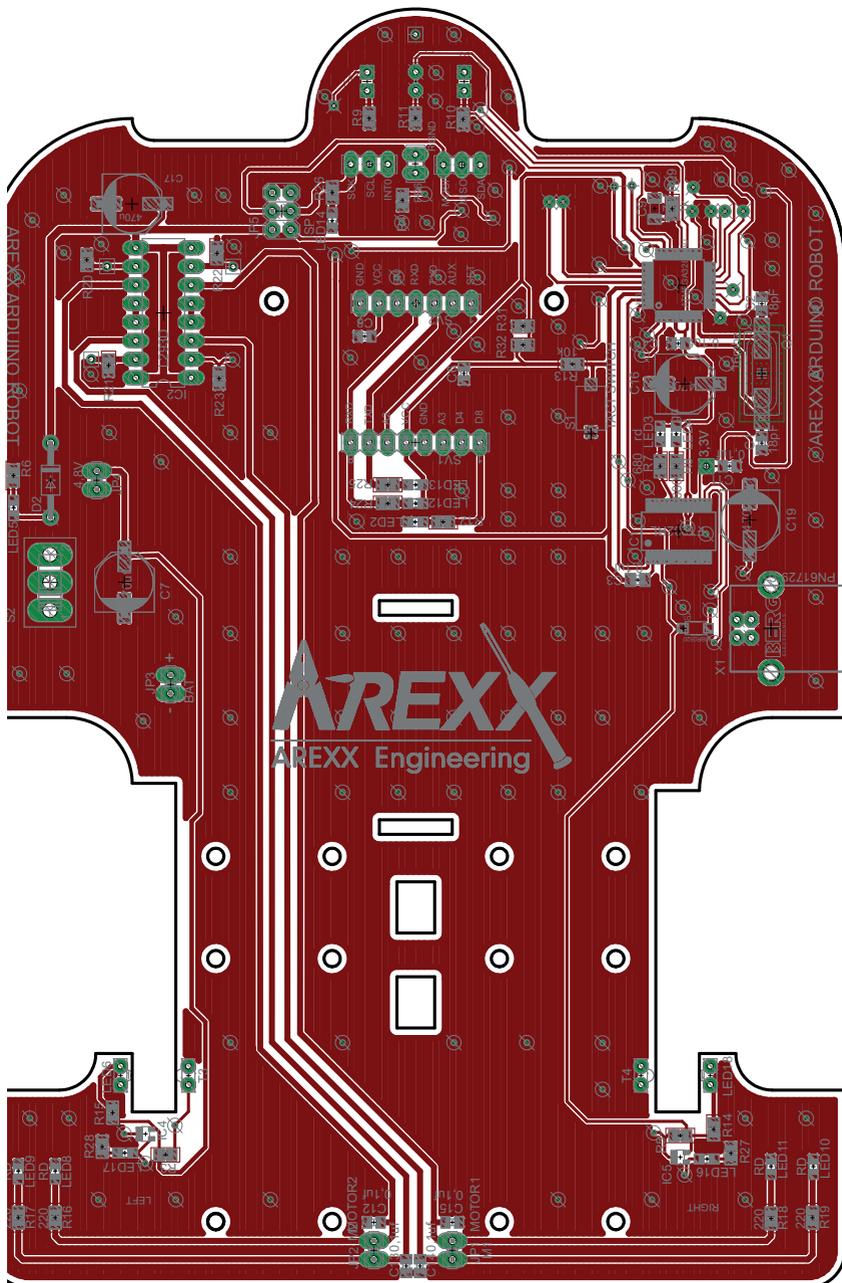
A. Elenco componenti

Componente	Valore	Contenitore
C1	18pF	0805
C2	18pF	0805
C3	0.1uF	C0805K
C4	0,1uF	0805
C6	0,1uF	0805
C7	470uF	CPOL-USF
C8	0,1uF	0805
C9	4,7uF	1206
C11	0,1uF	0805
C12	0,1uF	0805
C13	0,1uF	0805
C14	0,1uF	0805
C15	0,1uF	0805
C16	470uF	CPOL-USF
C17	470uF	CPOL-USF
C19	470uF	CPOL-USF
D1	MBR0520	SOD-123
D2	1N4001	DO41-10
IC1	FT232RL	SSOP28
IC2	L293D	DIL16
IC3	ATMEGA328AU	ATMEGA328P-AU
IC4	74AHC1G14DCK	74AHC1G14DCK
IC5	74AHC1G14DCK	74AHC1G14DCK
JP1	M1	1X02
JP2	M2	1X02
JP3	BAT	1X02
JP4	4,8V	1X02
JP5	ISP	2X03
SV2	fem header	FE07-1
T1	SFH300	LED5MM
T2	SFH300	LED5MM
T3	LPT80A	LPT80A
T4	LPT80A	LPT80A
U\$1	3,3V	PIN-T
U\$2	FE03-1	FE03-1
U\$3	FE03-1	FE03-1
U\$4	FE02-1	FE02-1
X1	PN61729-S	PN61729-S
LED1	Rd	LED5MM
LED2	BI	LEDCHIP-LED0805
LED3	Rd	LEDCHIP-LED0805
LED4	Gn	LEDCHIP-LED0805
LED5	BI	LEDCHIP-LED0805
LED6	Rd	LEDIRL80A

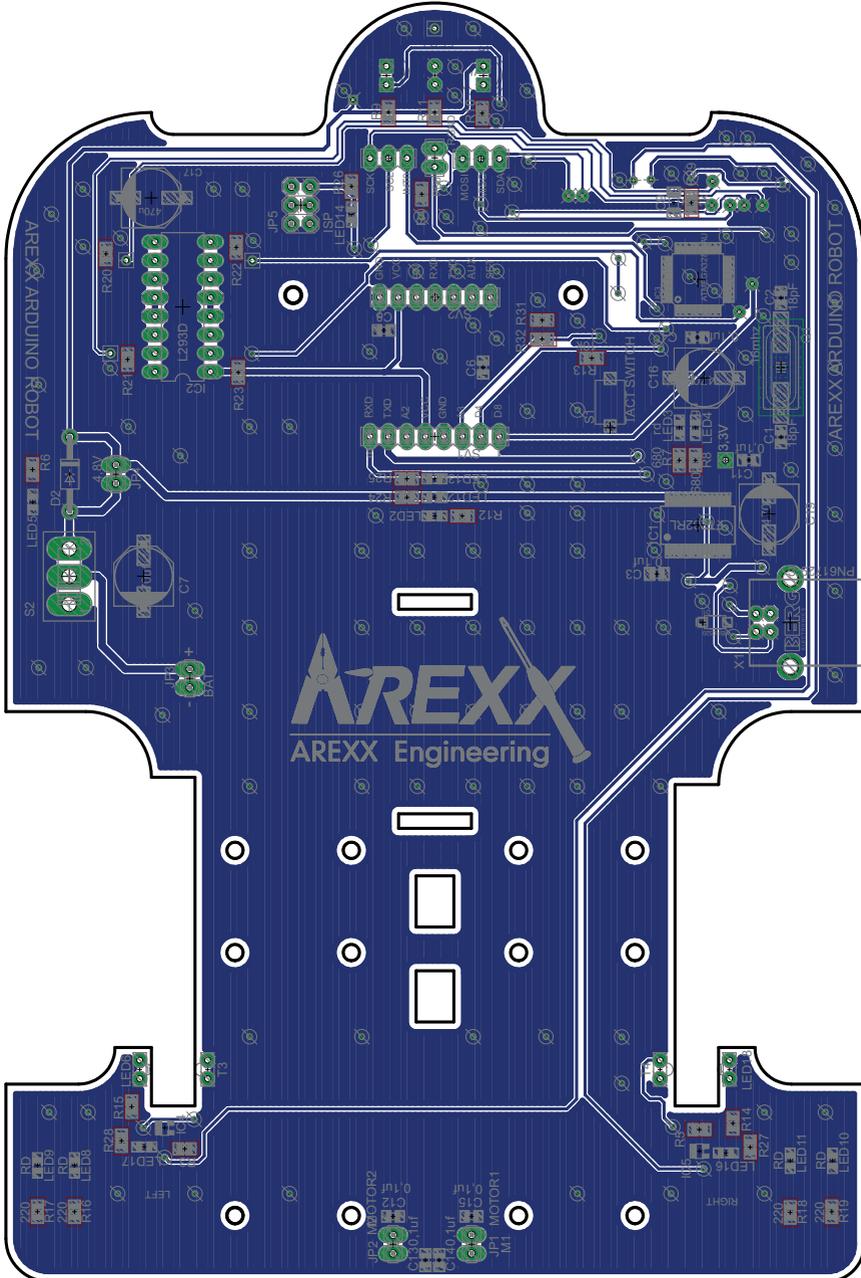
Componente Valore**Contenitore**

LED8	Rd	LEDCHIP-LED0805
LED9	Rd	LEDCHIP-LED0805
LED10	Rd	LEDCHIP-LED0805
LED11	Rd	LEDCHIP-LED0805
LED12	Gn	LEDCHIP-LED0805
LED13	Rd	LEDCHIP-LED0805
LED14	BI	LEDCHIP-LED0805
LED16	Rd	LEDCHIP-LED0805
LED17	Rd	LEDCHIP-LED0805
LED18	Rd	LEDIRL80A
Q1	16MHz	CRYSTALHC49UP
R1	20k	R-US_R0805
R2	20k	R-US_R0805
R3	1k5	R-US_R0805
R4	220	R-US_R0805
R5	1k5	R-US_R0805
R6	1k	R-US_R0805
R7	680	R-US_R0805
R8	680	R-US_R0805
R9	20k	R-US_R0805
R10	20k	R-US_R0805
R11	220	R-US_R0805
R12	220	R-US_R0805
R13	10k	R-US_R0805
R14	220	R-US_R0805
R15	220	R-US_R0805
R16	220	R-US_R0805
R17	220	R-US_R0805
R18	220	R-US_R0805
R19	220	R-US_R0805
R20	10k	R-US_R0805
R21	10k	R-US_R0805
R22	10k	R-US_R0805
R23	10k	R-US_R0805
R24	220	R-US_R0805
R25	220	R-US_R0805
R26	220	R-US_R0805
R27	220	R-US_R0805
R28	220	R-US_R0805
R29/C3	0.1uF	C0805
R31	10k	R-US_R0805
R32	12k	R-US_R0805
S1	TACT SWITCH	TACT_SWITCH
S2	255SB	255SB
SV1	fem header	FE08-1

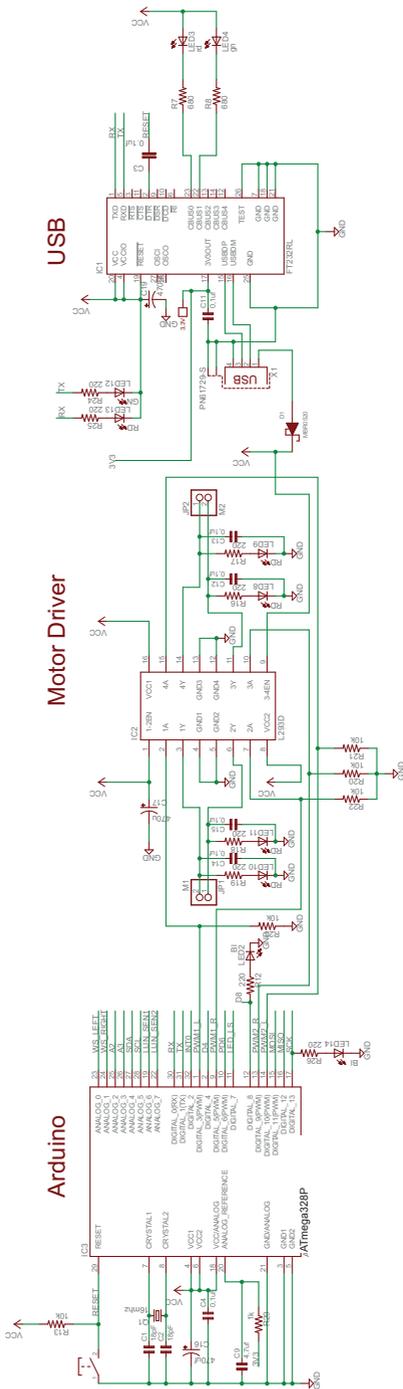
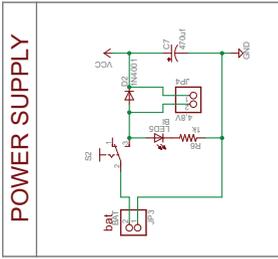
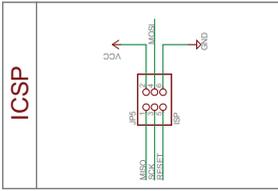
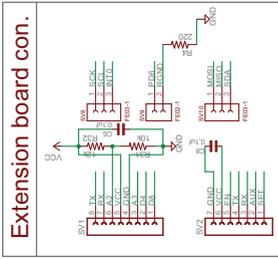
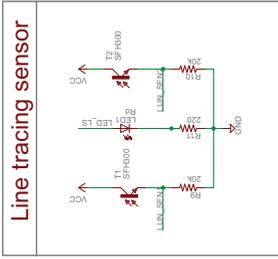
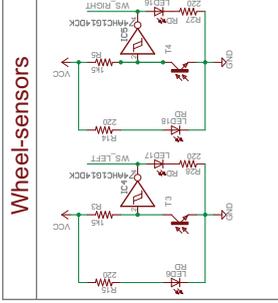
B. Lato superiore del Circuito Stampato



C. Lato inferiore del Circuito Stampato



D. Schema elettrico di AAR



USB

Motor Driver

Arduino

E. Scheda in 3D di AAR

