

Istituto Tecnico Industriale Statale B. Castelli -Brescia

Anno Scolastico 2008/2009

Classe 5^F

TOSAERBA AUTOMATICO



Alunni:
Invernici Diego
Provenzano Filippo
Sansoni Roberto

Corso di: Elettronica e Telecomunicazioni

Indice

• Introduzione.....	Pag. 3
• Abstract.....	Pag. 4
• Obiettivo del progetto.....	Pag. 5
• Descrizione generale.....	Pag. 6
○ Parte Meccanica	
○ Funzionamento	
- Automatico	
- Manuale da PC	
• Schema a blocchi.....	Pag. 11
• Schema elettrico.....	Pag. 12
• Stampati del circuito.....	Pag. 13
• Elenco Componenti.....	Pag. 14
• Il microcontrollore.....	Pag. 16
○ PIC16F876	
- Piedinatura	
- Specifiche tecniche	
- Periferiche utilizzate	
• Macchina a stati e Flow Chart del programma.....	Pag. 32
• Software.....	Pag. 38
• Motori in corrente continua.....	Pag. 51
○ Funzionamento generale di un motore in CC	
○ Circuito di comando per movimento con ponte H a RELE'	
○ Circuito di comando Lama con darlington BDX53	
• Moduli ad ultrasuoni.....	Pag. 54
○ Introduzione	
○ SRF05 (ULTRA-SONIC RANGER)	
○ Dati tecnici del SFR05	
• Sensore di pioggia.....	Pag. 57
• Sensore di temperatura.....	Pag. 59
• Rilevatore stato batteria.....	Pag. 60
• Varie ed Eventuali.....	Pag. 60
• Buzzer e LED segnalazione.....	Pag. 61
• Ricezione e Trasmissione dei dati: Interfaccia MAX232.....	Pag. 62
• Alimentazione e Consumi.....	Pag. 63
• Carica batteria.....	Pag. 64
• CONCLUSIONE.....	Pag. 67
• Costi di Realizzazione.....	Pag. 68
• Ringraziamenti.....	Pag. 69
- Docenti	
- Assistenti di Laboratorio	
- Studenti	
• Fonti.....	Pag. 70
- Internet	
- Riviste di elettronica	
• Allegati, Filmato e File.....	CD

Introduzione

Un bel giardino rappresenta sicuramente una parte decorativa di una bella casa o un condominio. Tuttavia, in buona parte dell'anno e specialmente con l'arrivo della stagione estiva, per i possessori di residenze di questo tipo c'è l'impegno della cura delle piante e del taglio del manto erboso. Ogni settimana o due, il sabato o la domenica, proprio quando ci si vorrebbe rilassare dopo una settimana di lavoro, ecco la necessità di dover prendere in mano il tosaerba (elettrico o a motore) per dedicare una buona quantità di tempo al taglio dell'erba perché, come ci ricorda il vecchio proverbio, il prato del vicino è sempre più verde, ma anche il nostro non deve sfigurare.

Ovviamente esiste l'ipotesi di assumere un giardiniere per gestire questo compito, ma negli ultimi anni progettisti ed appassionati di elettronica hanno messo a punto dei tosaerba automatici e dotati di "intelligenza", in grado di tagliare l'erba nella più completa autonomia.

Inizialmente essi erano presentati più come curiosità e attrazioni che come sistemi di vera e propria utilità, ma con lo sviluppo dell'elettronica e l'utilizzo dei più svariati sensori, se si esegue una ricerca in internet o si sfoglia una rivista specializzata se ne trovano di infinite fattezze e a prezzi variabili, anche se non sempre accessibili a tutti.

Sulla stessa linea si sono sviluppati anche sistemi per la pulizia in casa, più precisamente si tratta di aspirapolvere automatici, che possono muoversi autonomamente nella stanza o nel piano prescelto per eseguire il loro lavoro, anche quando gli abitanti della casa sono altrove.

Naturalmente sistemi di questo genere presentano dei limiti nel loro funzionamento, legati alle loro dimensioni: nel caso dei tosaerba, l'erba tagliata non viene raccolta e smaltita come farebbe un normale giardiniere, ma rimane sul manto erboso, agendo da concime per il terreno stesso. Inoltre per riuscire a mantenere una buona autonomia di funzionamento le potenze in gioco su questi tosaerba sono ridotte, consentendo un taglio ottimale solo nel caso di erba non troppo alta.

Con le ridotte dimensioni del robot e la lentezza negli spostamenti che, qualche volta, percorrono tratti già precedentemente falciati, il taglio di un intero prato richiede molto tempo, addirittura una giornata.

In considerazione di questi limiti, il sistema funziona se il robot lavora continuamente e quindi si ritrova a tagliare pochi millimetri di erba appena cresciuta e quindi più facile da tagliare.

L'erba tagliata in questo modo, come anticipato in precedenza, è di piccolissime dimensioni e non necessita di essere raccolta in quanto con la sua decomposizione provvede alla concimazione del giardino stesso. Altri limiti sono dovuti alla conformazione del giardino che deve essere ben livellato, privo di buche o pozzetti: piante, recinzioni, gradini, vasi e aiuole normalmente non comportano nessun problema in quanto i sensori a bordo del robot possono rilevarli e anzi funzionano da limite per il movimento del dispositivo, evitando che esso si fermi in una via senza uscita o vada inavvertitamente su una superficie diversa.

I costi di un'apparecchiatura commerciale non sono proprio alla portata di tutti e non avrebbe neppure senso confrontarlo con uno tradizionale.

Il costo finale sembra quindi giustificato dalla componente tecnologica e dalle varie parti utilizzate, come i motoriduttori, la parte logica di controllo, le batterie ricaricabili e molti altri.

Alla fine, pur eseguendo una funzione relativamente semplice, quale il taglio dell'erba, la tecnologia presente è notevole: rilevamento ostacoli, determinazione del percorso ideale, rilevamento delle condizioni ambientali e visualizzazione di dati su LCD o display simili.

Abstract

A beautiful garden is certainly a decorative part of a beautiful house or a condominium. However, during most of the year and especially during the summer for owners of residences of this type there is commitment for the care of plants and cutting of the turf. Every one or two weeks, on Saturdays or Sundays, just when you want to relax after a hard week of work, here is the need to use a lawnmower (petrol engine or electric motor) and spend a good amount of time cutting the grass because, as the old proverb says "the grass is always greener on the other side of the fence", but we must not be less. Of course there is the possibility to hire a gardener to manage this task, but in recent years, designers and fans of electronics have developed automatic lawnmowers with intelligence, capable of cutting the grass in complete autonomy. Initially they were presented more as curiosities and attractions than systems of real utility, but with the development of electronics and the use of more advanced sensors, if you search the internet or browse a magazine you can find infinite features and varying prices. In the same direction systems for cleaning the house have also developed and more specifically the automatic vacuum cleaner, which can move around the room or on the chosen floor to do its job, even when the inhabitants of the house are elsewhere. Of course, such systems have limitations in their operation, linked to their size: in the case of the lawnmower, the cut grass is not collected and disposed of as an ordinary gardener would do, but it remains on the turf acting as fertilizer for the land itself. In addition to this, in order to be able to maintain a good operational autonomy, powers at play on these mowers are reduced, allowing an optimal cut only if the grass is not too high. Because of the small size of the robot and its slow movements, sometimes crossing sections previously cut, cutting an entire lawn takes a while, even a day. According to these limits, the system works if the robot is working continuously and then found a few millimeters of grass just grown and therefore easier to cut. In this way the cut grass is very small and doesn't need to be collected since its decomposition provides fertilizer to the garden. Other limitations are due to the conformation of the garden which must be well trimmed, with no holes or manholes: plants, fences, steps, flower beds and pots usually do not involve any problem because the sensors on board the robot can identify them and even act as the limit for movement of the device, so that it is still in a dead-end or, inadvertently, goes to a different area. The cost of commercial equipment are high and would not even make sense to compare it with a traditional lawnmower. The final cost seems justified by the technology used by the various parts, such as motors, the control logic, rechargeable batteries, and many others. In conclusion, while performing a relatively simple function, such as the cutting of grass, this technology is substantial: obstacle detection, determination of the ideal path, detection of environmental conditions and show data on LCD displays or the like.

Obiettivo del progetto

L'obiettivo del nostro progetto è riuscire a creare un dispositivo che riesca ad orientarsi autonomamente e possa funzionare da prototipo per lo sviluppo di funzioni per rilevamento ostacoli, lettura di parametri ambientali (pioggia, temperatura e livello di carica della batteria) e implementi al suo interno la possibilità di svolgere un lavoro, aspirapolvere, lucida pavimenti o come nel nostro caso, TOSAERBA.

Il progetto da noi ideato non ha la pretesa di raggiungere l'efficienza degli apparati commerciali in vendita, ma vogliamo che replichi con le proprie funzionalità un sistema semplificato che possa muoversi autonomamente, sempre casualmente, su un terreno di natura variabile, esso sia il pavimento di casa nostra come il manto erboso del giardino della nostra casa, permettendo di svolgere la funzione di tagliare l'erba tramite una piccola "lama" posta sul ventre del tosaerba.

Objective of the project

The proposal of our project is to create a device that can orientate itself automatically and can act as a prototype for the development of functions to detect obstacles, read environmental parameters and has within with the ability to work as vacuum cleaners or, as in our case, as LAWNMOWERS.

The project as we have conceived by us doesn't want to reach the efficiency of commercial devices for sale, but we want to replicate a simplified system that can move independently and randomly, on different grounds, from floors of our house to the grassy garden, performing the function of cutting the grass with a small "blade" placed on the belly of the lawnmower.



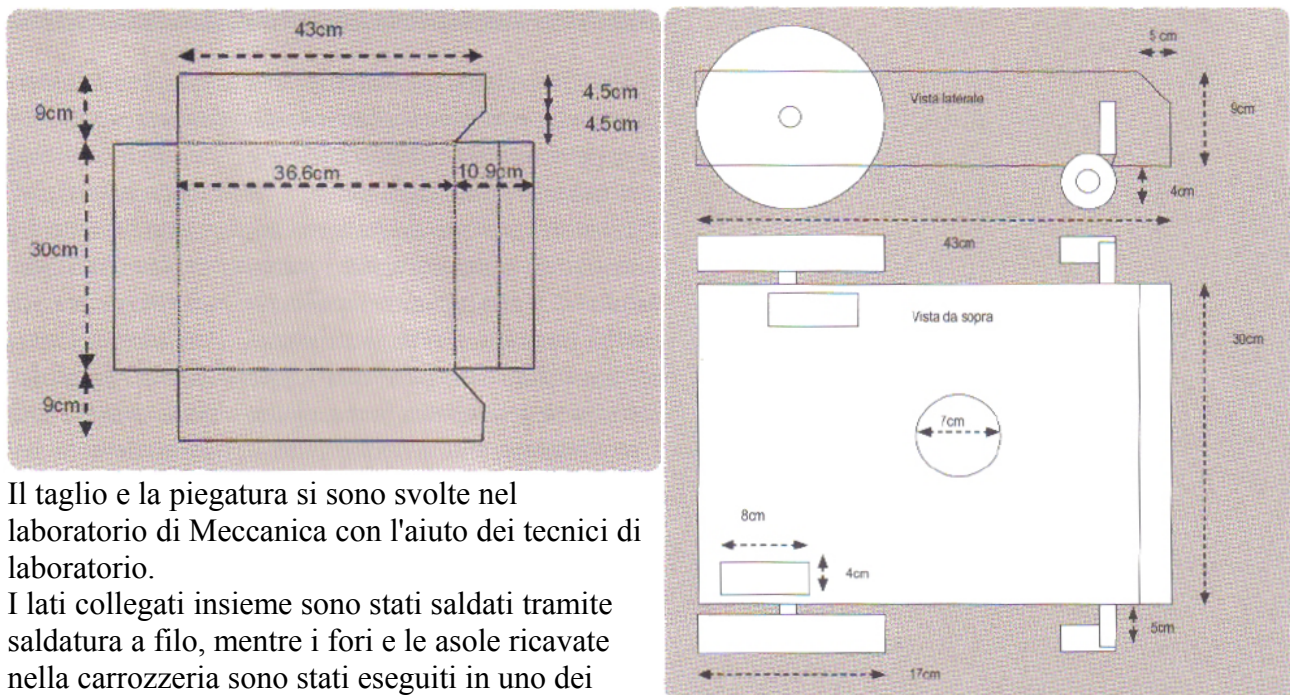
Foto: Alcuni esempi di dispositivi in commercio.

Descrizione generale

PARTE MECCANICA

Il nostro robot è composto da una struttura metallica sagomata per formare un supporto adatto a fissare sia gli apparati che consentono il movimento sia i dispositivi elettronici che formano la logica di controllo del sistema e i sensori.

La carrozzeria è stata realizzata partendo da una lamina di acciaio dello spessore di circa un millimetro (recuperata da un vecchio armadio da ufficio), successivamente dimensionata e tagliata seguendo queste indicazioni:



Il taglio e la piegatura si sono svolte nel laboratorio di Meccanica con l'aiuto dei tecnici di laboratorio.

I lati collegati insieme sono stati saldati tramite saldatura a filo, mentre i fori e le asole ricavate nella carrozzeria sono stati eseguiti in uno dei laboratori del nostro dipartimento di elettronica che dispone di vari trapani a colonna.

Nella parte posteriore sono stati ricavati gli alloggiamenti per i motori in corrente continua, recuperati da due gruppi di alzacvetri elettrici di automobili.

Il requisito fondamentale ricercato nella scelta di questi motori non è una elevata velocità di rotazione, bensì una notevole coppia che permetta al robot di muoversi su terreni di diversa conformazione e inoltre riescano a trasportare senza problemi il carico della batteria da 12V, della carcassa metallica e di ogni altro componente presente, visto che il peso del robot a montaggio ultimato è circa 10 - 12 Kg.

Per questi motivi la scelta è ricaduta su questo tipo di componente perché presenta un organo di motorizzazione solidale all'albero del motore stesso che trasforma l'elevata velocità di rotazione in un movimento a velocità minore e costante con una coppia decisamente maggiore.

Abbiamo progettato l'accoppiamento delle ruote ai motori in modo tale da non far gravare il peso del robot stesso sugli alberi di trasmissione; questo è stato possibile fissando la ruota su un supporto solidale alla struttura con un perno centrale sul quale è infilata la ruota mentre la trasmissione del moto è realizzata tramite tre viti che si vanno ad innestare nella ruota e girano attorno al perno.

Vista la forma dei motori e relativa riduzione, non ci è stato possibile occultarli interamente al di sotto della carrozzeria ma abbiamo dovuto praticare due fori sulla parte superiore per poi fissarli sia

con colla siliconica per realizzare un giunto elastico sia con dei sostegni avvolgenti in metallo che contribuiscono a mantenere in sede e in asse il motore.

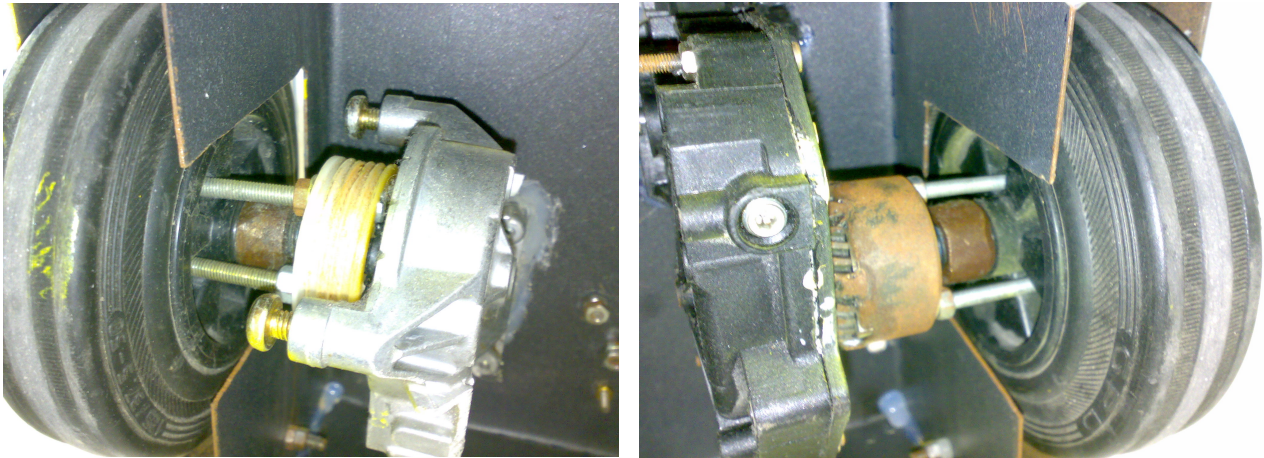


Foto: Motore Destro e Sinistro con trasmissione del movimento assiale

Per sorreggere la struttura sono necessarie altre due ruote: per fornire la libertà di movimento sono state scelte quelle pivotanti, generalmente utilizzate nelle poltroncine mobili da ufficio, fissate alla struttura con delle staffe a L: queste permettono libertà di movimento e il robot risulta guidato dalle ruote posteriori che creano la direzione in base al loro senso di rotazione.

Abbiamo poi verniciato la struttura così ottenuta (che precedentemente era opaca e color antracite) con una bomboletta spray color giallo brillante.

Per dare l'idea delle misure e della disposizione la seguente foto presenta il robot in fase di allestimento con motori, batteria e scheda elettronica:

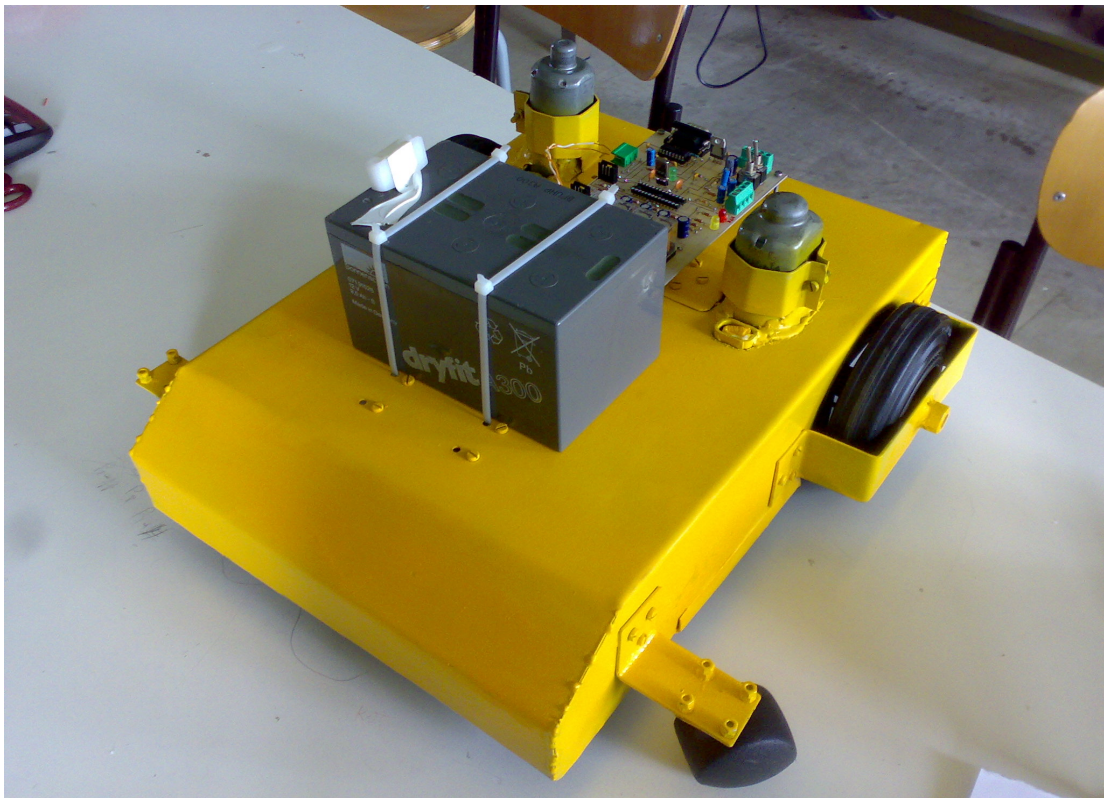


Foto: robot tosaerba durante le fasi di lavorazione.

Successivamente abbiamo voluto creare una copertura per le parti elettriche e la batteria poste sulla parte superiore del robot.

Si era pensato, inizialmente, ad un tettuccio in legno compensato, visto che a scuola non disponevamo di altri materiali e oltretutto non volevamo appesantire il sistema, che già presenta un notevole peso a causa della sua robusta struttura in acciaio e della batteria da 12V a secco (l'elemento più pesante del sistema) che ci fornisce l'autonomia e la potenza necessaria.

Fatte le dovute constatazioni però sullo scadente apporto estetico di una tale copertura e, non da meno, il fatto che esso non avrebbe permesso di osservare gli elementi interni, ci siamo orientati in un'altra direzione, ovvero utilizzare un tettuccio in Plexiglas, che oltre ad essere ancora più leggero del legno, avrebbe permesso l'osservazione della circuiteria e reso più "accattivante" l'estetica finale del robot.

Questo elemento è stato quindi recuperato da una lamina di tale materiale in disuso che abbiamo trovato a casa.

Non abbiamo seguito uno schema ben preciso per creare la copertura, ma per il perimetro esterno abbiamo seguito le dimensioni della scocca superiore, **36cm di lunghezza per 30cm di larghezza**, mentre per l'altezza ci siamo basati sul corpo della batteria e più nello specifico sulle connessioni presenti sulla parte superiore di essa. Abbiamo quindi tenuto **un'altezza di 13cm** come riferimento. Per creare questa specie di "scatola rovesciata" da applicare sul corpo del robot abbiamo anche tenuto conto del fatto che sulla parte anteriore i sensori ad ultrasuoni non dovevano essere ostacolati nell'invio degli impulsi e soprattutto le onde acustiche di ritorno dovevano arrivare perfettamente alle capsule senza problemi. Per questo **il lato anteriore è più corto verso il basso di 5cm**, in modo da creare l'apertura appena citata.

La parte più interessante nella realizzazione di questa copertura è stata quella di scegliere il metodo migliore per la costruzione della stessa: inizialmente avevamo pensato di tagliare ogni singolo lato e poi unirli con dei supporti ad L, ma in seguito abbiamo ideato di ritagliare la sagoma intera di lati e parte centrale in blocco per poi piegarla con il calore fornito da una pistola ad aria calda, per creare una struttura simile ad una normale scatola di cartone per imballaggi.

Abbiamo ritagliato il corpo centrale assieme al lato anteriore e ai due laterali lasciando da parte momentaneamente quello posteriore: volevamo infatti realizzare anche un portello che ci permettesse di accedere facilmente alla scheda senza dover togliere tutta la struttura.

Detto questo abbiamo scaldato con la pistola ad aria calda i lati da piegare e li abbiamo posti su una superficie che presentava un angolo retto (tutte queste operazioni sono state svolte per ogni singola piega).

Reso stabile il tutto abbiamo incernierato anche il portello posteriore a formare una specie di bauletto apribile verso l'alto.

Abbiamo poi creato quattro fori nella scocca del tosaerba e fissato altrettanti cilindretti metallici al Plexiglas in modo che la copertura si fissasse in un modo semplice e veloce.

Sulla parte superiore è stato poi praticato un foro per permettere il passaggio dei cavetti elettrici che connettono la piastra rilevatrice di pioggia alla scheda e il risultato finale è apprezzabile anche nella foto di copertina.

FUNZIONAMENTO

Abbiamo previsto due modalità di funzionamento del sistema, selezionabili tramite deviatore apposito: nella prima il robot si muove in completa autonomia evitando gli ostacoli grazie alla logica di controllo che interpreta i segnali provenienti dai sensori ad ultrasuoni mentre nella seconda sfrutta la comunicazione seriale per ricevere i comandi di movimento da eseguire dal PC.

Questa scelta va effettuata PRIMA di accendere il dispositivo, accorgimento adottato per aumentare la sicurezza di utilizzo e per il sistema stesso (es.: partenza con cavi di connessione al PC ecc).

MODALITA' AUTOMATICA

In questa modalità il sistema deve essere in grado di orientarsi autonomamente evitando gli ostacoli: dalla scheda della logica il microcontrollore abilita alternativamente i due moduli ad ultrasuoni: questi rispondono con un impulso di periodo variabile a seconda della distanza dell'oggetto (per una spiegazione più approfondita vedere il paragrafo “*Moduli ad ultrasuoni*”). Tramite un controllo interno vengono settati due bit, DX e SX, che forniscono informazioni su dove si trova l'ostacolo:

DX	SX	POSIZIONE OSTACOLO	NUOVA DIREZIONE
0	0	Nessun ostacolo	Avanti
0	1	Sinistra	Destra
1	0	Destra	Sinistra
1	1	Frontale	Retromarcia + Destra

I controlli vengono effettuati nella situazione in cui il robot si sposta in avanti, in modo che quando si presenta un ostacolo possa decidere lo stato successivo, che corrisponde a una direzione predefinita, opposta alla posizione dell'oggetto.

Ogni cambio di direzione è preceduto da uno stato intermedio di pausa nel quale i motori si fermano al fine di evitare un'inversione di marcia troppo rapida che comporterebbe una corrente inversa di picco superiore che a lungo andare rovinerebbe sia i motori che i loro giunti, sia i contatti dei relè. Gli stati in cui il robot gira sono temporizzati, in modo da creare angoli di sterzata fissi di circa 90°, ottenuti tramite calcoli e prove pratiche. La retromarcia è anch'essa temporizzata con valore doppio rispetto alle sterzate e abbiamo predisposto come stato successivo ad essa una svolta a destra prima di ritornare nello stato AVANTI.

Per effettuare la marcia e le sterzate abbiamo deciso di seguire la seguente logica:

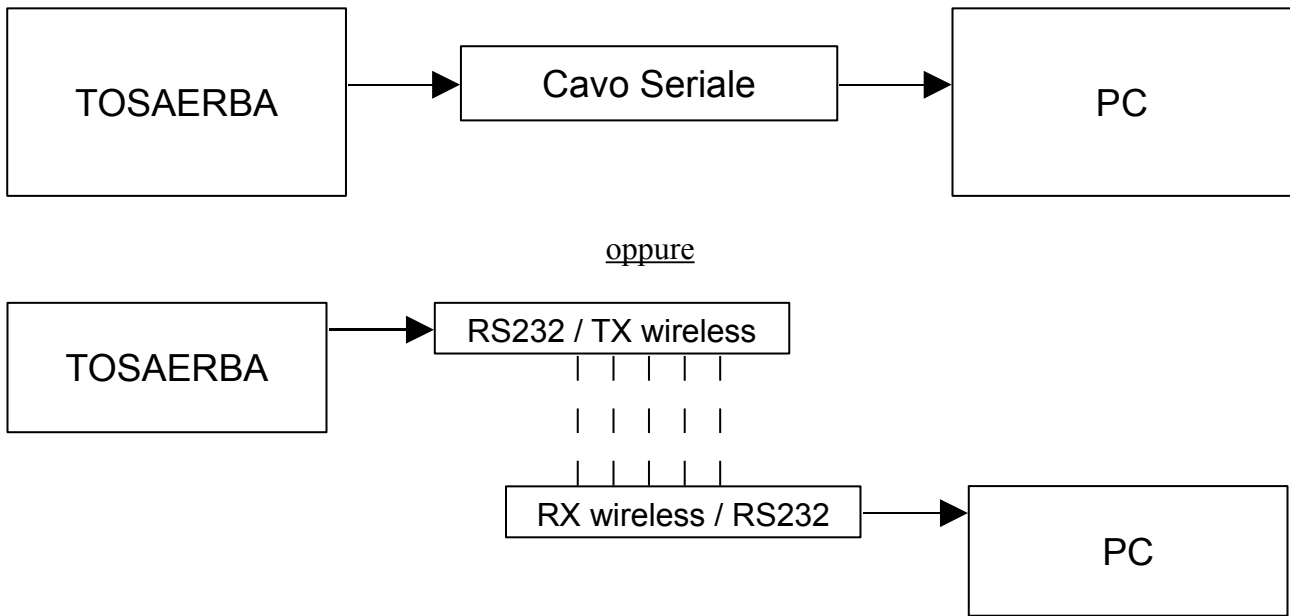
DIREZIONE	MOTORE DX	MOTORE SX
AVANTI	Avanti	Avanti
DESTRA	Indietro	Avanti
SINISTRA	Avanti	Indietro
RETROMARCIA	Indietro	Indietro
STOP	Fermo	Fermo

Il metodo migliore per pilotare i motori è stato quello di collegarli con due Ponti ad H composti da quattro relè, in modo da creare tre posizioni di funzionamento: Avanti, Indietro e Fermo.

In questo modo si può fornire l'alimentazione in un senso oppure nel senso inverso semplicemente cambiando la combinazione di eccitazione dei RELE'. (Il ponte ad H è spiegato successivamente nel capitolo “*Motori in corrente Continua*” alla voce “*Circuito di comando*”).

MODALITA' MANUALE

In questa modalità il sistema è gestito in remoto dal personal computer tramite interfaccia seriale RS232 provvisoriamente via cavo, in seguito migliorabile con l'introduzione di trasmettitore e ricevitore wireless in grado di trasmettere direttamente il segnale seriale senza bisogno di ulteriori conversioni.



E' necessario aprire e configurare il programma per la comunicazione seriale presente su Windows chiamato "Hyperterminal", situato in:

"START/Programmi/Accessori/Comunicazione/Hyperterminal"(dettagliatamente spiegato nel capitolo "Ricezione e Trasmissione dei dati").

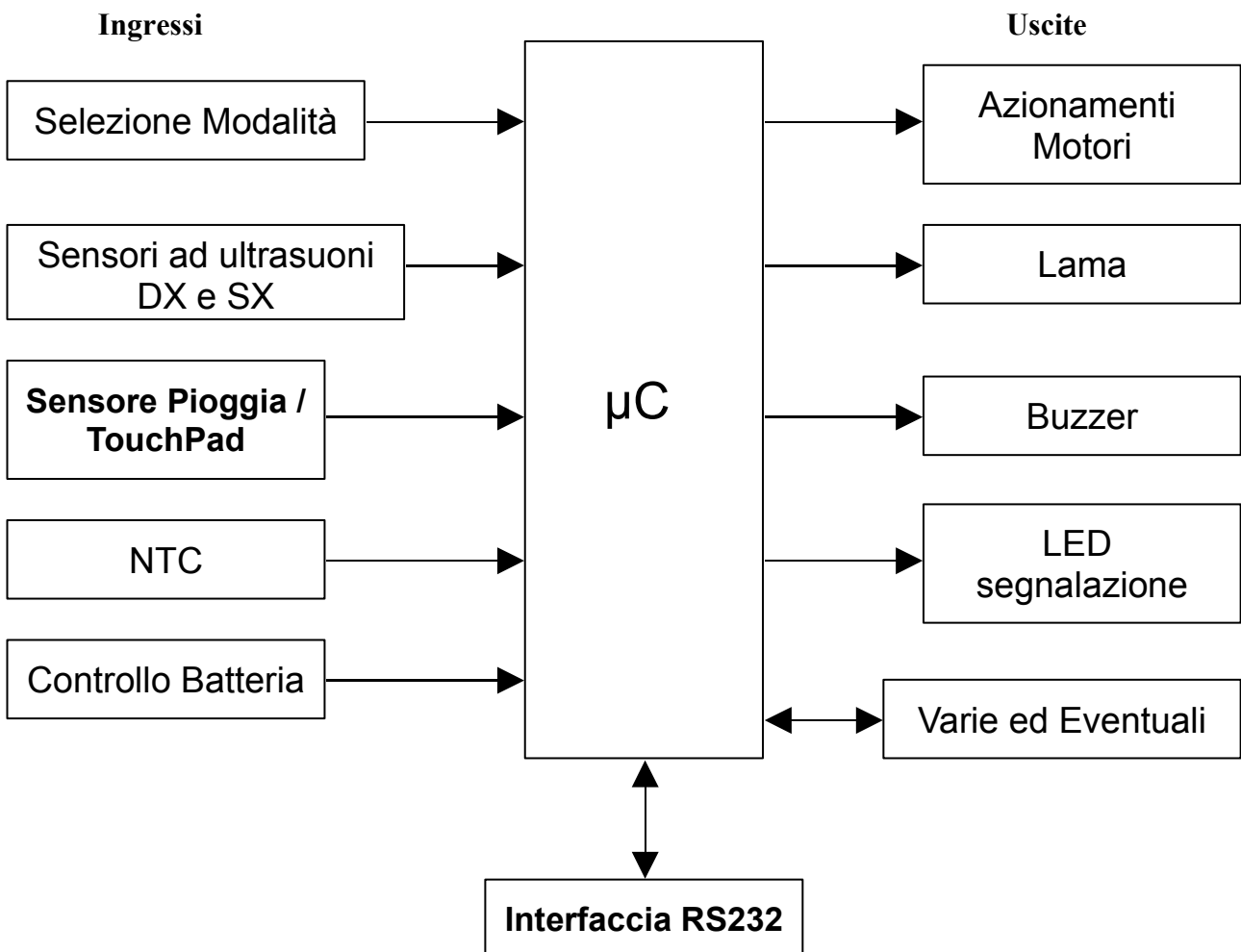
Inizialmente viene stabilita una comunicazione tra il microcontrollore e il pc dove il PIC16F876 invia una stringa contenente tutti i caratteri ASCII come verifica visiva per l'operatore che sta usando il sistema per sapere se la connessione funziona correttamente.

Successivamente è possibile, tramite la pressione di tasti predefiniti del PC, decidere il movimento da far eseguire al robot.

AZIONE	TASTO
AVANTI	W
SINISTRA	A
INDIETRO	S
DESTRA	D
STOP	X (o qualsiasi altro tasto per emergenze)

In questa modalità i sensori ad ultrasuoni sono esclusi appunto per avere il più completo controllo manuale sul sistema. In entrambe le modalità però sono presenti la lettura dei parametri ambientali e il controllo della batteria, spiegate nei capitoli relativi ai sensori dedicati ai vari scopi.

Schema a blocchi

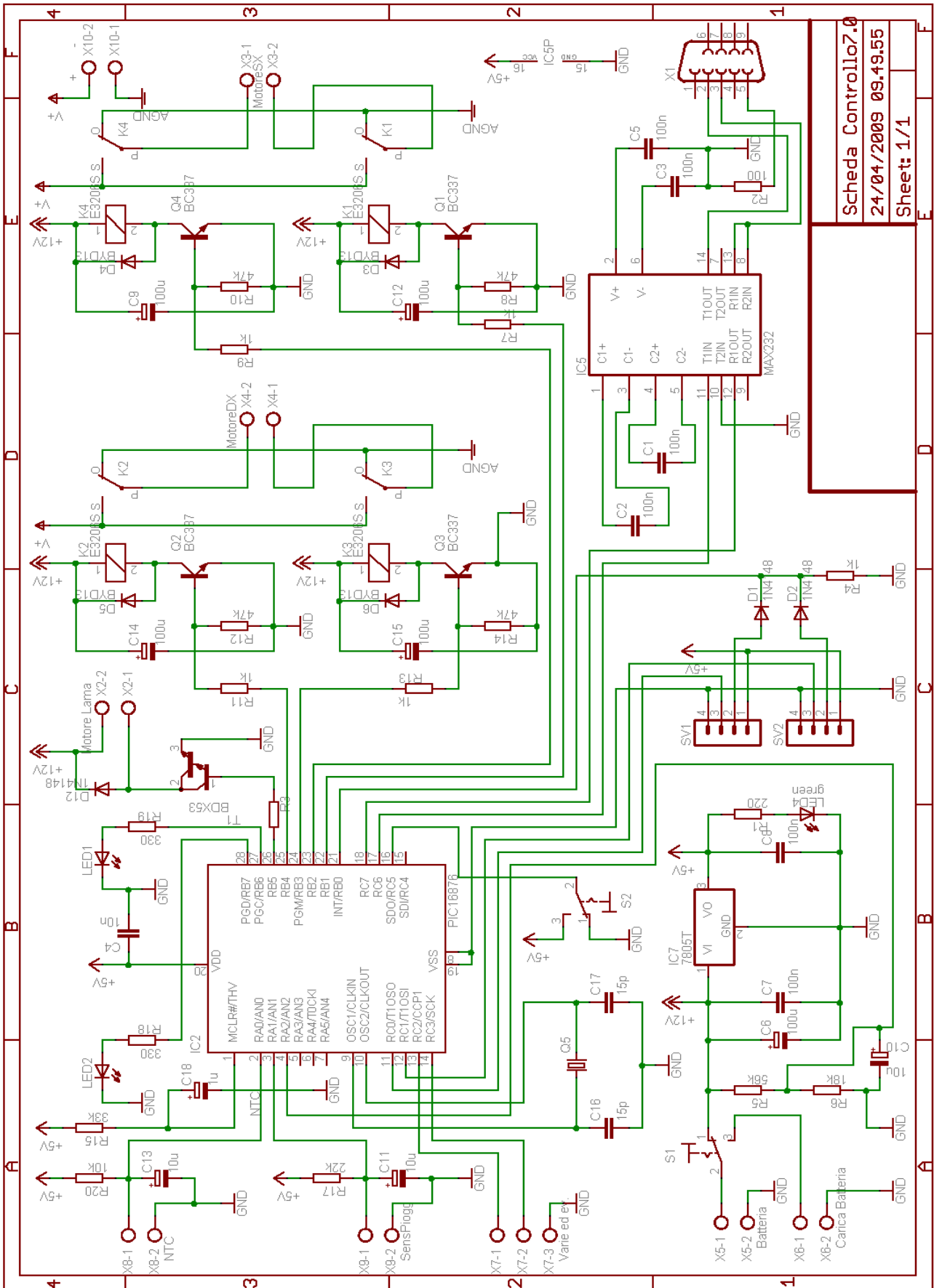


Nello schema a blocchi sono evidenziati tutti gli apparati esterni disponibili al microcontrollore presenti sulla scheda elettronica: per comodità sono stati suddivisi in ingressi (sulla sinistra) ed uscite (sulla destra).

Due periferiche dispongono di una funzionalità bidirezionale: infatti l'Interfaccia RS232 permette al PIC16F876 di Trasmettere e Ricevere dati su una linea seriale a due fili e la morsettiera "Varie ed eventuali" riserva due pin (più un punto di massa) del microcontrollore che può essere utilizzato per aggiunte future (infatti in fase di realizzazione su uno di questi I/O è stato inserito il BUZZER per le segnalazioni acustiche e che viene descritto separatamente come uscita).

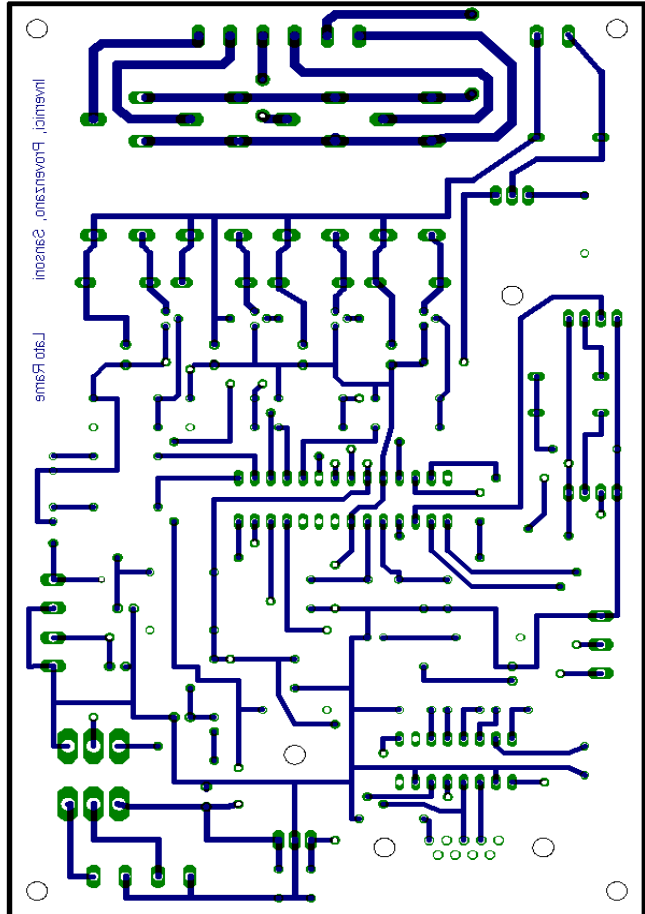
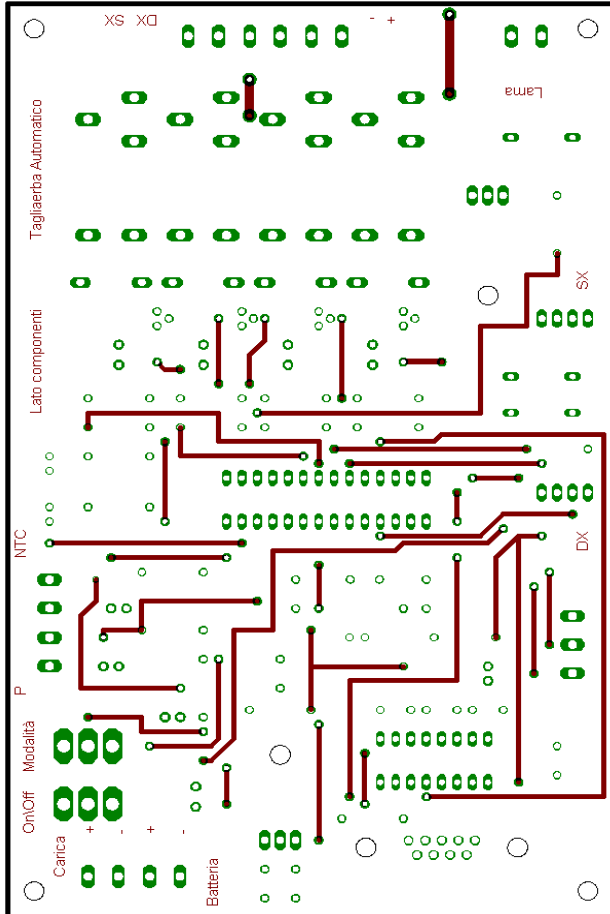
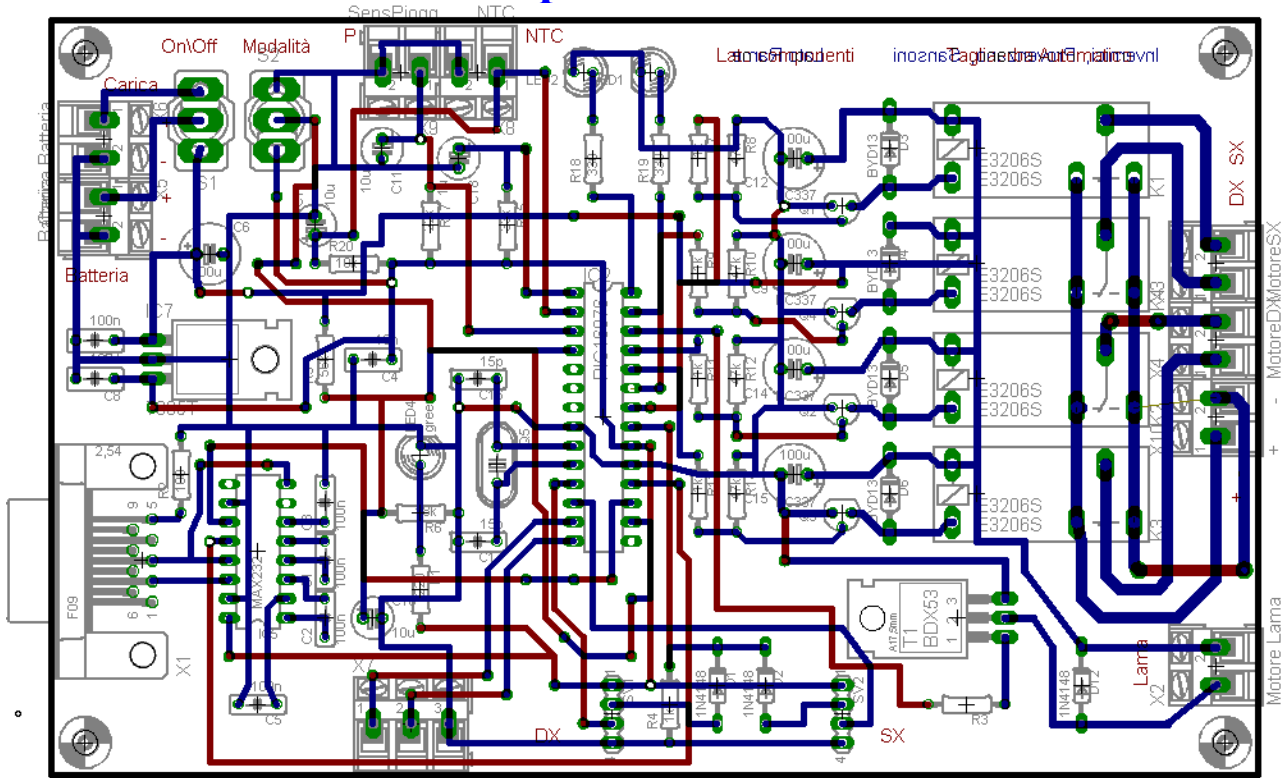
Ogni periferica viene descritta in modo completo nel relativo capitolo ad essa dedicato in seguito.

Schema elettrico



Scheda Controllo7.0
24/04/2009 09.49.55
Sheet: 1/1

Stampati del circuito



Elenco Componenti

Nome	Valore	Componente	Package	Descrizione
C1	100nF	C-EU050-025X075	C050-025X075	CAPACITOR
C2	100nF	C-EU050-025X075	C050-025X075	CAPACITOR
C3	100nF	C-EU050-025X075	C050-025X075	CAPACITOR
C4	10nF	C-EU050-030X075	C050-030X075	CAPACITOR
C5	100nF	C-EU050-025X075	C050-025X075	CAPACITOR
C6	100uF	CPOL-EUE3.5-8	E3,5-8	POLARIZED CAPACITOR
C7	100nF	C-EU050-030X075	C050-030X075	CAPACITOR
C8	100nF	C-EU050-030X075	C050-030X075	CAPACITOR
C9	100uF	CPOL-EUE3.5-8	E3,5-8	POLARIZED CAPACITOR
C10	10uF	CPOL-EUE2.5-6	E2,5-6	POLARIZED CAPACITOR
C11	10uF	CPOL-EUE2.5-6	E2,5-6	POLARIZED CAPACITOR
C12	100uF	CPOL-EUE3.5-8	E3,5-8	POLARIZED CAPACITOR
C13	10uF	CPOL-EUE2.5-6	E2,5-6	POLARIZED CAPACITOR
C14	100uF	CPOL-EUE3.5-8	E3,5-8	POLARIZED CAPACITOR
C15	100uF	CPOL-EUE3.5-8	E3,5-8	POLARIZED CAPACITOR
C16	15pF	C-EU050-025X075	C050-025X075	CAPACITOR
C17	15pF	C-EU050-025X075	C050-025X075	CAPACITOR
C18	1uF	CPOL-EUE2.5-6	E2,5-6	POLARIZED CAPACITOR
D1	1N4148	1N4148	DO35-10	DIODE
D2	1N4148	1N4148	DO35-10	DIODE
D3	BYD13	BYD13	SOD81	DIODE
D4	BYD13	BYD13	SOD81	DIODE
D5	BYD13	BYD13	SOD81	DIODE
D6	BYD13	BYD13	SOD81	DIODE
D12	1N4148	1N4148	DO35-10	DIODE
IC2	PIC16F876	PIC16F876	DIL28-3	MICROCONTROLLER
IC5	MAX232	MAX232	DIL16	RS232 TRANSEIVER
IC7	7805T	7805T	TO220H	Positive VOLTAGE REGULATOR
K1	E3206S	E3206S OMRON	E3206S	RELAY
K2	E3206S	E3206S OMRON	E3206S	RELAY
K3	E3206S	E3206S OMRON	E3206S	RELAY
K4	E3206S	E3206S OMRON	E3206S	RELAY
LED1		LED5MM	LED5MM	LED yellow
LED2		LED5MM	LED5MM	LED red
LED4		LED5MM	LED5MM	LED green
Q1	BC337	BC337	SOT54A	NPN TRANSISTOR
Q2	BC337	BC337	SOT54A	NPN TRANSISTOR
Q3	BC337	BC337	SOT54A	NPN TRANSISTOR
Q4	BC337	BC337	SOT54A	NPN TRANSISTOR
Q5	4 MHz	CRYTALHC49U70	HC49U70	CRYSTAL
R1	220 Ω	R-EU_0207/10	0207/10	RESISTOR 1/4W
R2	100 Ω	R-EU_0207/10	0207/10	RESISTOR 1/4W
R3	Ω	R-EU_0207/10	0207/10	RESISTOR 1/4W

R4	1 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R5	56 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R6	18 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R7	1 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R8	47 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R9	1 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R10	47 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R11	1 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R12	47 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R13	1 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R14	47 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R15	33 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R17	22 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
R18	330 Ω	R-EU_0207/10	0207/10	RESISTOR 1/4W
R19	330 Ω	R-EU_0207/10	0207/10	RESISTOR 1/4W
R20	10 kΩ	R-EU_0207/10	0207/10	RESISTOR 1/4W
S1		320-916	320-916	TOGGLE SWITCH
S2		320-916	320-916	TOGGLE SWITCH
SV1		MA04-1	MA04-1	PIN HEADER
SV2		MA04-1	MA04-1	PIN HEADER
T1	BDX53	BDX53	TO220	NPN DARLINGTON TRANSISTOR
X1		F09H	F09H	SUB-D
X2	Motore Lama	AK300/2	AK300/2	CONNECTOR
X3	MotoreSX	AK300/2	AK300/2	CONNECTOR
X4	MotoreDX	AK300/2	AK300/2	CONNECTOR
X5	Batteria	AK300/2	AK300/2	CONNECTOR
X6	Carica Batteria	AK300/3	AK300/3	CONNECTOR
X7	Varie ed ev.	AK300/2	AK300/2	CONNECTOR
X8	NTC	AK300/2	AK300/2	CONNECTOR
X9	SensPiogg	AK300/2	AK300/2	CONNECTOR
X10	+ -	AK300/2	AK300/2	CONNECTOR
F1	10A			FUSE
BATTERY	12V	Dryfit A300		12Vdc BATTERY lead-free

La batteria a 12V utilizzata è del tipo a secco e di 9,5Ah che garantisce una ottima autonomia (si possono usare batterie anche inferiori in termini di Ah a scapito della loro durata durante l'utilizzo)

Il Fusibile da 10A con supporto (modello da automobile) è stato inserito in serie lungo il cavo di alimentazione che porta i 12V dalla batteria alla scheda di alimentazione come protezione contro un corto circuito accidentale.

Tutti gli altri componenti sono invece montati sulla scheda elettronica.

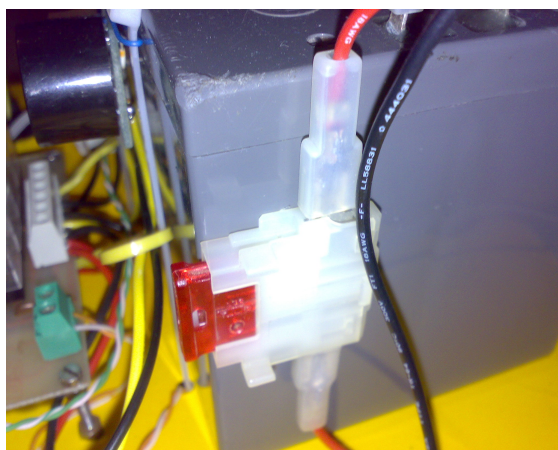


Foto: Particolare del Fusibile applicato solidalmente alla batteria.

Il Microcontrollore

Un microcontrollore o microcontroller, detto anche computer single chip è un sistema a microprocessore completo, integrato in un solo chip, progettato per ottenere la massima autosufficienza funzionale ed ottimizzare il rapporto prezzo-prestazioni per una specifica applicazione, a differenza, ad esempio, dei microprocessori impiegati nei personal computer, adatti per un uso più generale.

I microcontroller sono la forma più diffusa e più invisibile di computer.

Comprendono la CPU, un certo quantitativo di memoria RAM e memoria ROM (può essere PROM, EPROM, EEPROM o FlashROM) e una serie di interfacce di I/O (input/output) standard, fra cui molto spesso bus (I²C, SPI, CAN, LIN).

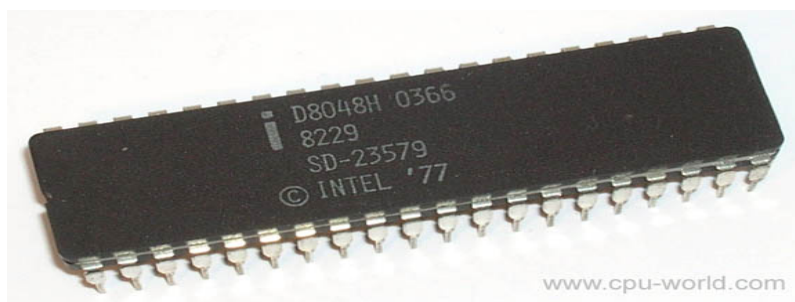
Le periferiche integrate sono la vera forza di questi dispositivi: si possono avere convertitori ADC e convertitori DAC multicanale, timer/counter, USART, numerose porte esterne bidirezionali bufferizzate, comparatori, PWM.

Sono contenuti in una quantità di apparecchi ed elettrodomestici, come ad esempio, videoregistratori e la quasi totalità dei televisori costruiti dopo il 1984, nelle macchine fotografiche e nelle videocamere, nei lettori CD e DVD, nei forni a microonde, nei controlli automatici di macchine industriali, in molte lavatrici e frigoriferi di ultima generazione, nelle centraline di controllo delle motociclette e delle automobili, (anche molte decine di microcontroller in una sola automobile) negli antifurto elettronici, nei registratori di cassa dei negozi, negli sportelli Bancomat, nei distributori automatici e nelle centraline dei semafori.

La loro capacità di calcolo è molto limitata, a dispetto della velocità ragguardevole che possono raggiungere e di solito eseguono lo stesso programma (firmware) per tutta la durata del loro funzionamento.

Negli ultimi anni il loro uso è aumentato grazie all'estrema versatilità ed al costo bassissimo, diffondendosi anche fra gli hobbisti e gli appassionati di elettronica.

Il primo microcontrollore della storia fu il piccolissimo 8048 della Intel con solo 1kB di ROM segmentata in pagine ed appena 64 Bytes di RAM, basato sull'architettura Harvard (bus dei dati e del codice separati).



Per la realizzazione del progetto, il PIC16F876 si è rivelato il più adatto, proprio per la presenza di un numero notevole di PORT di I/O (PORTA e PORTB da 8 bit e PORTC da 4 bit, tutti bidirezionali), timer interni, convertitori A/D e altre periferiche come spiegato nel paragrafo successivo.

PIC 16F876

La scelta del tipo di controllo è stata dettata sia da esigenze meccaniche sia specifiche di progetto legate a ciò che desideravamo svolgesse il robot.

L'esigenza di un sistema versatile, dal basso costo e di facile realizzazione ci ha spinti alla scelta di un sistema a microcontrollore, che sovrintendesse al movimento completo, con la possibilità di espansioni e modifiche future facilmente realizzabili tramite aggiunte e migliorie al software presente su di esso.

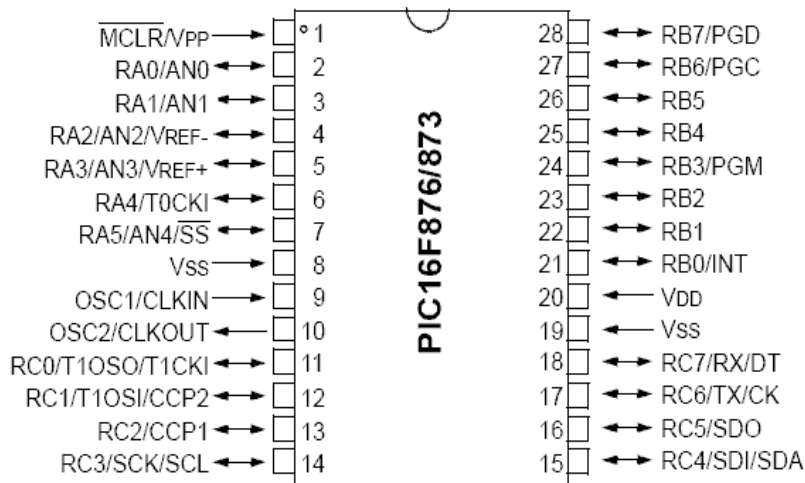
Valutando il numero di ingressi e uscite utilizzati (16 in totale tra I/O suddivisi su 3 PORT) e le periferiche di cui abbiamo bisogno, come il convertitore A/D e il modulo AUSART, la scelta è ricaduta sul dispositivo PIC16F876 della Microchip, sfruttando come software per la creazione del programma MPLAB (sempre fornito da Microchip) e il compilatore per il linguaggio C denominato CC5X (Knudsen).

Essendo molte le funzioni già implementate al suo interno, la circuiteria di supporto specifico al PIC è abbastanza semplificata, e consiste nelle alimentazioni e nell'oscillatore.

La parte più imponente sul circuito stampato quindi è costituita dal condizionamento degli ingressi (sensori vari) e dagli azionamenti per le uscite (relè o darlington), oltre al modulo RS232 (comprendente l'integrato con la pompa di carica) e i LED di segnalazione.

Questo dispositivo possiede 28 pin, ognuno dei quali ha specifiche funzioni in seguito riportate.

Piedinatura e Descrizione PINOUT:



Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS (3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/V _{PP}	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0 RA1 can also be analog input1. RA2 can also be analog input2 or negative analog reference voltage. RA3 can also be analog input3 or positive analog reference voltage. RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST (1)	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST (2)	
RB7/PGD	28	28	I/O	TTL/ST (2)	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I2C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I2C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
V _{SS}	8,19	8,19	P	—	Ground reference for logic and I/O pins.
V _{DD}	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power

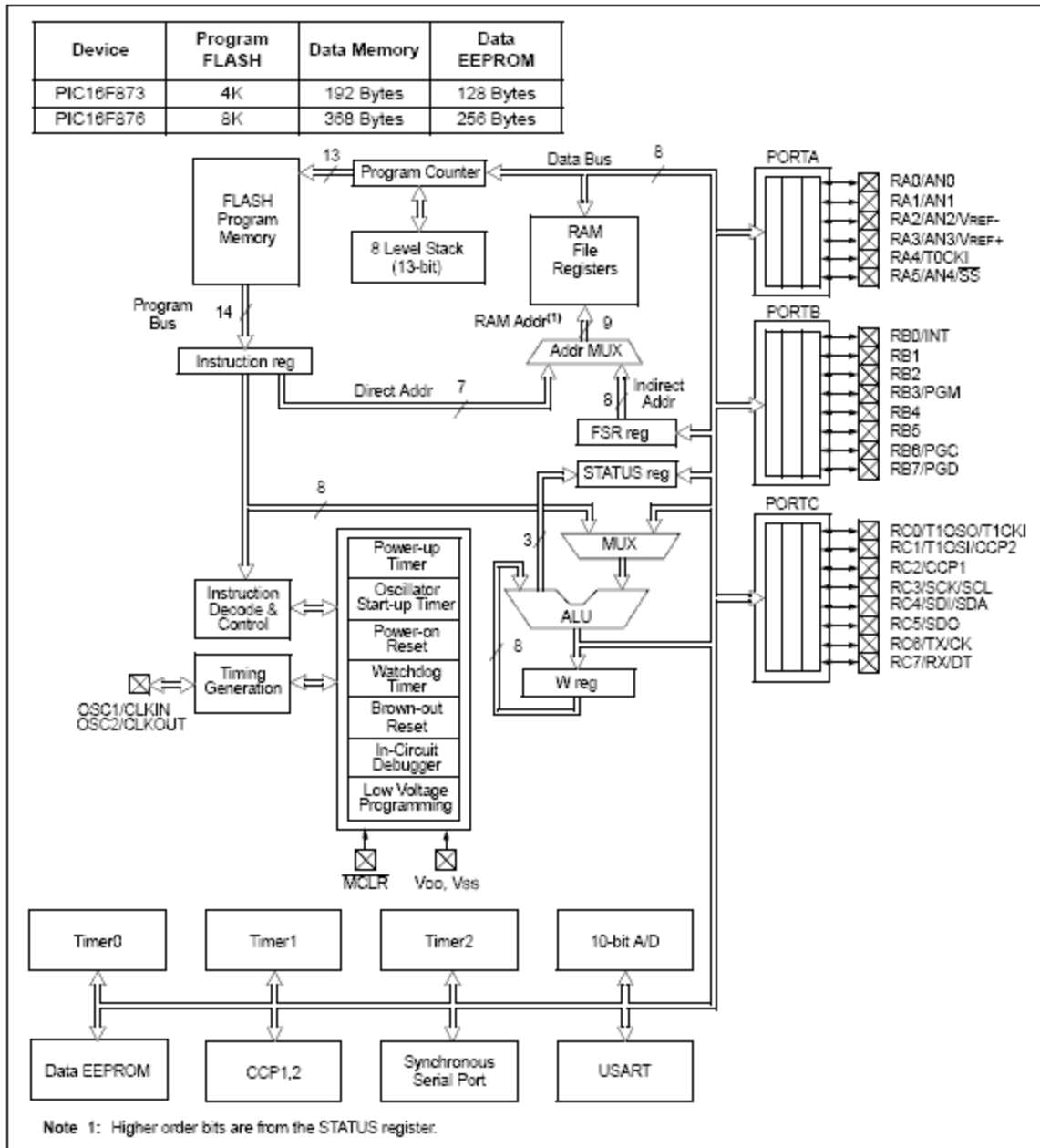
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

Schema a Blocchi interno del PIC16F876



Caratteristiche interne del microcontrollore (specifiche tecniche):

- Alte prestazioni della RISC CPU;
- Linguaggio di programmazione semplificato tramite l'uso di compilatori C;
- Velocità del sistema variabile a seconda del quarzo impiegato fino ad un massimo di 20MHz
- Fino a 8k*14 words di memoria di programma FLASH;
- Fino a 368*8 byte di memoria dati RAM;
- Fino a 256*8 byte di memoria dati EEPROM (non sfruttata nel nostro sistema);
- Possibilità di INTERRUPT (sorgenti da periferiche interne e pin esterno RB0);
- *Power-on Reset* (POR): reset all'accensione;
- Power-up Timer (PWTR) e Oscillator Start-up Timer (OST) ;
- Watchdog Timer (WDT) (non sfruttata nel nostro sistema);
- Protezione del codice programmabile;
- Opzione di selezione dell'oscillatore (interno, RC, Quarzo);
- Tecnologia *CMOS/EEPROM* ad alta velocità e bassa potenza;
- Programmazione seriale *in-circuit* attraverso due pin;
- Intervallo di tensione di lavoro: da 2,2V a 5,5V;
- Alta corrente di ingresso e di uscita: 25mA;
- Consumi e dissipazione di potenza ridotti: < 2mA @ 5V, 5Mhz;
20µA @ 3V, 32 KHz;
1µA tipico di corrente di *stand-by*

Periferiche del microcontrollore:

- Timer0: timer/counter a 8 bit con 8 bit di prescaler;
- Timer1: timer/counter a 16 bit con prescaler: può essere incrementato anche mentre è spento tramite una sorgente/clock esterno;
- Timer2: timer/counter a 8 bit con 8 bit di registro periodico, prescaler e postscaler;
- Modulo *CCP*: Capture è a 16 bit, con risoluzione massima di 12,5 ns;
Compare è a 16 bit, con risoluzione massima di 200 ns;
PWM con risoluzione massima di 10 bit;
- Convertitore A/D multi-canale a 10 bit (con possibilità di lettura giustificata a DX o a SX)
- Porta seriale Sincrona *SSP*;
- *USART*: Universal Synchronous Asynchronous Receiver Transmitter
- *PSP*: Parallel Slave Port con 8 bit di larghezza, con controlli esterni di RW, WR, CS;

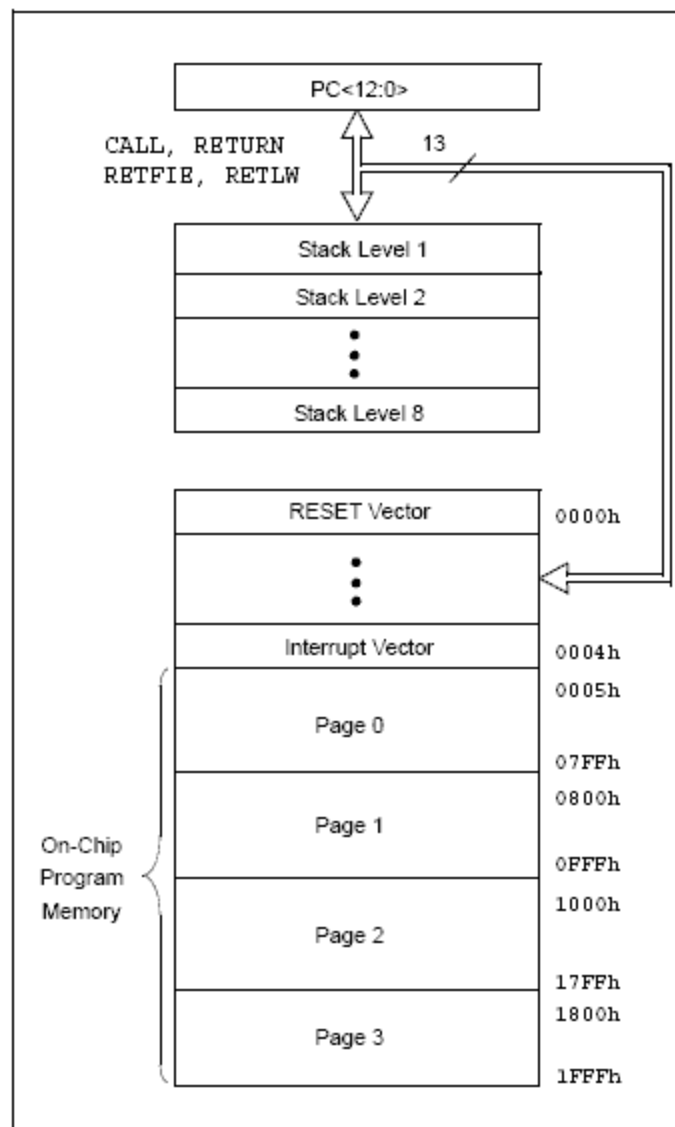
Organizzazione della Memoria di Programma (*Program Memory*):

Ci sono tre blocchi di memoria in ogni microcontrollore della serie PIC16F87X.

La memoria di programma (*Program Memory*) e la memoria dati (*Data Memory*) hanno bus separati, in modo da avere un accesso simultaneo alle due memorie.

Il microcontrollore inoltre ha un contatore di programma di 13 bit capace di indirizzare 8k*14 spazi di memoria di programma e 8k*14 words di memoria di programma FLASH.

Il RESET VECTOR è 0000h mentre l'INTERRUPT VECTOR è 0004h (nel software è indicato).



Organizzazione della Memoria Dati (DATA Memory):

La memoria dati è divisa in più banchi (4), i quali contengono i registri GENERAL PURPOSE REGISTER e SPECIAL FUNCTION REGISTER.

I bit 5 e 6 dello STATUS REGISTER sono adibiti alla selezione dei banchi di memoria.

Le basse locazioni di questi banchi sono riservate per gli Special Function Register e i General Purpose Register sono implementati come RAM statica.

Tutti i banchi implementati contengono speciali registri funzionali, mentre qualche registro in particolare può essere rispecchiato in un altro banco per la riduzione di codice e ottenere un accesso più velocizzato (ad esempio lo STATUS REGISTER oppure il registro FSR ecc..).

Si può accedere a entrambi i file di registro direttamente, o indirettamente attraverso il FILE SELECT REGISTER (FSR).

Gli Special Function Register sono registri usati dalla CPU e dai moduli esterni per controllare le operazioni desiderate del componente.

Questi registri sono implementati come RAM statica.

Gli Special Function Register possono essere classificati in due tipologie: interni e periferici.

File Address		File Address		File Address		File Address	
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ^(*)	08h	TRISD ^(*)	88h		108h		188h
PORTE ^(*)	09h	TRISE ^(*)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h	General Purpose Register 16 Bytes	111h	General Purpose Register 16 Bytes	191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h				120h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
	7Fh	accesses 70h-7Fh	EFh F0h	accesses 70h-7Fh	16Fh 170h	accesses 70h - 7Fh	1EFh 1F0h
Bank 0		Bank 1	FFh	Bank 2		Bank 3	1FFh

Unimplemented data memory locations, read as '0'.
^{*} Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
 2: These registers are reserved, maintain these registers clear.

Periferiche utilizzate

PORT per I/O:

Le porte ed i registri ad esse legati presenti nel PIC16F876 sono le seguenti:

PORT-A

Il PORTA è un port bidirezionale a 6 bit di larghezza sia digitale che con alcuni ingressi analogici e il registro TRISA è il registro di direzione dati ad esso connesso.

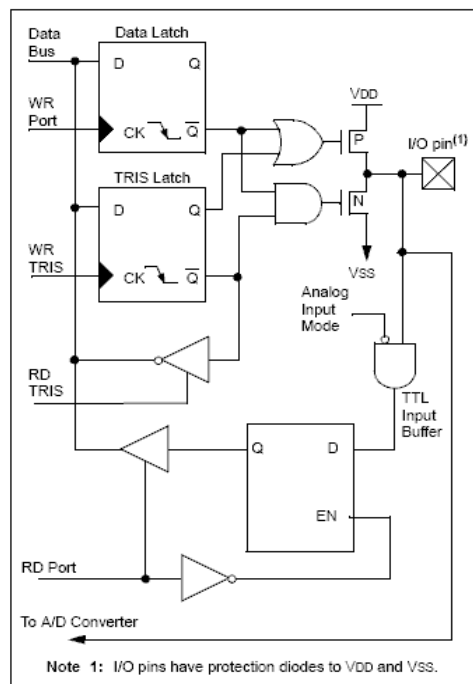
Settando o resettando i bit del TRISA in corrispondenza del bit del PORT che si vuole definire si decide se essi saranno INPUT oppure OUTPUT: ponendo a 1 il bit si imposta il corrispondente pin come ingresso ad alta impedenza, ponendo a 0 il bit invece si imposta il corrispondente pin come uscita.

Il registro TRISA controlla le direzioni dei pin RAX (RA0,RA1 ecc..), anche quando questi vengono utilizzati come ingressi analogici.

L'utilizzatore deve quindi assicurarsi in ogni caso che i bit del TRISA siano mantenuti a 1 o a 0 a seconda dell'utilizzo del rispettivo pin del microcontrollore.

Leggendo il registro di PORTA viene letto lo stato dei pin in ingresso mentre scrivendo un certo valore in uscita questi pin saranno impostati a 1 o 0 a seconda del dato scritto.

Schema a blocchi generale di input/output del PORTA



Registri associati al PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.

Shaded cells are not used by PORTA.

PORT-B

Il PORTB è un port bidirezionale a 8 bit di larghezza e anch'esso possiede il registro associato TRISB per l'assegnazione di I/O.

Tre pin del PORTB sono multiplexati e presentano la particolarità della funzione di programmazione a bassa tensione (*Low Voltage Programming Function*): RB3/PGM, RB6/PGC, RB7/PGD.

Tutti i pin del PORTB hanno un pull-up interno più debole rispetto agli altri port. Un bit singolo di controllo può accendere tutti i pull-up. Questo è ottenibile tramite la modifica del bit RBPU.

Il debole pull-up è automaticamente spento quando il pin della porta è configurato come uscita.

I pull-up sono disabilitati con il *Power-on Reset*.

Registri associati al PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.

Shaded cells are not used by PORTA.

PORT-C

Il PORTC è un port bidirezionale a 8 bit di larghezza e anch'esso possiede il registro associato TRISC per l'assegnazione di I/O.

Qualche periferica associata al PORTC non tiene conto (*override*) del fatto che i bit del TRISC possono configurare il i pin come uscite; altre periferiche, invece, non tengono conto del fatto che i bit del TRISC possono configurare i pin come ingresso. Poiché i bit “override” di TRISC sono attivi, le periferiche sono abilitate e le istruzioni di lettura-modifica-scrittura con TRISC come destinazione possono essere evitate.

L'utente può riferirsi alla sezione periferica corrispondente per il corretto settaggio del TRISC.

Registri associati al PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged

Modulo Timer0:

Il modulo contatore/timer del Timer0 ha le seguenti caratteristiche:

- 8 bit di timer/contatore;
- Leggibile e scrivibile;
- Clock interno o esterno selezionabile;
- Selezione del fronte del clock esterno;
- Prescaler programmabile via software a 8 bit;
- Interrupt alla fine del conteggio (*overflow*);

La periferica in questione può operare come timer o come contatore.

La modalità timer è selezionata mettendo a 0 il bit T0CS (*Timer0 Clock Source*) nel registro OPTION_REG.

Nel modo timer il modulo incrementerà a ogni ciclo di istruzione il proprio valore (senza prescaler).

Se il registro timer0 viene scritto, l'incremento è inibito per i due seguenti cicli di istruzione.

L'utente può quindi lavorare intorno a questo fattore scrivendo, aggiustando o resettando (come nel nostro caso) il valore contenuto nel registro del Timer0.

La modalità contatore è selezionata mettendo a 1 il bit T0CS. In modalità contatore il modulo Timer0 incrementerà su entrambi i fronti del pin RA4/T0CK1.

Il fronte di salita è determinato dal bit T0SE (*Timer0 Source Enable Select*).

Quando un clock esterno è usato per il Timer0 deve avere determinati requisiti. Questi garantiscono che il clock esterno possa essere sincronizzato con il clock interno (T_{osc}). Inoltre c'è un ritardo nell'effettivo incremento del Timer0 dopo la sincronizzazione.

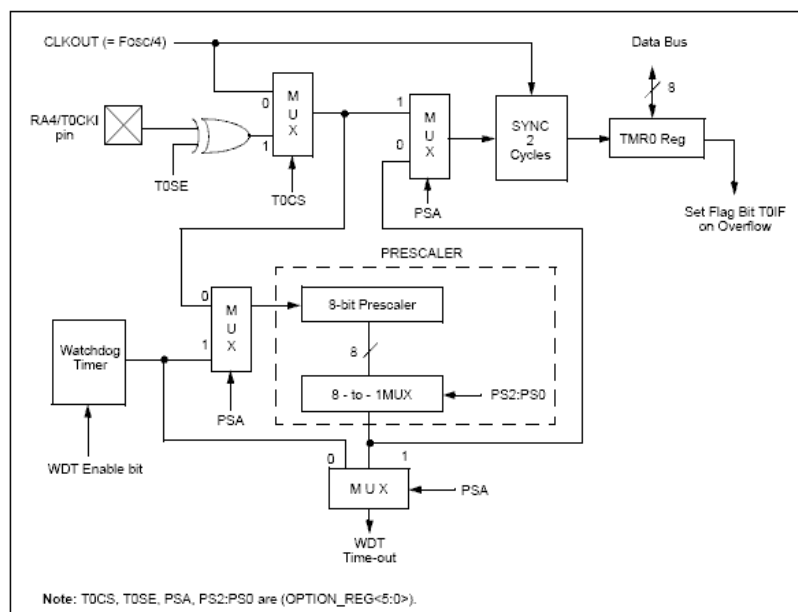
Prescaler

Un contatore ad 8 bit è utilizzabile come prescaler per il modulo Timer0, o come postscaler per il Watchdog Timer (da noi non utilizzato).

Il prescaler non è leggibile o scrivibile ma i bit PSA e PS0,PS1,PS2 del registro OPTION_REG determinano i compiti e i rapporti del prescaler.

Ponendo a 0 il PSA si assegnerà il prescaler al Timer0.

Seguendo la tabella riportata sul datasheet invece è possibile regolare il valore effettivo del prescaler tramite i bit PS0,PS1,PS2.



Schema a blocchi del Timer0

OPTION_REG e impostazioni del prescaler

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **RBP**
- bit 6 **INTEDG**
- bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

USART

Il modulo USART (*Addressable Synchronous Asynchronous Receiver Transmitter*) è uno dei due moduli seriali di I/O. L'USART può essere configurato come un sistema asincrono full duplex che può comunicare con componenti esterni come terminali CRT e il PC, oppure può essere configurato come un sistema sincrono half duplex che può comunicare con periferiche come i circuiti integrati, convertitori A/D esterni al PIC, EEPROM per il salvataggio di dati ecc..

L'USART può essere configurato nei seguenti modi:

- Asincrono (full duplex)
- Sincrono-Master (half duplex)
- Sincrono-Slave (half duplex)

Il bit SPEN e i bit TRISC <7:6> devono essere settati in modo da configurare i pin RC6/TX/DT e RC7/RX/DT come canali per la trasmissione seriale.

Il modulo USART ha anche una capacità di comunicazione in grado di gestire più processi contemporaneamente, che utilizza 9 bit per l'indirizzamento, ma questo accorgimento non è da noi utilizzato.

Generatore del BAUD-Rate dell'USART

Il BRG (*Baud Rate Generator*) supporta sia la modalità asincrona che quella sincrona dell'USART. E' un generatore dedicato a 8 bit.

Il registro SPBRG controlla il periodo di un timer free-running a 8 bit.

In modalità asincrona il bit BRGH controlla anche il Baud-rate. In modalità sincrona, invece, questo bit viene ignorato.

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

X = value in SPBRG (0 to 255)

TABLE 10-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

Il valore da inserire nel registro SPBRG è quindi legato al Baud-rate desiderato e alla frequenza dell'oscillatore scelto per il microcontrollore.

Nel nostro caso, con quarzo a 4 Mhz e Baud-rate=9600 il valore corretto da inserire è 25. (per chiarimenti consultare il datasheet del PIC16F876 nella sezione USART)

Modo ASINCRONO dell'USART

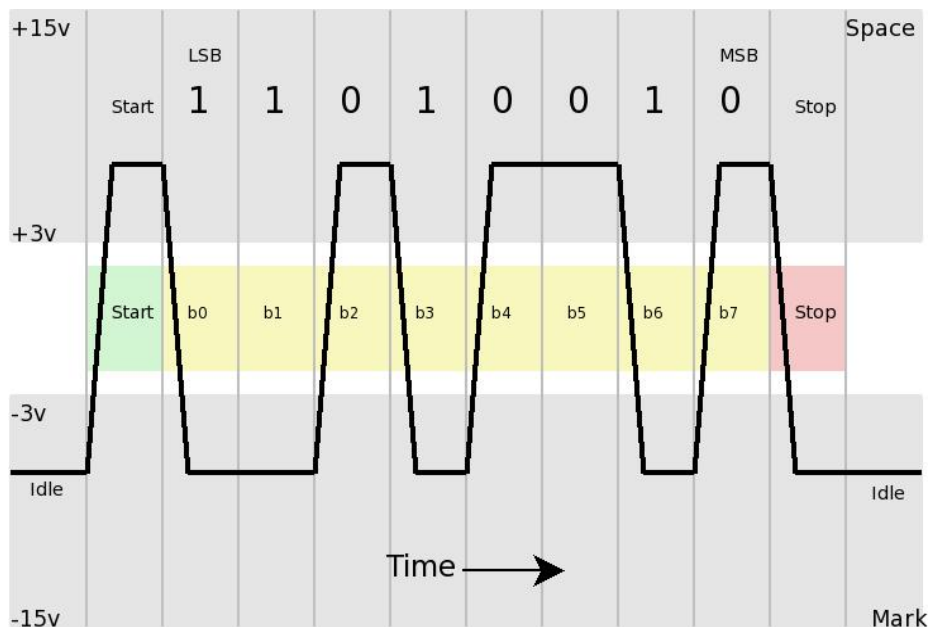
In questa modalità, l'USART utilizza il formato standard di comunicazione NRZ (*Not Return to Zero*): un bit di start, 8/9 bit di dati (a seconda delle impostazioni), un bit di stop. Il più comune formato di dati è a 8 bit e anche quello utilizzato da noi rispetta queste caratteristiche.

L'USART trasmette e riceve per primo l'LSB (Low Significant Bit). I trasmettitori e ricevitori sono funzionalmente indipendenti ma devono utilizzare gli stessi Baud-rate e formato dati per funzionare correttamente.

Il generatore di Baud-rate produce un clock che può essere 16 volte o 64 volte la velocità di shift dei bit, dipendente dal bit BRGH (bassa velocità o alta velocità).

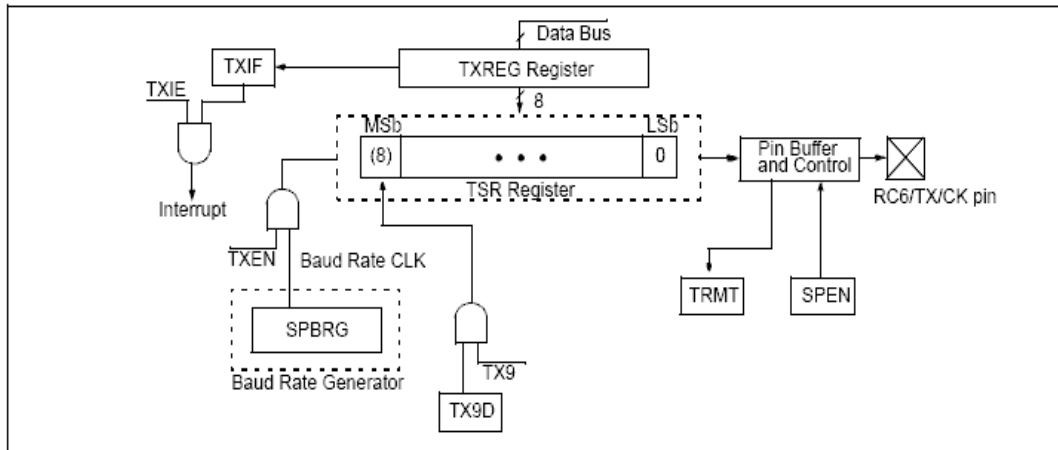
La parità e il relativo bit non è supportata dall'hardware del PIC, ma può essere implementata nel software (immagazzinata nel nono bit dati).

La modalità ASINCRONA è ottenuta mettendo a 0 il bit SYNC.



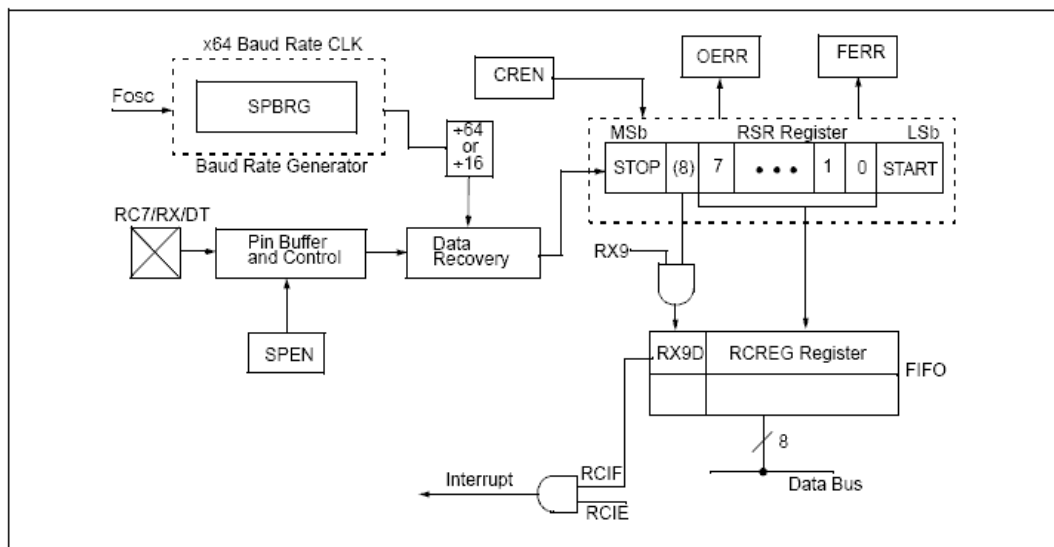
Trasmettitore asincrono dell'USART

Il cuore del trasmettitore è il registro di trasmissione TSR (Transmit Shift Register). Questo registro ottiene i suoi dati dal buffer di scrittura/lettura TXREG, che è caricato con i dati tramite il software. Il TSR non viene caricato finché il bit di STOP non è stato trasmesso. Non appena il bit è stato trasmesso, il TSR viene caricato con i nuovi dati dal registro TXREG, se questo li presenta. Una volta fatto questo, TXREG è vuoto e il bit di flag TXIF è settato. Questo interrupt può essere abilitato o disabilitato settando o resettando il bit TXIE. Questo verrà resettato solo quando nuovi dati verranno caricati nel registro TXREG.



Ricevitore asincrono dell'USART

I dati vengono ricevuti sul pin RC7/RX/DT e mandati al blocco di recupero dati. Il modulo USART ha una speciale preparazione per la comunicazione a più processi.



CONVERTITORE A/D

Il convertitore Analogico digitale ha 5 pin disponibili su cui può operare (nella versione del PIC a 28 pin). Gli ingressi analogici caricano il condensatore di un sample and hold interno e l'uscita del sample and hold interno è l'input per il convertitore vero e proprio.

Il convertitore a questo punto genera un valore digitale di 10 bit (risoluzione del convertitore) per questo livello di segnale analogico con la tecnica delle approssimazioni successive.

Il modulo A/D ha riferimenti di tensione alti e bassi che sono selezionabili tramite software con combinazioni di V_{DD} , V_{SS} , RA2 e RA3.

Il convertitore A/D ha quattro registri che sono:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

Il registro ADCON0 controlla le operazioni del modulo A/D. Il registro ADCON1 configura le funzioni sui pin dei vari PORT, questo perché i pin devono essere configurati come ingressi Analogici al posto che ingressi Digitali per sfruttare questo dispositivo.

I registri ADRESH e ADRESL contengono i 10 bit del risultato della conversione. Quando la conversione è completa, il risultato è caricato in questi registri e il bit GO/DONE viene resettato, settando invece il flag di interrupt ADIF.

Dopo che il modulo A/D è stato configurato come desiderato, il canale deve essere selezionato prima che la conversione abbia inizio e i canali selezionati devono avere i corrispettivi bit del TRIS impostati come input (=1).

Per come abbiamo inizializzato il convertitore A/D, la risoluzione è: $V_{cc}/2^8 = 5/255 = \mathbf{0,02 \text{ mV/bit}}$

Questi sono i passi da seguire per eseguire una corretta conversione A/D:

- 1) Configurare il modulo A/D:
 - Configurare i pin analogici e i riferimenti (ADCON1)
 - Selezionare il canale da convertire (ADCON0)
 - Selezionare il clock per la conversione (ADCON0)
- 2) Configurare l'interrupt per l'A/D:
 - Resettare il bit ADIF
 - Settare il bit ADIE
 - Settare il bit PEIE
 - Settare il bit GIE
- 3) Attendere il tempo di acquisizione richiesto
- 4) Iniziare la conversione:
 - Settare il bit GO/DONE (ADCON0)
- 5) Aspettare che la conversione sia completa (2 modi):
 - Polling sul bit GO/DONE per controllare quando va a 0
 - Aspettare l'interrupt dal convertitore A/D
- 6) Leggere il risultato della conversione tramite ADRESH e ADRESL insieme, resettando il bit ADIF se necessario
- 7) Per la prossima acquisizione, andare ai punti 1 e 2. Il tempo per la conversione A/D è definito $T_{A/D}$. E' necessario aspettare un valore minimo di $2T_{A/D}$ prima che la conversione successiva abbia inizio.

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

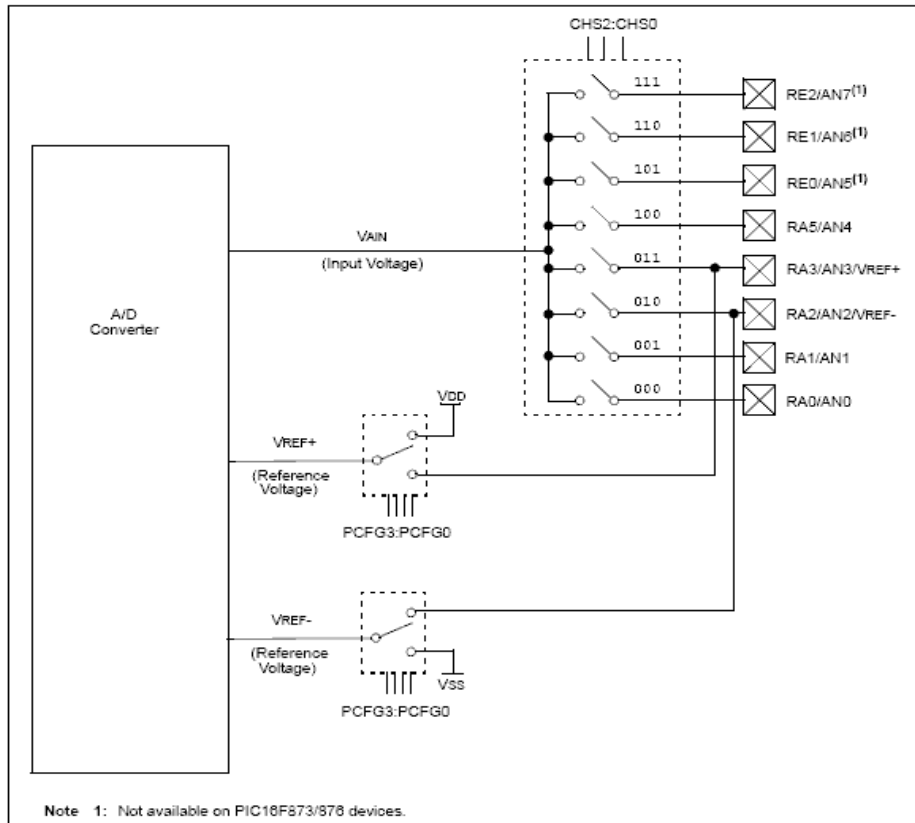
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	

- bit 7-8 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
 00 = Fosc/2
 01 = Fosc/8
 10 = Fosc/32
 11 = FRC (clock derived from the internal A/D module RC oscillator)
- bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
 000 = channel 0, (RA0/AN0)
 001 = channel 1, (RA1/AN1)
 010 = channel 2, (RA2/AN2)
 011 = channel 3, (RA3/AN3)
 100 = channel 4, (RA5/AN4)
 101 = channel 5, (RE0/AN5)⁽¹⁾
 110 = channel 6, (RE1/AN6)⁽¹⁾
 111 = channel 7, (RE2/AN7)⁽¹⁾
- bit 2 **GO/DONE:** A/D Conversion Status bit
If ADON = 1:
 1 = A/D conversion in progress (setting this bit starts the A/D conversion)
 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)
- bit 1 **Unimplemented:** Read as '0'
- bit 0 **ADON:** A/D On bit
 1 = A/D converter module is operating
 0 = A/D converter module is shut-off and consumes no operating current

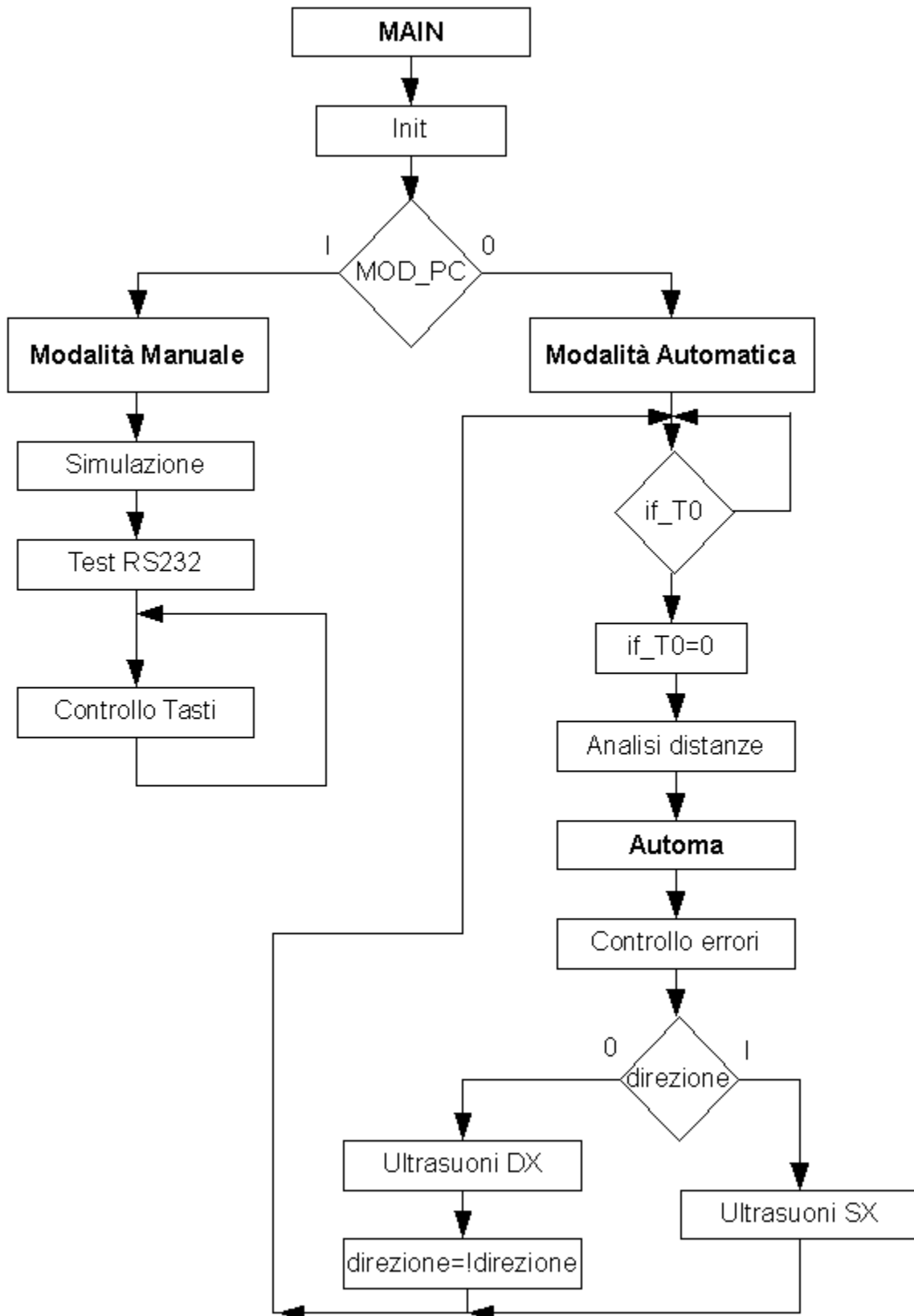
Note 1: These channels are not available on PIC16F873/878 devices.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

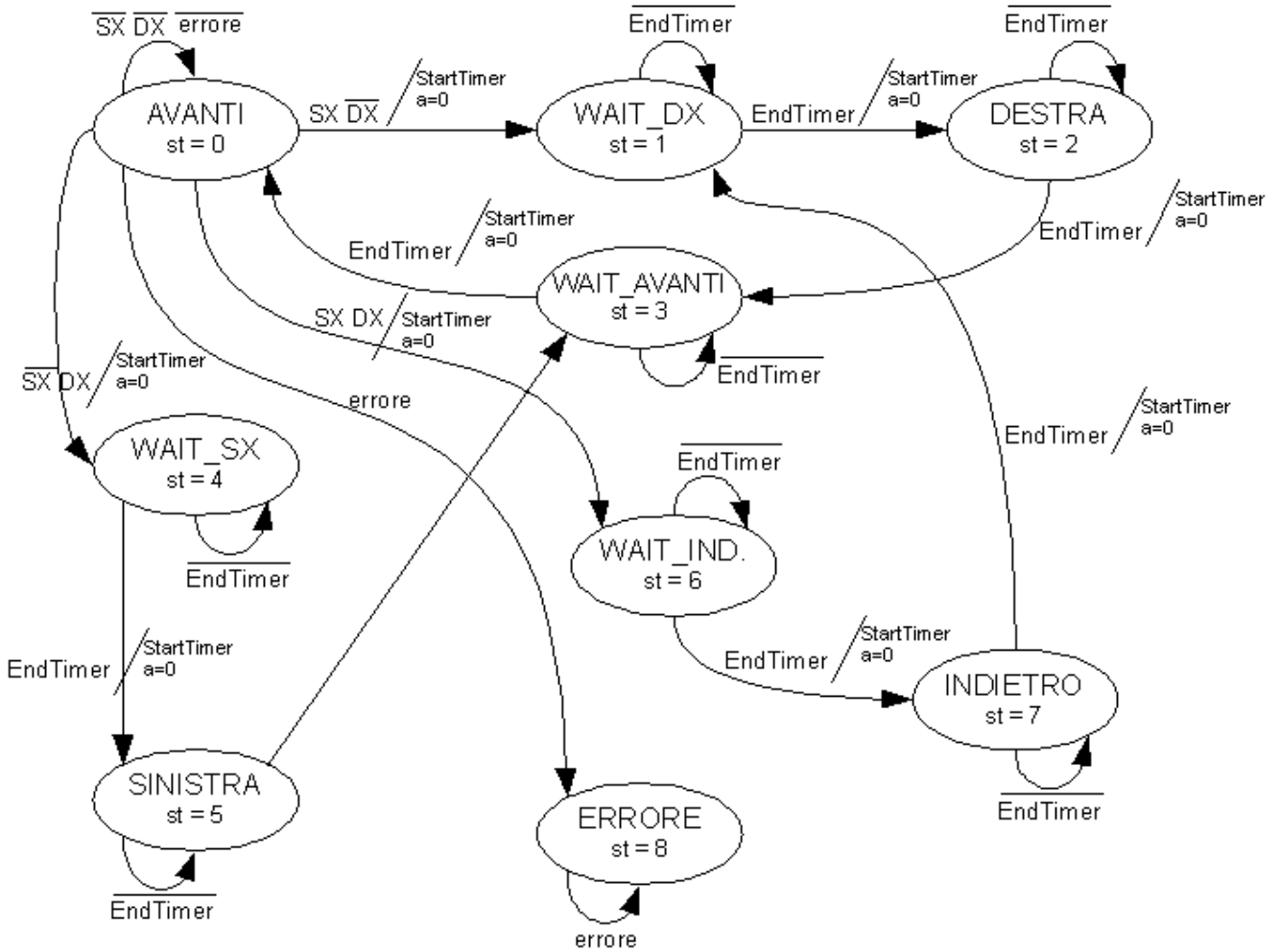
FIGURE 11-1: A/D BLOCK DIAGRAM



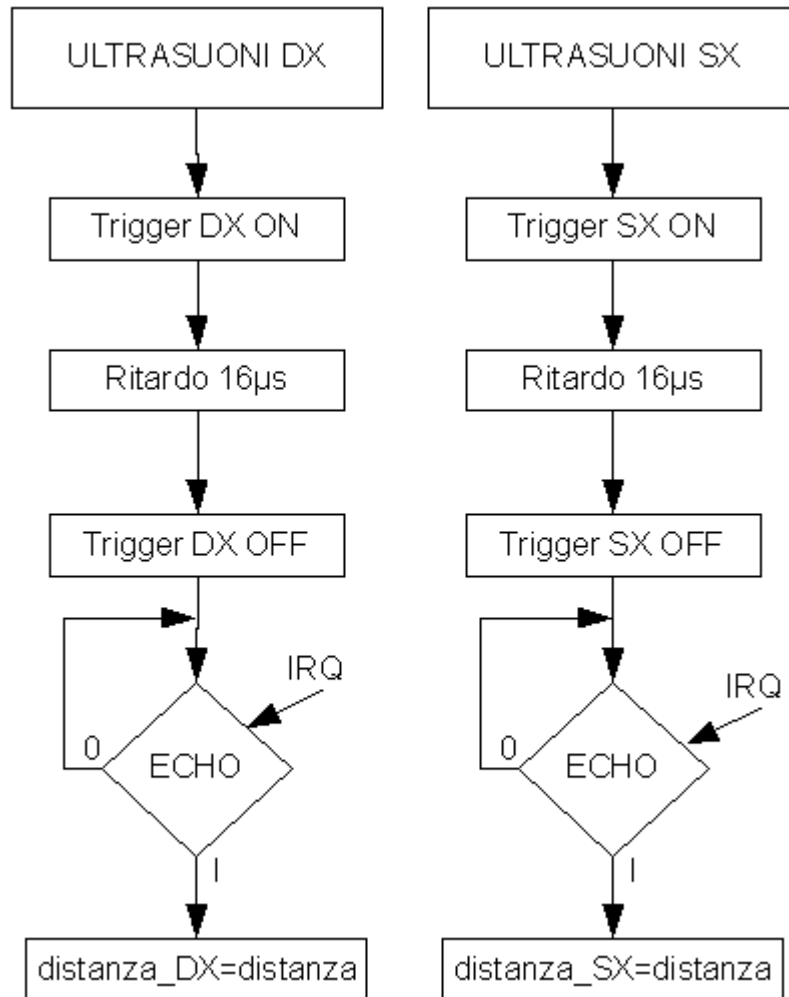
Macchina a Stati e Flow Chart del Programma



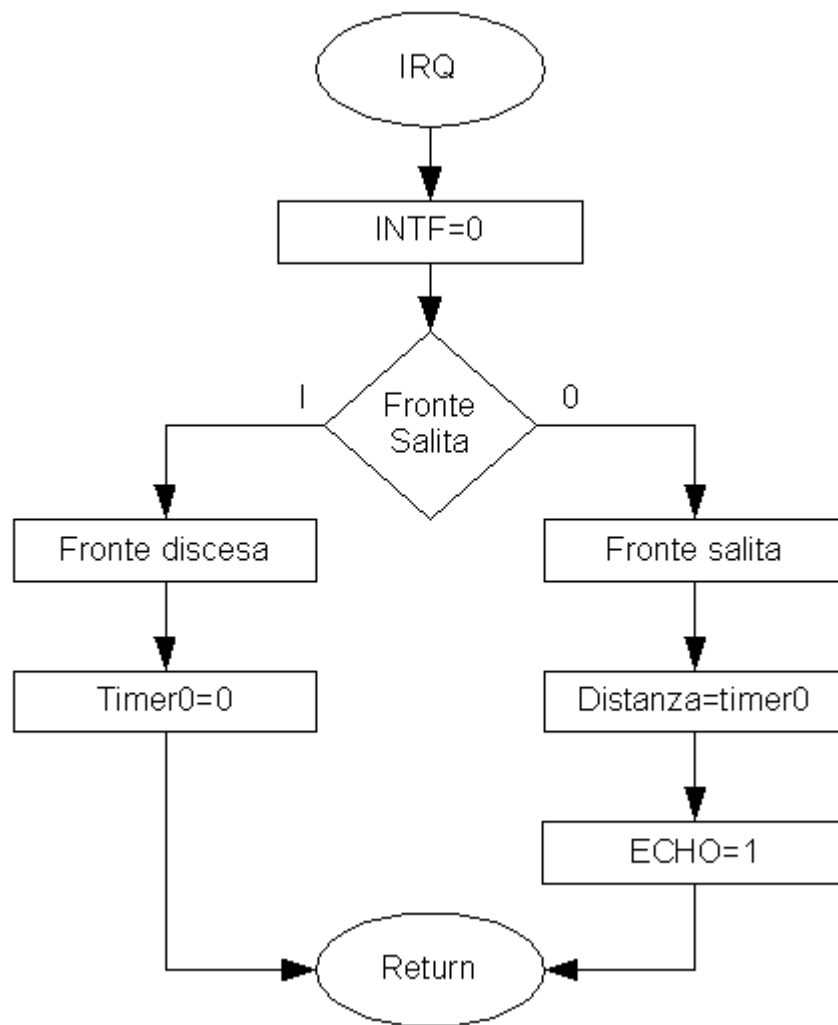
AUTOMA



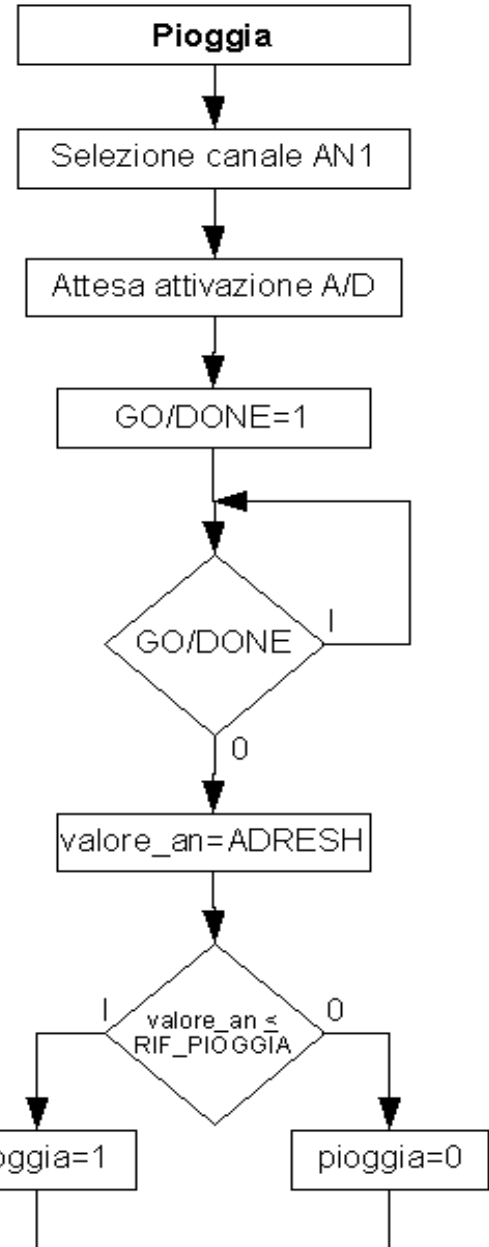
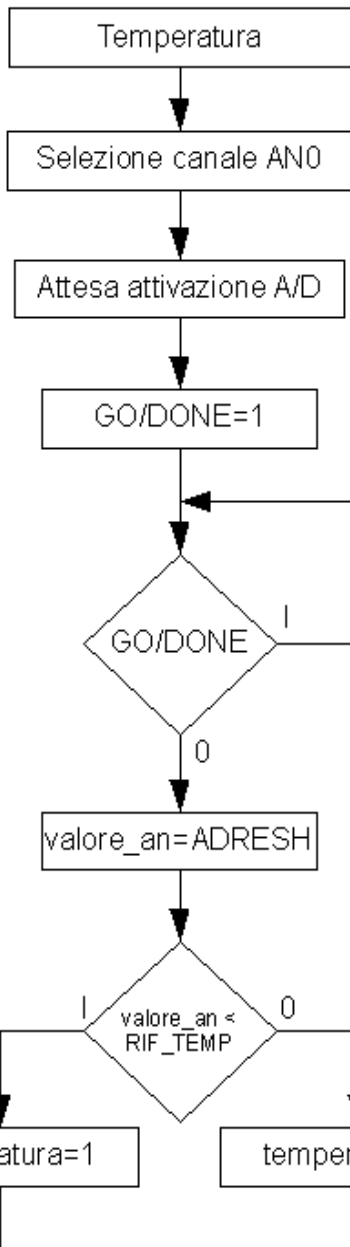
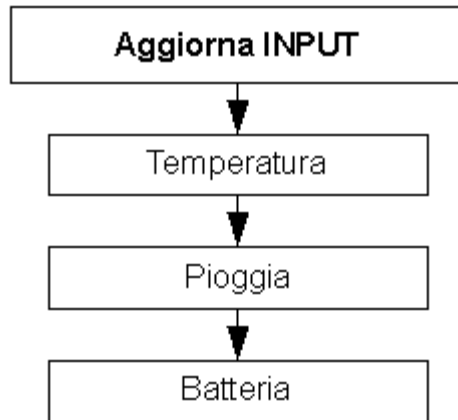
Comando ULTRASUONI

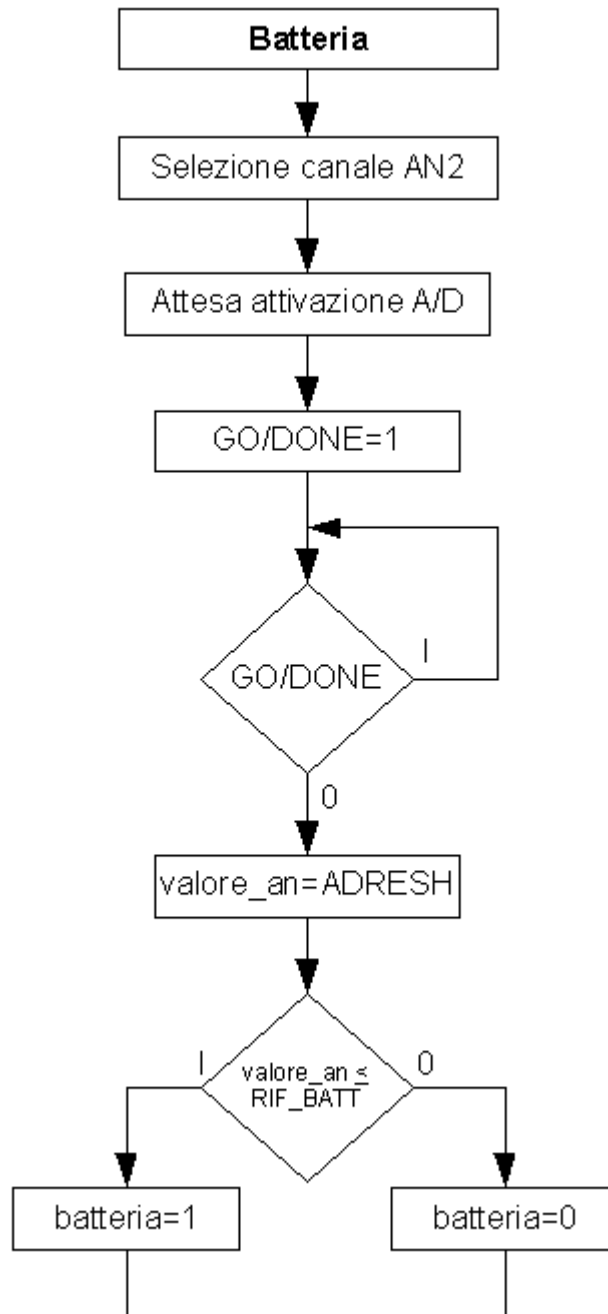


Risposta all'INTERRUPT



Aggiorna INPUT





Software

Il programma del PIC ha costituito la parte più interessante e complicata del progetto.

Si trattava di un argomento a noi in buona parte sconosciuto, riguardo il quale non avevamo molta esperienza. Di conseguenza ci siamo dovuti impegnare al massimo per riuscire con i mezzi a nostra disposizione a creare il programma per il microcontrollore che svolgesse le operazioni al meglio nonostante l'uso di funzioni e procedure poco complesse e articolate.

In questo modo siamo riusciti ad ottenere, dopo innumerevoli adeguamenti in corso d'opera, un programma chiaro, semplice ma allo stesso tempo funzionale e, non da meno, funzionante in ogni suo dettaglio.

Come già anticipato in precedenza il sistema prevede due principali modalità di funzionamento: la prima completamente automatica, con il robot in grado di evitare gli ostacoli eseguendo dei controlli con i moduli ad ultrasuoni e controllando lo stato dei sensori disposti sulla scocca e sulla copertura in Plexiglas in modo da segnalare errori o malfunzionamenti tramite segnali visivi (LED rosso) e acustici (BUZZER).

La seconda modalità è manuale ed è legata alla trasmissione dei comandi per i motori tramite la comunicazione seriale per far eseguire al robot un movimento tele/filo guidato.

In fase di progetto della scheda elettronica era stato predisposto il modulo RS232 e il selettore della modalità (S2), in modo tale che se avessimo avuto ulteriore tempo dopo lo sviluppo della modalità automatica ci saremmo dedicati a questa, ritenuta inizialmente meno importante.

Impegnandoci a fondo e recuperando tutto il tempo possibile durante le ore di laboratorio oltre al tempo speso a casa abbiamo avuto tempo a sufficienza per completare la parte automatica che ricercavamo nell'obiettivo di questo progetto e in più l'aggiunta delle basi per una seconda modalità che risulta, seppur semplice, funzionante e soddisfacente per il nostro grado di esperienza nella programmazione ad alto livello, ricorrendo in minima parte ad aiuti esterni.

Per la programmazione ci siamo affidati al tool di sviluppo di proprietà della MICROCHIP chiamato MPLAB, scegliendo tra i vari compilatori per linguaggio C il software CC5X della Knudsen.

Nelle pagine successive è riportato l'intero listato del programma con i commenti laterali che spiegano nel dettaglio le varie particolarità di certi comandi o funzioni utilizzate.

La macchina a stati del sistema è stata regolata con un tempo di 50ms, che normalmente è un tempo abbastanza lungo per un dispositivo come il PIC, ma si è reso necessario per l'utilizzo dei moduli ad ultrasuoni che potevano avere una risposta massima di 30ms.

```

/*****

```

```

I.T.I.S. BENEDETTO CASTELLI BRESCIA

```

```

A.S. 2008-2009

```

```

CLASSE 5^ SEZIONE F

```

```

PROGETTO D'ESAME

```

```

Aluni: INVERNICI, PROVENZANO, SANSONI

```

```

TAGLIAERBA AUTOMATICO

```

```

PIC16F876

```

```

Xtal 4Mhz

```

```

Note: Oggetto deambulante munito di estirpatore vegetale

```

```

*****/

```

```

#pragma chip PIC16F876

```

```

#include "int16CXX.h"

```

```

// ----- Costanti -----

```

```

#define CONFIG &=0x0000, |=0x3F7A //inizializzazione registro configuration word

```

```

#pragma config CONFIG

```

```

#define RIF_TEMP      0b11100000      //soglia 11,7V NTC

```

```

#define RIF_BATT      0b10000010      //soglia 10,5V BATTERIA

```

```

#define RIF_PIOGGIA   0b10100000      //soglia 3,13V PIOGGIA

```

```

#define DIST_MIN      0x07AA          //soglia di 25 cm circa (dipende anche dall'angolo dei sensori)

```

```

// ----- assegnazione port e macro -----

```

```

#define REL1          PORTB.4

```

```

#define REL2          PORTB.3

```

```

#define REL3          PORTB.2

```

```

#define REL4          PORTB.1

```

```

#define BUZZER        PORTC.2          //comando per Buzzer

```

```

#define MOD_PC        PORTC.5          //ingresso modalit  funzionamento

```

```

#define TX_232        PORTC.6          //TX seriale RS232

```

```

#define RX_232        PORTC.7          //RX seriale RS232

```

```

#define ERRORE1_ON    PORTB.6=1;      //led giallo on      //viene utilizzato come monitor per
#define ERRORE1_OFF   PORTB.6=0;      //led giallo off     //la scansione della macchina a stati

```

```

#define ERRORE2_ON    PORTB.7=1;      //led rosso on

```

```

#define ERRORE2_OFF   PORTB.7=0;      //led rosso off

```

```

#define LAMA_ON       PORTB.5=1;      //LAMA=1

```

```

#define LAMA_OFF      PORTB.5=0;      //LAMA=0

```

```

#define TRIGGER_DX_ON PORTC.0=1;

```

```

#define TRIGGER_DX_OFF PORTC.0=0;

```

```

#define TRIGGER_SX_ON PORTC.1=1;

```

```

#define TRIGGER_SX_OFF PORTC.1=0;

```

```

#define MOT_SX_STOP   REL1=0 ; REL2=0;

```

```

#define MOT_SX_AVANTI REL1=1 ; REL2=0;

```

```

#define MOT_SX_INDIETRO REL1=0 ; REL2=1;

```

```

#define      MOT_DX_STOP          REL3=0 ; REL4=0;
#define      MOT_DX_AVANTI       REL3=1 ; REL4=0;
#define      MOT_DX_INDIETRO     REL3=0 ; REL4=1;

//----- definizione funzioni-----

void init(void);
void automa(uns8 s1);
void controlloerrori(void);
void ultrasuoni_dx(void);
void ultrasuoni_sx(void);
void analisi_dist(void);
void aggiorna_input(void);
void AVANTI(void);
void WAIT_DX(void);
void DESTRA(void);
void WAIT_AVANTI(void);
void WAIT_SX(void);
void SINISTRA(void);
void WAIT_INDIETRO(void);
void INDIETRO(void);
void ERRORE(void);
void simulazione (void);
void test_RS232 (void);
void tx_chr(uns8 n);
uns8 rd_ser();

//----- definizione variabili-----

uns8  valore_an,cont,a,p,rx_data,buff_rx,st,s1;      //registri a 8 bit
uns16 i,distanza,distanza_DX,distanza_SX;          //registri a 16 bit
bit   if_t0,RX_FLAG,pioggia,temperatura,batteria,errore,sonar,SX,DX,EndTimer,direzione,errore_sonar,primo;

/* -----
FUNZIONE      : INTERRUPT

DESCRIZIONE   : Risponde alla richiesta di Interrupt

----- */

#pragma origin 4
interrupt int_server(void)
{
    int_save_registers          //salvataggio registri
    if(T0IF)                   //interrupt TMR0
    {
        T0IF = 0;              //reset flag interrupt TMR0
        TMR0 = 52;              //periodo di 50ms
        if_t0 = 1;              //setta flag interrupt timer0 per automi
        if (sonar)
        {
            if (direzione)      {
                distanza_SX=distanza;
                errore_sonar=0;
            }
            else {
                distanza_DX=distanza;
                errore_sonar=0;
            }
            sonar=0;
        }
        else {errore_sonar=1;}
    }
}

```



```

    if (INTF)                                // azzera FLG IRQ del TMR0
    {
        INTF=0;
        if (INTEDG)
        {
            INTEDG =0;                       // inverte il fronte su RB0
            TMR1H=0;                           //TMR1=0 (registro alto + basso)
            TMR1L=0;
        }
        else
        {
            INTEDG =1;
            distanza = TMR1H;
            distanza= distanza << 8;
            distanza=distanza|TMR1L;
            sonar=1;
        }
    }

int_restore_registers        //ripristino registri
}

/* -----
FUNZIONE      : Main
DESCRIZIONE   : Programma principale
----- */

void main(void)
{
    init();
    if (!MOD_PC)
    {
        while(1)
        {
            while(!if_t0);                    //attesa clock per macchina a stati
            if_t0 = 0;                          //reset del flag
            if (primo)
            {
                PORTB.6=!PORTB.6;
                analisi_dist();
                automa(st);                     //macchina a stati movimento
                controlloerrori();             //verifica condizioni ambientali
            }
            primo=1;
            if (direzione)
                ultrasuoni_sx();               //gestione ultrasuoni
            else
                ultrasuoni_dx();
            direzione=!direzione;
        }
        else simulazione();
    }
}

/* -----
FUNZIONE      : Inizializzazione
DESCRIZIONE   : Impostazioni iniziali del sistema
----- */

```

```

void init(void)
{
  ADCON0 = 0b01000001;
  ADCON1 = 0b00000010;
  OPTION_REG = 0b11000111;
  INTCON = 0xE0;
  INTEDG = 1;
  TMR0 = 52;
  T1CON = 0b01000001;
  TMR1IE = 0;

  TRISA = 0b11110111;
  PORTA = 0x00;

  TRISB = 0b00000001;
  PORTB = 0x00;

  TRISC = 0b10111000;
  PORTC = 0x00;

  if_t0 = 0;
  cont = 0;
  st = 0;
  EndTimer = 0;

  BUZZER = 0;
  SX = 0;
  DX = 0;
  distanza_DX = 0xFFFF;
  distanza_SX = 0xFFFF;
  direzione = 0;
  errore_sonar = 1;
  sonar = 0;
  primo = 0;
  a = 0;

  // ----- init IRQ MASK -----

  INTCON = 0;
  //RCON = 0;
  TOIE=1;
  INTE=1;
  // RCIE = 1;
  // PIE2=0;
  PEIE = 1;
  GIE = 1;

  // ----- init USART -----

  TXIF=0;
  RCIF=0;
  SPBRG=25;
  SYNC=0;
  BRGH=1;
  TXEN=1;
  SPEN=1;
  CREN=1;
  RX_FLAG=0;
}

//----- determinazione stato automa -----
void automa(uns8 s1)
{
  switch (s1)
  {

```

```

    case 0: AVANTI();
                break;
    case 1: WAIT_DX();
                break;
    case 2: DESTRA();
                break;
    case 3: WAIT_AVANTI();
                break;
    case 4: WAIT_SX();
                break;
    case 5: SINISTRA();
                break;
    case 6: WAIT_INDIETRO();
                break;
    case 7: INDIETRO();
                break;
    case 8: ERRORE();
                break;
}
}

//----- procedura corrispondente allo stato 0-----
void AVANTI(void)
{
    if (errore)
        st=8;
    else
    {
        if (SX)
        {
            if (DX) {
                st=6;
            }
            else {
                st=1;
            }
        }
        else
        {
            if (DX) {
                st=4;
            }
            else {
                st=0;
            }
        }
    }
    BUZZER=0;
    MOT_SX_AVANTI; //marcia in avanti
    MOT_DX_AVANTI;
    LAMA_ON;
    //ERRORE1_OFF;
    ERRORE2_OFF;
}

//----- procedura corrispondente allo stato 1-----
void WAIT_DX(void)
{
    if (EndTimer)
    {
        EndTimer=0;
        st=2;
    }
    else
    {
        st=1;
    }
}

```

```

        if (a!=10)
            a++;
        else {
            a=0;
            EndTimer=1;
        }
    }
    BUZZER=0;
    MOT_DX_STOP;
    MOT_SX_STOP;
}

//----- procedura corrispondente allo stato 2-----
void DESTRA(void)
{
    if (EndTimer)
    {
        EndTimer=0;
        st=3;
    }
    else
    {
        st=2;
        if (a!=10)
            a++;
        else {
            a=0;
            EndTimer=1;
        }
    }
    BUZZER=!BUZZER;
    MOT_SX_AVANTI;
    MOT_DX_INDIETRO;
}

//----- procedura corrispondente allo stato 3-----
void WAIT_AVANTI(void)
{
    if (EndTimer)
    {
        st=0;
        EndTimer=0;
    }
    else
    {
        st=3;
        if (a!=10)
            a++;
        else {
            a=0;
            EndTimer=1;
        }
    }
    BUZZER=0;
    MOT_DX_STOP;
    MOT_SX_STOP;
}

//----- procedura corrispondente allo stato 4-----
void WAIT_SX(void)
{
    if (EndTimer)
    {
        st=5;
        EndTimer=0;
    }
    else

```

```

        {
            st=4;
            if (a!=10)
                a++;
            else {
                a=0;
                EndTimer=1;
            }
        }
    BUZZER=0;
    MOT_DX_STOP;
    MOT_SX_STOP;
}

//----- procedura corrispondente allo stato 5-----
void SINISTRA(void)
{
    if (EndTimer)
    {
        st=3;
        EndTimer=0;
    }
    else
    {
        st=5;
        if (a!=10)
            a++;
        else {
            a=0;
            EndTimer=1;
        }
    }
    BUZZER=!BUZZER;
    MOT_SX_INDIETRO;
    MOT_DX_AVANTI;
}

//----- procedura corrispondente allo stato 6-----
void WAIT_INDIETRO(void)
{
    if (EndTimer)
    {
        st=7;
        EndTimer=0;
    }
    else
    {
        st=6;
        if (a!=10)
            a++;
        else {
            a=0;
            EndTimer=1;
        }
    }
    BUZZER=0;
    MOT_DX_STOP;
    MOT_SX_STOP;
}

//----- procedura corrispondente allo stato 7-----
void INDIETRO(void)
{
    if (EndTimer)
    {

```

```

        st=1;
        EndTimer=0;
    }
else
    {
        st=7;
        if (a!=20) //la retromarcia è l'unico stato che prevede un tempo più lungo
            a++;
        else {
            a=0;
            EndTimer=1;
        }
    }
    BUZZER=!BUZZER;
    MOT_SX_INDIETRO;
    MOT_DX_INDIETRO;
}

//----- procedura corrispondente allo stato 8-----
void ERRORE(void)
{
    MOT_DX_STOP;
    MOT_SX_STOP;
    LAMA_OFF;
    BUZZER=1;

    if (batteria)
    {
        ERRORE2_ON;
        //ERRORE1_ON;
    }
    else if (pioggia)
    {
        ERRORE2_ON;
        //ERRORE1_OFF;
    }
    else if (temperatura)
    {
        ERRORE2_ON;
        //ERRORE1_ON;
    }
    else st=0;
}

//----- controlloerrori -----

void controlloerrori(void)
{
    aggiorna_input();

    if(batteria) //controllo tensione batteria
    {
        errore=1;
    }
    else if(temperatura) //controllo surriscaldamento
    {
        errore=1;
    }
    else if(pioggia) //controllo sensore pioggia
    {
        errore=1;
    }
    else //nessun errore
    {
        errore=0;
    }
}

```

```

    }

}

//----- ultrasuoni DX-----

void ultrasuoni_dx(void)
{
    TRIGGER_DX_ON;
    i=0;
    while (i<1)      {          //ritardo 16us circa
                        i++;
                    }
    TRIGGER_DX_OFF;

    INTEDG=1;
}

//----- ultrasuoni SX -----

void ultrasuoni_sx(void)
{
    TRIGGER_SX_ON;
    i=0;
    while (i<1)      {          //ritardo 16us circa
                        i++;
                    }
    TRIGGER_SX_OFF;

    INTEDG=1;
}

//----- analisi distanze -----

void analisi_dist(void)
{
    if (!errore_sonar)
    {
        if(distanza_SX<DIST_MIN)
            SX=1;
        else
            SX=0;
        if(distanza_DX<DIST_MIN)
            DX=1;
        else
            DX=0;
    }
    else {
        SX=0;
        DX=0;
    }
}

/* -----
FUNZIONE      : Aggiorna_input

DESCRIZIONE   : Legge lo stato degli ingressi (NTC, Pioggia, Batteria) e aggiorna le variabili;
                devo convertire gli input analogici e conf. con soglie per attivare
                correttamente i bit batteria, pioggia, temperatura.

PAGE          : 0
----- */

void aggiorna_input(void)

```

```

{
ADCON0=0b01000001; // Selezione canale AN0 per lettura NTC
#asm // Wait
    nop
    nop
    nop
    nop
    nop
    nop
#endasm
GO=1; // Start conversione
while (GO);
valore_an=ADRESH;
if (valore_an<RIF_TEMP)
{
    temperatura=1;
}
else
{
    temperatura=0;
}

//temperatura=0;

ADCON0=0b01001001; // Selezione canale AN1 per lettura PIOGGIA
#asm // Wait
    nop
    nop
    nop
    nop
    nop
    nop
#endasm
GO=1; // Start conversione
while (GO);
valore_an=ADRESH;
if (valore_an<=RIF_PIOGGIA)
{
    pioggia=1;
}
else
{
    pioggia=0;
}
// pioggia=0;

ADCON0=0b01010001; // Selezione canale AN2 per lettura BATT
#asm // Wait
    nop
    nop
    nop
    nop
    nop
    nop
#endasm
GO=1; // Start conversione
while (GO);
valore_an=ADRESH;
if (valore_an<=RIF_BATT)
{
    batteria=1;
}
else
{
    batteria=0;
}

```



```

    }
//batteria=0;

}

/* -----
FUNZIONE      : Simulazione via RS232

DESCRIZIONE   :

PAGE          : 0
----- */

void simulazione (void)
{
    //test_RS232();

    while(1)
    {
        rx_data=rd_ser();

        #asm                                // Wait
            nop
            nop
            nop
            nop
            nop
            nop
        #endasm

        if (rx_data=='W')
        {
            MOT_DX_AVANTI;
            MOT_SX_AVANTI;
            LAMA_ON;
            tx_chr('1');
        }
        else
        {
            if (rx_data=='D')
            {
                MOT_SX_INDIETRO;
                MOT_DX_AVANTI;
                LAMA_ON;
                tx_chr('2');
            }
            else
            {
                if (rx_data=='S')
                {
                    MOT_DX_INDIETRO;
                    MOT_SX_INDIETRO;
                    LAMA_ON;
                    tx_chr('3');
                }
                else {
                    if(rx_data=='A')
                    {
                        MOT_DX_INDIETRO;
                        MOT_SX_AVANTI;
                        LAMA_ON;
                        tx_chr('4');
                    }
                    else {
                        MOT_DX_STOP;
                        MOT_SX_STOP;
                    }
                }
            }
        }
    }
}

```

```
LAMA_OFF;
tx_chr('5');
```

```

    }
    }
}
}
```

```
/* -----
```

FUNZIONE : TEST_RS232
DESCRIZIONE : Trasmette l'intero set di caratteri al PC
PAGE : 0

```
----- */
```

```
void test_RS232 (void)
{ int p;
  for(p=0; p<= 255; p++)
    tx_chr(p);
}
```

```
/* -----
```

FUNZIONE : TX_CHR
DESCRIZIONE : Trasmette un carattere ASCII
PAGE : 0

```
----- */
```

```
void tx_chr(uns8 n)
RS232
{
  TXREG=n;
  while(!TRMT);
}
```

// Trasmette il carattere in INPUT sul canale seriale

```
/* -----
```

FUNZIONE : rd_ser
DESCRIZIONE : Ricezione carattere ASCII
PAGE : 0

```
----- */
```

```
uns8 rd_ser()
{
  while(!RCIF);
  return RCREG;
}
```

```
/* -----
```

FINE

```
----- */
```

Motori in corrente continua

Funzionamento generale di un motore in CC

Il **motore in corrente continua** (brevemente **motore in CC**) è stato il primo motore elettrico realizzato, ed è tuttora utilizzato ampiamente per piccole e grandi potenze, inoltre tale motore può funzionare da dinamo. Sono a corrente continua (o comunque alimentabili in corrente continua) numerosi motori di piccola potenza per usi domestici, come anche motori per trazione ferroviaria e marina della potenza di molte centinaia di kW.

Motore a spazzole

I motori degli alzavetri da noi impiegati sono esempi di classici motori in corrente continua che hanno una parte che gira detta appunto rotore o anche armatura e una parte che genera un campo magnetico fisso (nell'esempio i due magneti colorati) detta statore. Un interruttore rotante detto commutatore o collettore a spazzole inverte due volte ad ogni giro la direzione della corrente elettrica che percorre i due avvolgimenti generando un campo magnetico che entra ed esce dalle parti arrotondate dell'armatura. Nascono forze di attrazione e repulsione con i magneti permanenti fissi (indicati con N ed S nelle figure).

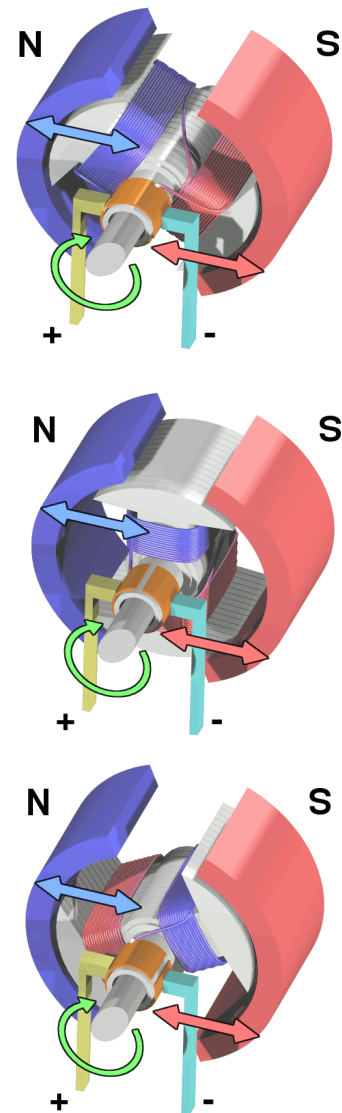
La velocità di rotazione dipende da:

- Tensione applicata.
- Corrente assorbita dal rotore.
- Carico applicato.

La coppia generata è proporzionale alla corrente ed il controllo più semplice agisce sulla tensione d'alimentazione, mentre nei sistemi più complessi si usa un Controllo automatico in retroazione che legge le variabili per generare la tensione da applicare al motore. Il motore CC a magneti permanenti ha un comportamento reversibile: diventa un generatore di corrente continua se si collega un altro motore all'albero. Si può allora prelevare l'energia elettrica prodotta collegandosi alle spazzole, inoltre da questo si può intuire la sua capacità di agire anche da freno e/o usato per il recupero dell'energia nei mezzi ibridi o essere semplificato come un semplice freno: applicando tra le spazzole un resistore l'energia meccanica trasmessa all'albero si dissipa su questo resistore). Riassumendo si può affermare che il motore CC ha tutte le funzioni necessarie per un mezzo mobile: oltre alla funzione di motore può recuperare l'energia funzionando da dinamo quando serve l'azione frenante o agire semplicemente da freno.

Motoriduzione

In associazione a questo motore c'è un apparato meccanico contenente ingranaggi plastici che aumenta la coppia rotante a discapito della velocità (motoriduzione); questo tuttavia non comporta alcuna modifica ai consumi, ma cambia solamente il rapporto di accoppiamento tra motore e ruote, con un'azione puramente meccanica.

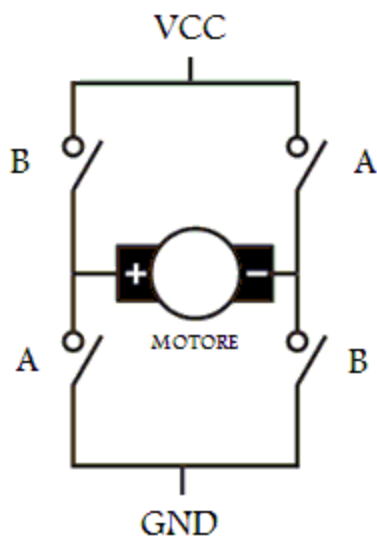


Circuito di comando per movimento con ponte H a RELE'

I motori richiedono una potenza molto elevata, inoltre non avevamo bisogno di regolarne la velocità ma ci era sufficiente comandarli con un metodo ON/OFF.

E' stato così necessario scegliere una struttura adeguata: a nostro giudizio il metodo migliore e meno complicato nonché economico è risultato essere l'utilizzo di quattro relè i cui contatti sono stati disposti a formare due ponti ad H in modo da utilizzare meno uscite possibili del microcontrollore. Infatti per i due motori di movimento siamo riusciti a utilizzare solo quattro uscite, permettendo ai motori tre combinazioni di moto (avanti, indietro e fermo), altrimenti impossibili da realizzare con così poche uscite.

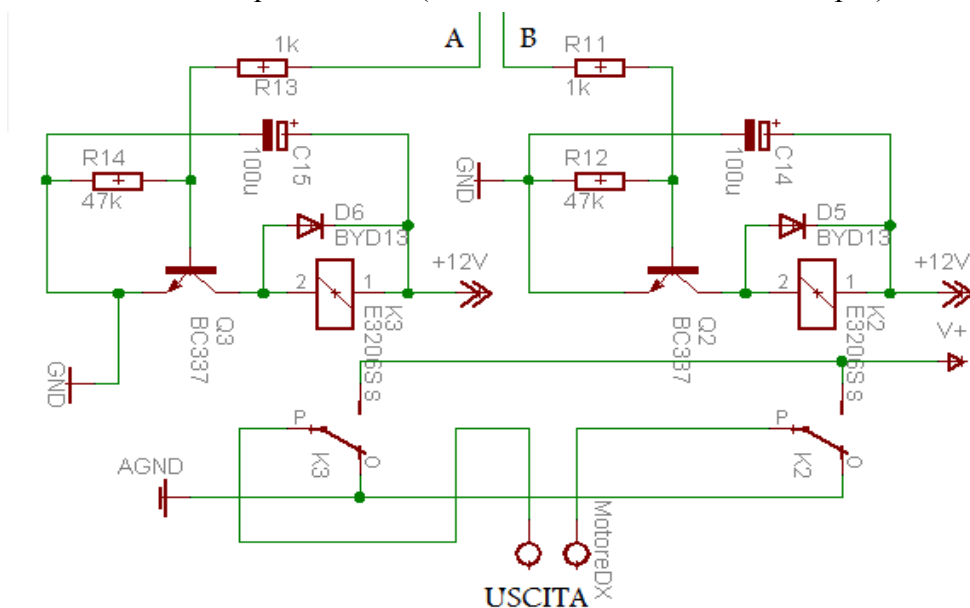
Il circuito tipico di un ponte ad H ha queste caratteristiche:



I contatti con la stessa denominazione agiscono in sincronia: in questo modo bastano due linee di comando per avere lo scorrimento della corrente in un senso attraverso il motore piuttosto che nel senso opposto.

A	B	DIREZIONE
0	0	Circuito aperto
0	1	+ ► -
1	0	+ ◄ -
1	1	Non deve mai verificarsi!

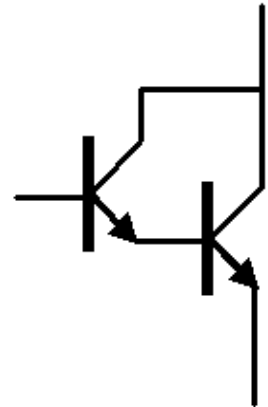
Il nostro circuito è realizzato in questo modo (controllo motore destro di esempio):



Circuito di comando Lama con darlington BDX53

Per comandare il terzo motore di minore potenza e di un solo verso di rotazione, corrispondente alla lama del Tosaerba, abbiamo deciso di non utilizzare un ulteriore RELE' bensì un Transistor di potenza di tipo Darlington NPN siglato BDX53 (anche per fini didattici).

In elettronica, un transistor Darlington (spesso chiamato anche coppia Darlington) è un transistor composito costituito da due transistor a giunzione bipolari in cascata. La caratteristica principale di questa configurazione è l'elevatissimo guadagno in corrente (indicato con β o h_{FE} , che nel caso del BDX53 è di 750 come valore minimo), pari al prodotto dei guadagni in corrente dei singoli transistor; i transistor Darlington possono essere realizzati mediante componenti discreti oppure integrati; sono realizzati anche a gruppi, riuniti in un unico package.



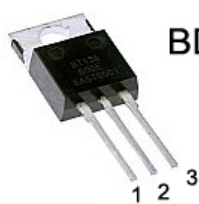
Il circuito necessita solo di un pin di abilitazione per essere comandato (funzionamento ON/OFF).

Il Darlington ha bisogno, per definire la corrente erogata, di una resistenza collegata sulla sua base, che abbiamo dimensionato seguendo le specifiche riportate sul datasheet; la corrente in uscita è circa 750 volte la corrente di base, poiché tale è il guadagno del dispositivo, quindi la formula da seguire è:

$$I_{OUT} = (V_{CC}/R_B) * 750$$

Utilizzando il segnale di ampiezza 5V proveniente dal PIC e volendo ottenere una corrente di almeno 0,7 Ampere in uscita otterremo:

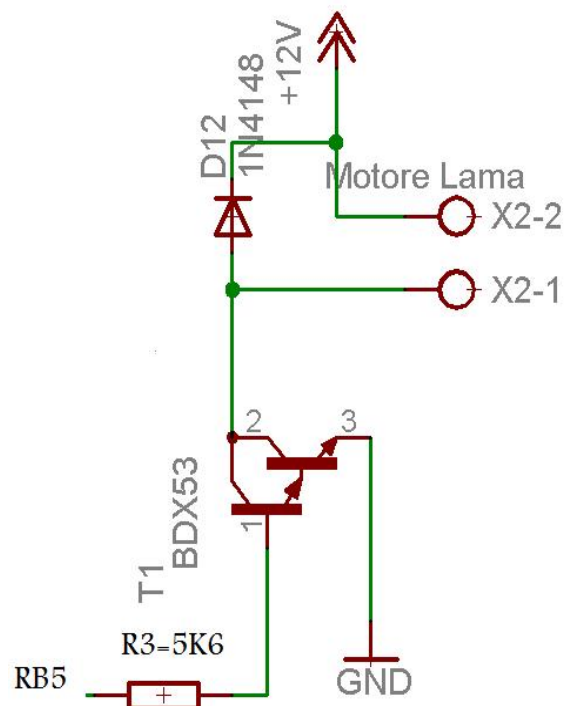
$$R_B = (V_{CC}/I_{OUT}) * 750 = 5V/0,7A * 750 = 5357 \Omega \quad \text{normalizzato per la serie E12 a } \mathbf{5600 \Omega}$$



BDX53F

- 1 > Base
- 2 > Collector
- 3 > Emitter

La corrente che scorre all'interno di questo dispositivo non comporta nessun riscaldamento particolare. Il BDX53 quindi non necessita di alcun mezzo di dissipazione di calore.



Moduli ad ultrasuoni

SRF05 - Ultra-Sonic Ranger

Introduzione

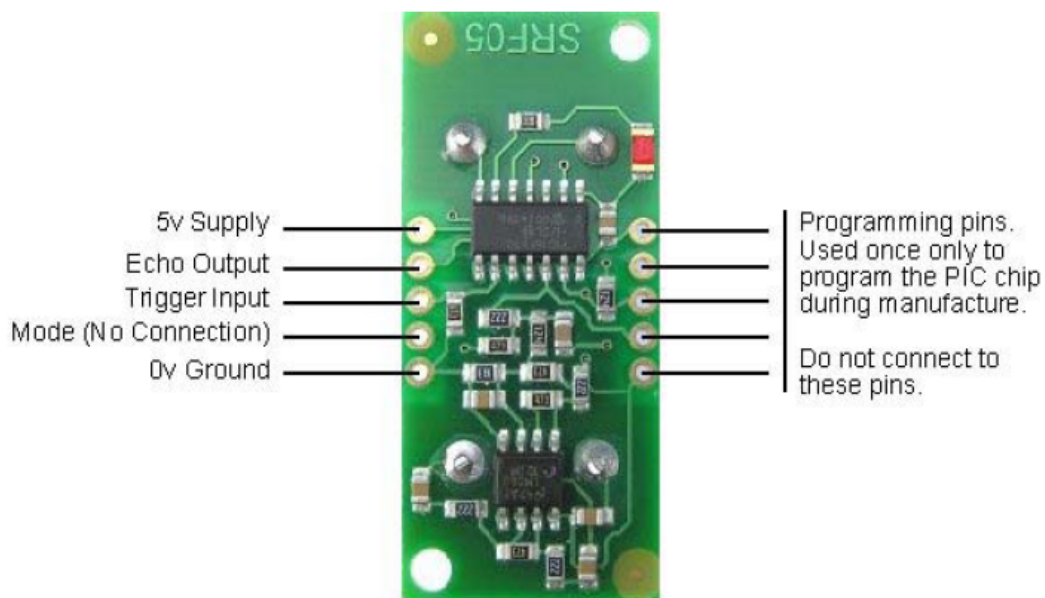
Questo dispositivo presenta un circuito gestito da microcontrollore che invia un segnale ad ultrasuoni e riceve l'eco prodotto dal ritorno delle onde acustiche riflesse dagli oggetti tramite le due apposite capsule di ricezione e trasmissione poste sul retro.

Il Range di distanza che il sensore è in grado di coprire è di 4 metri.

Modalità utilizzata

Separate Trigger and Echo

Questo sensore è in grado di operare in due differenti modalità. La prima, da noi scelta, prevede l'utilizzo di due differenti pin per i segnali di TRIGGER ed ECHO mentre la seconda consente di utilizzare un solo pin per TRIGGER ed ECHO. Per poter utilizzare il sensore nella prima modalità il pin MODE va lasciato sconnesso. L' SRF05 ha un resistore di pull up interno su questo pin.



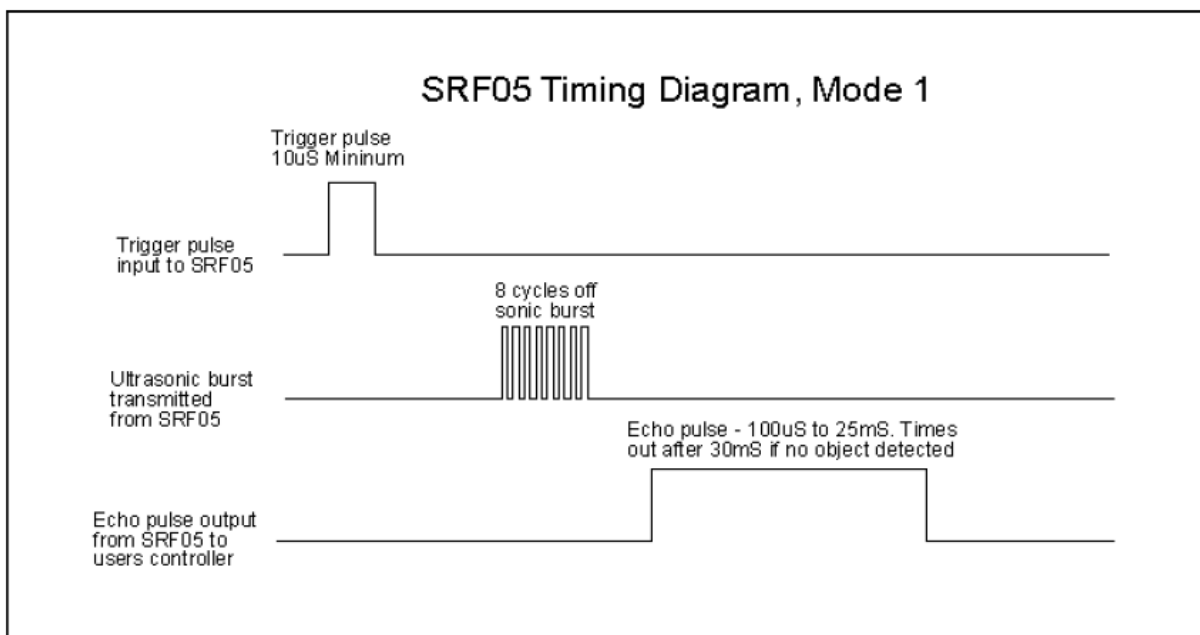
Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

Calcolo della Distanza

Il diagramma temporale (timing diagram) è presente per ogni tipo di modalità, ma noi riportiamo quella utilizzata nel nostro sistema. E' necessario fornire un breve impulso da $10\mu\text{s}$ sul piedino denominato TRIGGER per attivare il modulo. **(Grafico superiore)**

Il modulo SRF05 invierà nell'etere, tramite le capsule, 8 cicli di impulsi a 40 KHz che servono da onda acustica che si rifletterà sull'ambiente e più precisamente sul/sugli oggetti più vicini nel campo di misura. Questo segnale non ha nessuna relazione con il segnale elettrico del TRIGGER o dell'ECHO, ma svolge solo la funzione da sonar per il modulo. **(Grafico centrale)**

Quando le onde acustiche ritornano verso le capsule il modulo pone alta l'uscita di ECHO, il cui periodo alto (Ton) identificherà la distanza dall'oggetto (più l'oggetto è distante e più questo Ton aumenta). Terminata la misura il valore di ECHO ritorna basso. **(Grafico inferiore)**



Misurando il tempo di questo periodo alto è possibile risalire con precisione alla distanza dell'ostacolo.

Se si presentasse una situazione di campo libero e privo di ostacoli, il modulo provvede a fornire comunque un impulso di lunghezza fissa attorno ai 30ms.

Se la lunghezza dell'impulso è misurata in μs , dividendo per 58 tale valore si ottiene la distanza in centimetri, oppure dividendo per 148 si ottiene la distanza in pollici.

$$\mu\text{S}/58=\text{cm} \quad \text{oppure} \quad \mu\text{S}/148=\text{inches}$$

L' SRF05 può essere triggerato anche a velocità superiori, fino a 20 volte al secondo, ma è necessario aspettare 50 ms tra un impulso e il successivo perché esiste la possibilità che il modulo riceva parte dell'onda acustica inviata precedentemente e non ancora ritornata alla capsula.

Gli altri 5 pin presenti

I pin denominati "programming pins" sono utilizzati solo durante la costruzione per programmare la memoria flash all'interno del PIC16F630 presente sul circuito stampato.

Questi pin vanno lasciati non connessi.

Forma e larghezza del fascio di ultrasuoni

La forma del fascio di ultrasuoni dell' SRF05 è direzionale e riesce a coprire un raggio di 60° circa, con la larghezza del fascio che è funzione della superficie del trasduttore.

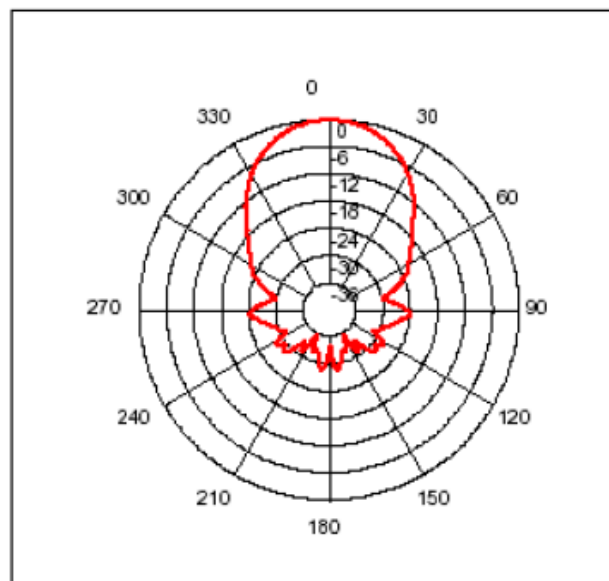
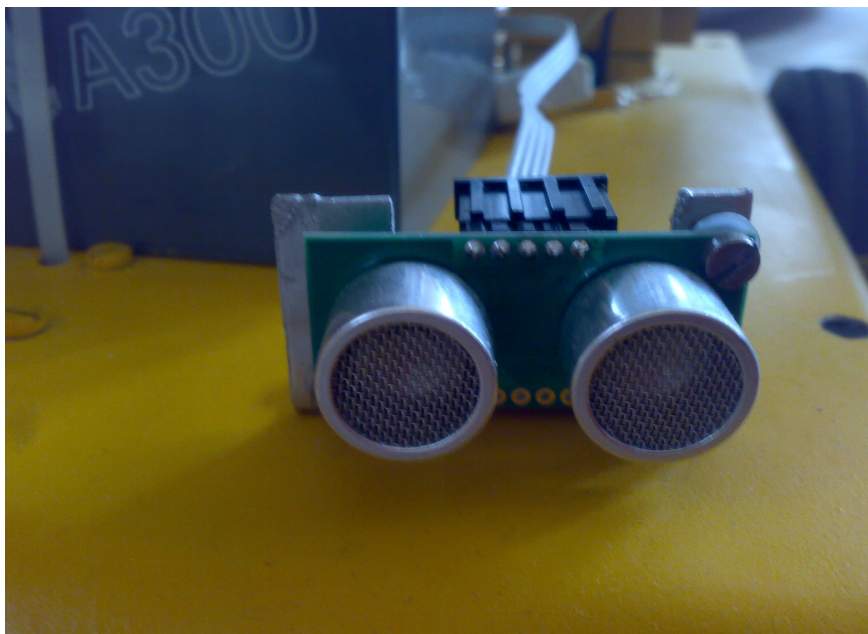


Foto: Particolare del sensore a ultrasuoni montato nella parte anteriore sinistra della scocca



Sensore Pioggia / TouchPad

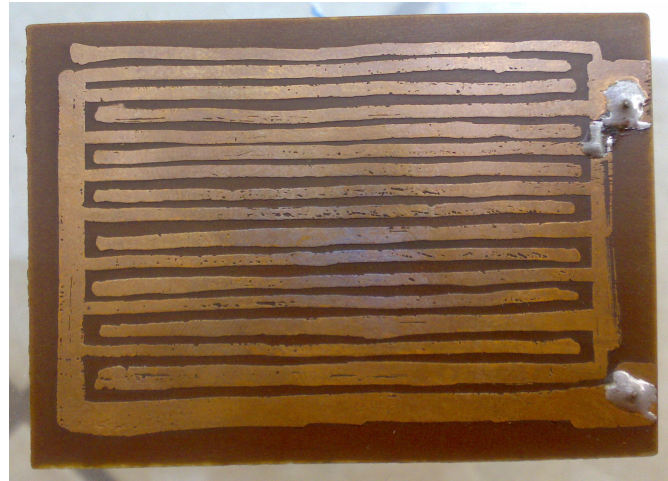
Questo sensore apre la sezione che descrive in modo approfondito i vari trasduttori predisposti da noi nel sistema.

Sensore Pioggia

Il sensore di pioggia serve per rilevare la presenza di precipitazioni atmosferiche di carattere piovoso o nevoso (anche se in inverno è sconsigliato utilizzare robot di questo genere, anche le versioni commerciali, all'esterno), in modo tale da interrompere le operazioni di taglio e annullare il movimento dei motori, ponendo poi il sistema in condizione di allarme segnalando all'operatore il raggiungimento dello stato ERRORE all'interno dell'automa tramite l'accensione del led rosso a luminosità fissa e BUZZER attivo e dal suono continuo.

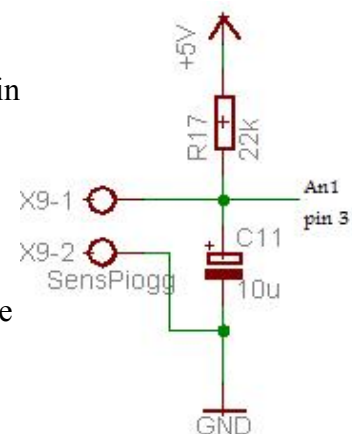
Il circuito di rilevamento precipitazioni si compone di due parti:

- il primo è un semplice circuito stampato (realizzato da noi manualmente tramite l'uso di un comune pennarello indelebile direttamente su di una piastra ramata a singola faccia scartata dal laboratorio perché non provvista di patina fotosensibile e quindi ormai in disuso) che presenta una caratteristica a “doppio pettine”:



quando la goccia d'acqua crea un corto circuito tra le due piste di rame, una collegata a massa e una collegata al circuito di condizionamento (figura inferiore), sul pin di uscita verso il PIC (lato destro del disegno figura inferiore) ci sarà un passaggio da un valore di 5V a un valore di circa 0V (in realtà il valore, a seconda della quantità d'acqua e della concentrazione di sali al suo interno, varia tra 0,5V e 1,7V: per questo abbiamo deciso di utilizzare un ingresso analogico al posto di uno digitale come si poteva ipotizzare con valori ben precisi di Vcc e GND, con soglia di commutazione tra livello logico alto e livello logico basso di 3,13V per ottenere un sicuro livello di presenza/assenza di acqua);

- il secondo è il semplice circuito di condizionamento proposto in figura: nel caso di assenza di acqua (circuito aperto) la resistenza carica il condensatore con il valore di Vcc (5V) che lo mantiene costante sull'ingresso del PIC. Viceversa nel caso di presenza di acqua (circuito chiuso) il condensatore è bypassato e il PIC riconoscerà il valore di tensione prossimo alla massa (tenendo sempre presente le considerazioni espresse prima).



Il valore digitale di questa soglia (RIF_PIOGGIA) è uguale a 0b10100000 (binario) o 0xA0 (hex).

TouchPad

Durante il collaudo di questo sensore, abbiamo notato che premendo con un dito sulla superficie a pettine della piastra di rame si ottiene l'effetto equivalente al corto circuito creato dalle gocce d'acqua: questo interessante, quanto relativamente inaspettato, utilizzo è risultato perfetto nel momento dell'applicazione della nostra copertura in Plexiglas sul corpo del robot, per proteggerlo meglio da urti, pioggia ed agenti esterni: infatti ci siamo resi conto che il portello creato sul lato posteriore del tosaerba permetteva agilmente di accendere il sistema, ma una volta in funzione e quindi in movimento era difficile "inseguire" il robot, aprire il portello e spegnere la scheda senza rischiare di provocare danni e/o rischiare di venire colpiti dal tosaerba durante il suo moto casuale di aggiramento degli ostacoli: risulta invece molto più semplice avvicinarsi, porre un dito (leggermente umido, come è solitamente la nostra pelle) sulla piastra di rame e simulare la presenza della pioggia.

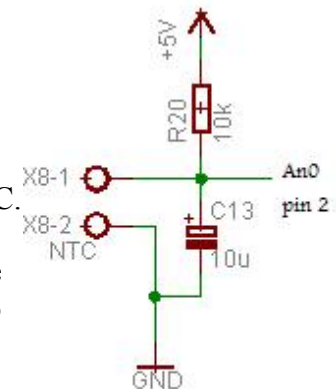
In presenza del segnale di allarme il sistema entra nello stato di errore come al solito e ferma motori e lama come normalmente previsto: è in questo momento che si può intervenire molto più comodamente all'apertura della finestra e spegnere il robot.

Sensore Temperatura

Questo sensore è formato da una semplice resistenza NTC del valore di 10 K Ω collegata ai morsetti del circuito presentato qui a fianco.

I termo resistori o termistori sono resistori che presentano elevati valori del coefficiente di temperatura; fra di essi si distinguono i PTC e gli NTC.

I resistori PTC (Positive Temperature Coefficient) hanno un coefficiente di temperatura positivo ossia aumentano la loro resistenza con l'aumento della temperatura mentre quelli detti NTC (Negative Temperature Coefficient) presentano un coefficiente di temperatura negativo (tra -6% e -2% per grado centigrado) ossia riducono la loro resistenza con l'aumentare della temperatura.



Questi termo-resistori possono essere impiegati per la misura diretta della temperatura e quindi abbiamo deciso di dotare il robot di uno di questi trasduttori collegato ai capi di un cavetto elettrico in modo da poterlo sistemare in più posizioni diverse per tenere sotto controllo varie parti del sistema, specialmente il surriscaldamento dei motori.



Il sistema è dimensionato in modo da avere un partitore di tensione che divide esattamente a metà la tensione Vcc alla temperatura ambiente di 20/25°C.

La soglia all'interno del PIC è fissata ad un valore di circa 2V.

Il condensatore in parallelo è inserito (come nel sensore di pioggia) per mantenere un valore continuo e non avere brusche variazioni.

Quando la temperatura sale, il valore di resistenza dell'NTC diminuisce, comportando una diminuzione della tensione sul pin di uscita verso il PIC: è stato adottato questo accorgimento in modo da fare funzionare il sistema anche in assenza della resistenza NTC; infatti se si considera il circuito privo della resistenza, il valore di tensione presente in uscita è prossimo a Vcc, e quindi il sistema trasformando il valore analogico in arrivo e comparandolo con la soglia lo equipara ad una temperatura bassa; viceversa quando la tensione scende al di sotto della soglia di 2V il sistema riconosce questa condizione come alta temperatura e manda il sistema nello stato di errore finché la resistenza non si raffredda e ritorna al di sopra della soglia oppure l'NTC non viene scollegata.

Il valore digitale corrispondente alla soglia (RIF_TEMP) di 2V è 0b11100000 (binario) oppure 0xE0 (hex).

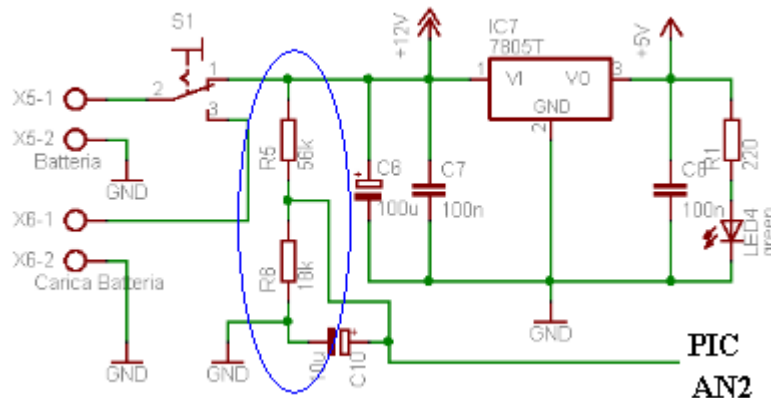
Rilevatore Stato della Batteria

Questo circuito è molto semplice ma permette di ottenere un trasduttore che fornisce al PIC un valore compatibile TTL di livello della carica (tensione) della Batteria.

Esso è composto da un semplice partitore di tensione applicato a monte del circuito regolatore che crea i 5V, in modo da prelevare la tensione direttamente dalla batteria.

Considerando che la batteria al piombo ha un valore massimo, quando completamente carica, che può aggirarsi anche attorno ai 13,7/14V e un valore minimo imposto da noi per motivi di sicurezza a 11,5V e dovendo fornire un valore TTL compatibile al PIC, si ottiene la seguente formula per ricavare la tensione di batteria:

$$V_o = V_{BATT} * (18k)/(56k+18k)$$



Nel caso di tensione di batteria entro valori normali di funzionamento ($14V > V_o > 11,7V$) la tensione in arrivo al PIC sarà $3,4V > V_o > 2,84V$, mentre quando è uguale a 11,5V o scende al di sotto di tale valore si avrà in ingresso al PIC un valore di tensione $< 2,79V$.

Il valore di 2,79V è riportato come soglia digitale (RIF_BATT) all'interno del PIC con il valore di 0b10000010 (binario) oppure 0x82 (hex).

Il sistema esegue quindi una comparazione tra questo valore e il valore in ingresso: se il valore misurato risulta minore o uguale della soglia il sistema entra nello stato di errore.

Varie ed Eventuali

Si tratta solo di una morsettiera con due canali di I/O disponibili per eventuali upgrade del sistema. Su uno di questi due pin è stato inserito infatti il BUZZER per creare un segnalatore acustico, assente in fase di progetto della scheda elettronica ma aggiunto in seguito.

Questa morsettiera dispone di un terzo pin con un punto di massa disponibile e utilizzabile.

BUZZER e LED segnalazione

Il sistema dispone di ben tre led di segnalazione, colorati diversamente, e di un segnalatore acustico.

Il led verde non è comandato direttamente ma è connesso lungo la linea di alimentazione, appena dopo il regolatore 7805, infatti indica l'accensione e la corretta alimentazione della scheda elettronica.

Il led giallo è pilotato direttamente dal PIC e nel software è implementato come monitor per il funzionamento dell'automa: cambia la propria luminosità (acceso/spento) ad ogni ciclo della macchina a stati e la mantiene per un ciclo (100ms), ottenendo un effetto di lampeggio continuo.

Il led rosso è pilotato direttamente dal PIC ma è totalmente disponibile e modificabile per segnalare gli errori quando il sistema entra in stato di allarme.

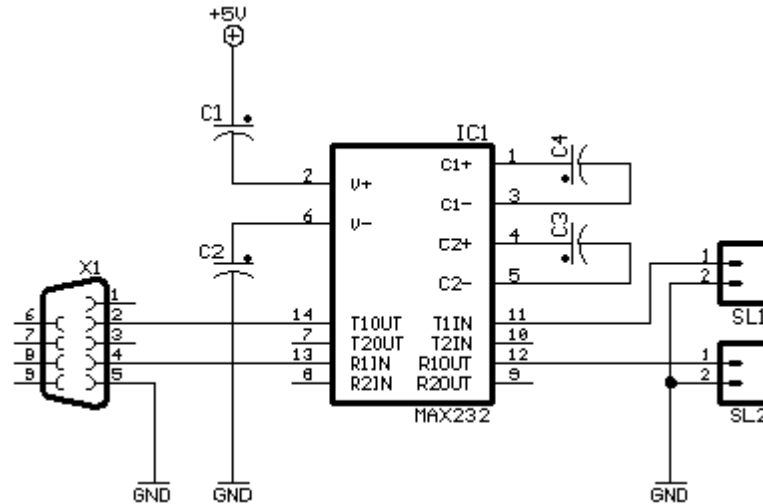
Attualmente, per ogni problema (pioggia, batteria o temperatura), il sistema accende il led rosso e lo mantiene acceso fino alla fuoriuscita dallo stato di errore.

Il buzzer invece funziona da segnalatore acustico: è comandato direttamente dal PIC (dopo esserci accertati che non assorbe più di 4mA) e svolge due principali funzioni:

- Quando il robot entra negli stati di rotazione (DX o SX) oppure compie una retromarcia, il segnalatore emette un impulso sonoro cadenzato ogni 100ms mantenendo il segnale alto per 100ms e basso per altrettanto tempo. Questa “beep beep” è stato aggiunto come caratteristica in più, ma non è indispensabile.
- Quando il robot arriva nello stato di Errore il Buzzer emette un suono continuo che non viene disabilitato finché la macchina a stati non esce dall'errore. Questo allarme è stato creato per essere udito in caso di malfunzionamento mentre il tosaerba è, ad esempio, in giardino e l'operatore si trova altrove: è un “arma” in più a disposizione del robot per attirare l'attenzione nel minor tempo possibile.

Ricezione e Trasmissione dati: Max232

Qui sotto è riportato lo schema elettrico di un interfaccia RS-232. Questo schema utilizza un integrato della MAXIM, il MAX232, per convertire i livelli logici della porta seriale in livelli TTL utilizzati dal PIC .



Come collegare una porta TTL o CMOS alla RS-232

In genere i segnali utilizzati dai sistemi digitali variano tra 0 e 5V e non sono quindi direttamente compatibili con lo standard RS-232. In commercio esistono appositi *traslatori di livello* che hanno il compito di fornire sia in trasmissione che in ricezione gli opportuni livelli pur non modificando la forma e il contenuto informativo del segnale trasmesso.

Il MAX232 (ed integrati simili, fatti dalla maggior parte dei produttori di semiconduttori) è un circuito integrato che permette il collegamento tra logica TTL o CMOS a 5V e le tensioni RS-232, partendo solo da un'alimentazione a 5V.

Per ottenere la tensione positiva e negativa necessarie per il funzionamento dell'integrato è usata una configurazione a *pompa di carica*, costituita da circuiti interni all'integrato e quattro condensatori esterni di circa 1 uF. La capacità effettiva dipende dal tipo di integrato e dalla relativa frequenza di commutazione; a volte i condensatori sono presenti all'interno dell'integrato stesso.

Sono disponibili anche integrati che richiedono un'alimentazione di solo 3,3V (per esempio il MAX3232).

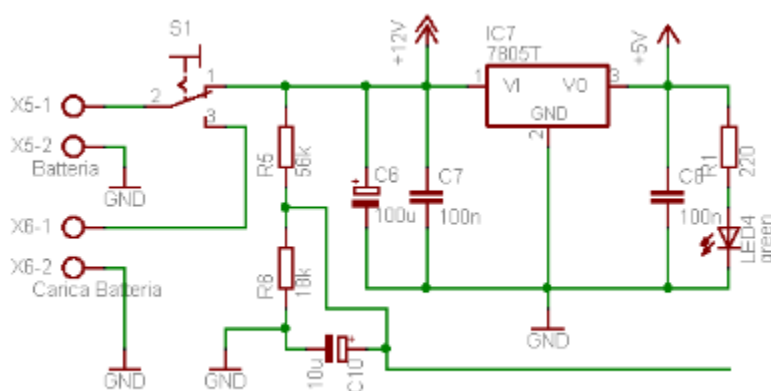
La sezione ricevente del MAX232 è costituita da due porte invertenti che accettano in ingresso una tensione di + o - 12V(o altra tensione compatibile allo standard RS-232) ed in uscita hanno un segnale TTL compatibile.

La sezione trasmittente ha due driver invertenti con un ingresso TTL compatibile, capaci di erogare a vuoto una tensione di poco inferiore a + o - 10V, compatibile con lo standard RS232.

Alimentazione e Consumi

L'alimentazione della scheda elettronica è stata creata modificando la tensione di batteria a 12V tramite un circuito che presenta un dispositivo regolatore a tre terminali denominato 7805. Esso crea una caduta di tensione su sé stesso (*Drop Out*) in modo da avere in uscita un valore fisso di 5V.

Con il drop out creato tra l'alimentazione da 12V e i 5V del regolatore, pari a 7V, e la corrente assorbita dalla scheda elettronica, attorno a 0,5A la potenza dissipata dal 7805 non è elevata, uguale a 3,5W. Con questo valore non è necessario disporre un dissipatore perché il calore che si forma si disperde già sufficientemente nell'aria. A conferma dei nostri calcoli abbiamo testato manualmente la temperatura e abbiamo valutato che non supera mai la temperatura ambiente. Di seguito è riportato il circuito di alimentazione presente a bordo del robot per la parte logica:



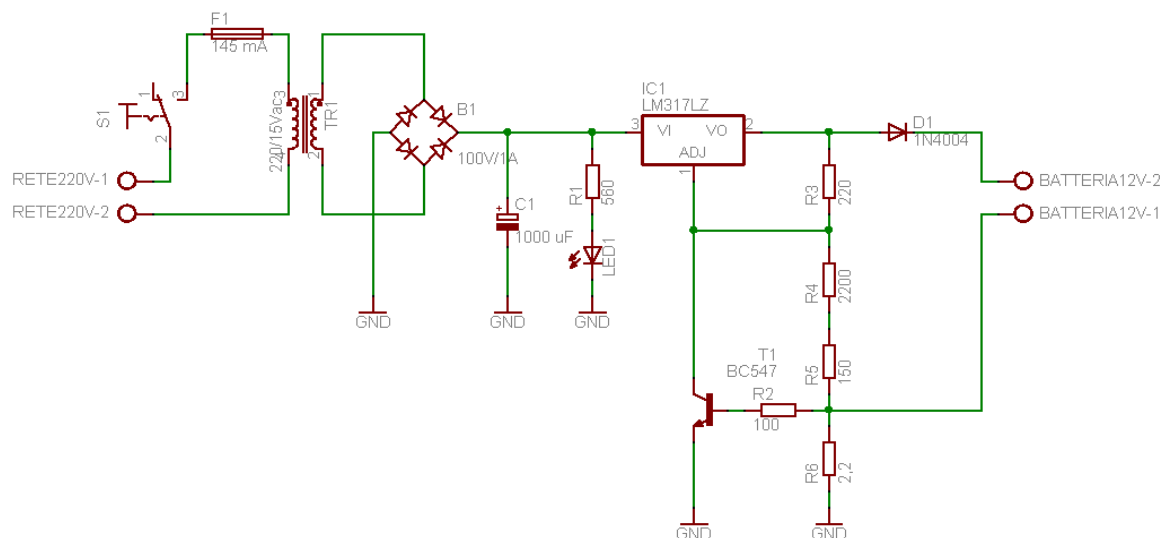
Abbiamo anche controllato e quantificato i consumi sia singoli che generali in modo da creare una previsione sulla durata della batteria e sull'autonomia disponibile al robot, il tutto riportato nella seguente tabella:

MOTORE	TENSIONE NOMINALE (V)	CORRENTE A VUOTO (A)	CORRENTE DI LAVORO (A)	POTENZA DI LAVORO (W)	AUTONOMIA PREVISTA (h)
DX	12	1,12	1,45	17,4	----
SX	12	0,98	1,3	15,6	----
LAMA	12	0,45	0,5	6	----
CIRCUITO	5	0,1	0,1	0,5	----
TOTALE	--	3,05	3,75	41,5	3 ~ 3,5

Carica Batteria

Per aumentare la sicurezza del sistema in fase di ricarica della batteria abbiamo realizzato un sistema di ricarica collegato alla tensione alternata di rete esterno. Questo per evitare che in caso di rottura del cordone di rete, esso venga in contatto con la scocca o con altre parti metalliche che metterebbero in pericolo la sicurezza degli utilizzatori.

Per questo motivo abbiamo realizzato un Carica Batterie esterno ad hoc per il nostro sistema:



Elenco Componenti:

Nome	Valore	Componente	Package	Descrizione
B1	100V/1A	RB1A	RB1A	PONTE A DIODI
BATTERIA12V		AK300/2	AK300/2	MORSETTIERA
C1	1000 uF	CPOL-EUE5-13	E5-13	CONDENSATORE ELETTROLITICO
D1	1N4004	1N4004	DO41-10	DIODO
F1	145 mA	GSH15	GSH15	PORTA FUSIBILE
IC1	LM317LZ	LM317LZ	TO92	REGOLATORE DI TENSIONE
LED1		LED3MM		LED3MM
R1	560 Ω	R-EU_0207/10	0207/10	RESISTENZA 1/4W
R2	100 Ω	R-EU_0207/10	0207/10	RESISTENZA 1/4W
R3	220 Ω	R-EU_0207/10	0207/10	RESISTENZA 1/4W
R4	2200 Ω	R-EU_0207/10	0207/10	RESISTENZA 1/4W
R5	150 Ω	R-EU_0207/10	0207/10	RESISTENZA 1/4W
R6	2,2 Ω	R-EU_0207/10	0207/10	RESISTENZA 1/4W
RETE220V	AK300/2	AK300/2		MORSETTIERA
S1		320-916	320-916	SWITCH
T1	BC547	BC547	TO92	NPN TRANSISTOR
TR1	220/15Vac	TRANSFORMER	U15	TRASFORMATORE

Principio di Funzionamento:

Il carica batteria presentato può essere utilizzato per qualsiasi batteria a secco da 12V.

Lo schema elettrico è un po' diverso dai comuni caricabatterie, perché differente è il sistema di ricarica previsto.

La tensione alternata di 15 volt 0,4 ampere circa (ma si può utilizzare uno anche superiore come valore di corrente), fornita dal secondario del trasformatore TR1, dopo essere stata raddrizzata dal ponte B1 viene applicata sul terminale di entrata dell'integrato IC1, un normale integrato stabilizzatore LM317.

Come si può notare, il terminale ADJ di questo integrato non risulta collegato a massa, bensì al collettore del T1, la cui base viene polarizzata dalla tensione presente sulla resistenza R6 collegata, tramite le resistenze R3 – R4 – R5, al terminale di uscita Vo.

Quando sui terminali +/- di uscita non risulta collegata nessuna batteria da ricaricare, ai suoi capi si ritrova la massima tensione che si aggira, considerando la caduta di tensione del diodo D1, sui 14V circa. Quando si collega una batteria scarica la tensione positiva presente ai capi del diodo D1 raggiunge la batteria e poiché il suo terminale negativo va a massa tramite la resistenza R6, ai suoi capi si ha una tensione di circa 0,6V, che polarizzando la base del transistor T1, lo porta in conduzione. In questo modo si riduce il valore ohmico delle due resistenze R4 ed R5 poste tra la base ed il collettore di T1. In pratica oltre il valore totale delle resistenze R4+R5 da 2350 Ohm bisogna considerare che collegato in parallelo a queste resistenze, c'è il valore ohmico presente tra il collettore e l'emettitore di T1. Ammettendo quindi che si abbia un valore totale di 2070 Ohm, per conoscere la tensione che si ottiene in uscita usiamo la formula:

$$V = [(R4+ R5)/R3+1]*1,25$$

Inserendo i valori del circuito si ottiene: $[(2070)/220+1]*1,25 = 13V$

In queste condizioni la batteria scarica assorbe la giusta corrente per la sua ricarica e non una corrente maggiore che potrebbe danneggiarla. Infatti se la corrente di carica aumentasse oltre il valore richiesto, il transistor condurrebbe maggiormente abbassando così il valore ohmico delle due resistenze R4 ed R5. Ammesso che il transistor T1 conduca tanto da portare il valore della R4+R5 a soli 1800 Ohm in uscita si avrà una tensione di soli: $[(1800)/220+1]*1,25 = 11,5V$

Poiché la tensione fornita dal caricabatterie è inferiore a quella della batteria, quest'ultima non assorbirà corrente. Una caratteristica molto interessante di questo alimentatore consiste nella capacità di erogare una corrente che rimane costante fino a quando la batteria non risulta completamente carica. A carica completata la tensione di 0,6V presente ai capi della resistenza R6 scende a 0V, e da questo istante la batteria non assorbe più nessuna corrente anche se l'uscita dell'alimentatore si ritrova una tensione di 14V. **Questa particolarità permette di inserire la batteria da ricaricare alla sera e di estrarla al mattino seguente già totalmente carica senza correre il rischio che possa danneggiarsi.** Il diodo D1 posto in serie sull'uscita serve come protezione per evitare, nel caso venisse a mancare la tensione di rete, che la batteria si scarichi sull'integrato LM317. **Facciamo presente che qualsiasi batteria a secco può essere lasciata inserita in questo caricabatterie anche per mesi interi senza che si danneggi, purché la batteria risulti da 12V.**

Vogliamo anche aggiungere, perché non è mai specificato nelle caratteristiche delle batterie a secco da 12V, che **si danneggiano molto velocemente se vengono scaricate fino a far scendere la loro tensione al di sotto dei 10V. Per tale motivo nel nostro programma abbiamo posto un limite a 11,7V** come protezione (spiegato nel capitolo relativo il software).

Foto: Fasi dell'assemblaggio

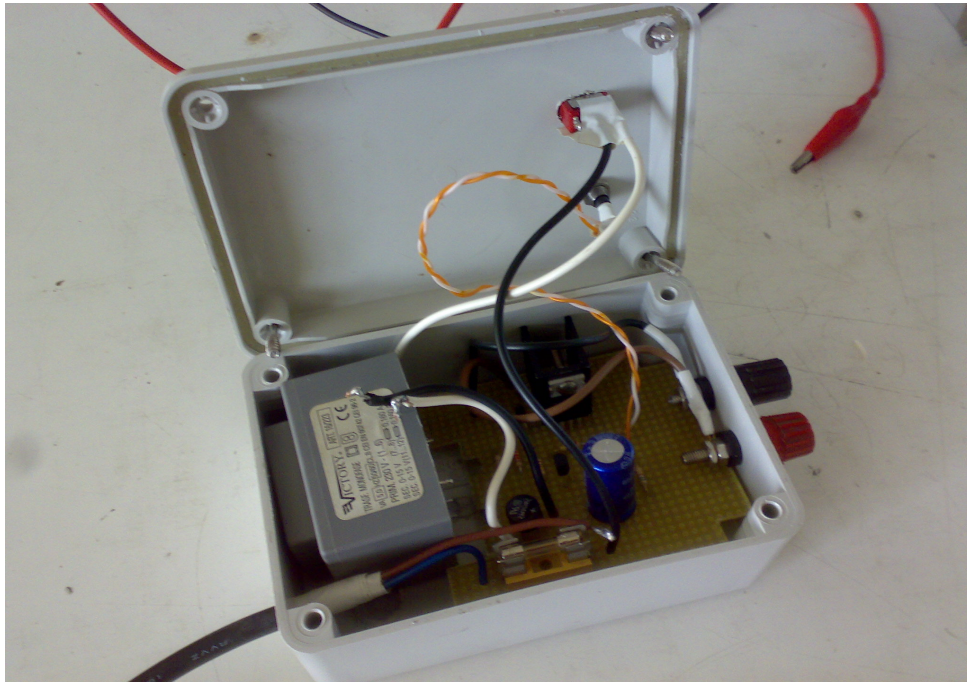
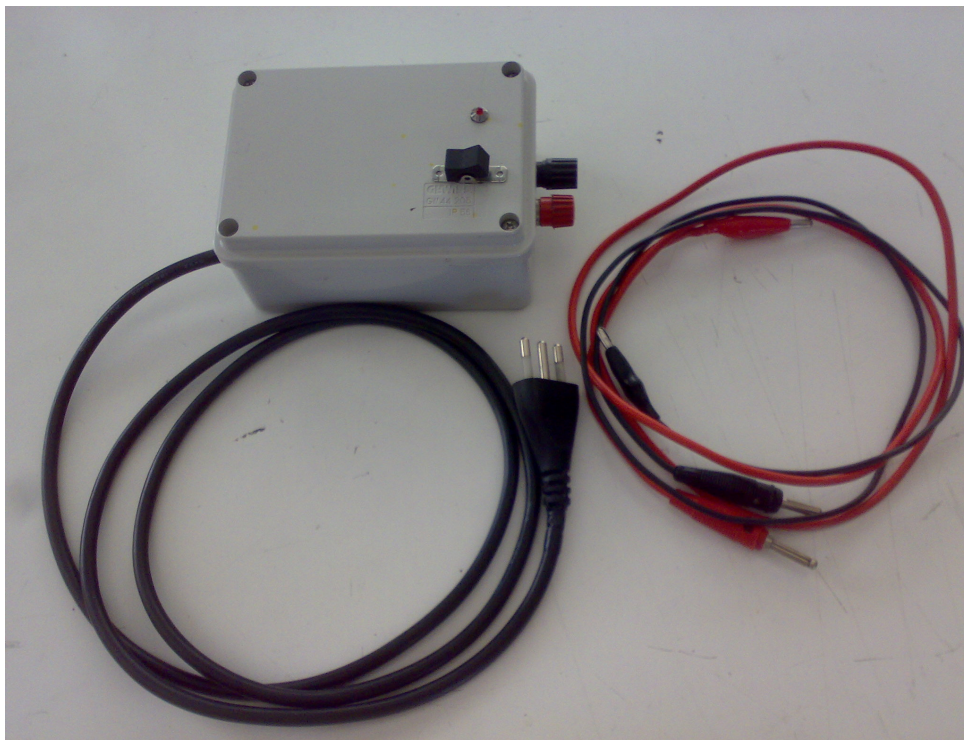


Foto: Il carica batteria ultimato



CONCLUSIONE

Il risultato del nostro lavoro ci ha soddisfatto pienamente in quanto per noi era la prima esperienza nel realizzare un sistema “a tutto tondo”, ovvero progettato dall'inizio alla fine e comprendente sia la parte hardware (meccanica, elettronica e fai-da-te) sia la parte software e di progettazione.

Abbiamo cercato di utilizzare tutte le conoscenze a nostra disposizione acquisite durante questi anni di specializzazione all'interno del Corso di Elettronica e Telecomunicazioni, infatti il nostro Tosaerba presenta caratteristiche di ogni materia d'indirizzo: Elettronica, Sistemi, Tecnologia, Disegno e Progettazione, Telecomunicazioni, Elettrotecnica e anche Meccanica.

La quantità di ore scolastiche di Laboratorio dedicate a questo progetto sono state molte, e il numero di problemi da risolvere che si presentavano di volta in volta numerosissimi. Il robot automatico ha presentato molti inconvenienti da gestire specialmente durante la fase di programmazione, uniti al fatto di utilizzare da poco un linguaggio di programmazione che, seppure semplificato, era per noi nuovo.

Tuttavia il risultato ottenuto è, a nostro giudizio, ottimo nel suo complesso, considerata la bassa quantità di aiuti esterni provenienti da Professori, altri Alunni o Tecnici di Laboratorio, non tanto per la loro disponibilità, quanto perché volevamo che questo progetto fosse il più possibile realizzato da noi in prima persona e rispecchiasse le nostre effettive capacità e soprattutto ci desse modo di creare a nostro piacimento un robot, sistema che integra perfettamente ogni campo della scienza delle automazioni vista la varietà di dispositivi di cui necessita per funzionare correttamente.

COSTI DI REALIZZAZIONE

Per realizzare questo progetto abbiamo acquistato alcuni elementi, in negozi specializzati o su internet, che non avremmo potuto costruire oppure reperire a scuola:

- I due sensori ad ultrasuoni SRF05 del costo di €25 cadauno.
- Il microcontrollore PIC16F876 (il magazzino non disponeva più di tale integrato) del costo di €13.
- I quattro RELE' degli azionamenti (il magazzino non disponeva di tali componenti) del costo di €2,50 cadauno.

Tutti i componenti montati sulla scheda elettronica sono stati forniti gratuitamente dal magazzino della scuola (costo stimato di circa €15).

Tutti gli altri componenti, motori, supporti, lamine metalliche e plastiche citati in questa relazione, batteria compresa, sono stati recuperati da materiale in disuso e riadattati al nostro progetto.

Questo rappresenta il punto di forza del nostro progetto: mantenendo come base la scheda elettronica, chiunque voglia riprodurre il nostro robot può ricostruirlo nel modo che più si addice al bisogno che questi ricerca.

Quindi, con una spesa minima di circa € 150 che comprende tutto il materiale necessario, si può realizzare una riproduzione più semplice e basilare ma al tempo stesso funzionale di un Tosaerba, Aspirapolvere o Lucida Pavimenti Automatici presenti in commercio spesso con cifre a partire da 4 ~ 5 volte il costo sopra citato fino a raggiungere modelli d'élite con costi proibitivi per la maggior parte dei cittadini di fascia media.

Ringraziamenti

Vogliamo ringraziare tutti coloro che ci hanno aiutato per portare a termine questo lavoro e far sì che funzionasse perfettamente, oppure che ci hanno offerto supporto per attività connesse come la lettura e la correzione della presente relazione:

- I Professori e le Professoresse del Corso F di Elettronica e Telecomunicazioni:
 - Prof. Stagnoli, docente di ELETTRONICA e tutor di classe che ha supervisionato la parte finale della stesura del programma.
 - Prof. Grasso, docente di TDP e TLC che ha messo a disposizione numerose ore di laboratorio per realizzare il progetto.
 - Prof. Frassine, docente di Laboratorio di TDP e SISTEMI che ha supervisionato il progetto della scheda elettronica.
 - Prof. Pangrazio, docente di SISTEMI, per le spiegazioni teoriche sulla gestione di un generico sistema a microcontrollore e sul Linguaggio C.
 - Prof. Spatarella, docente di Laboratorio di ELETTRONICA che ci ha aiutati a risolvere problemi con collegamenti elettrici e reperimento di componenti elettronici di recupero.
 - Prof.ssa Rezzola, docente di INGLESE che si è resa disponibile per correggere le parti in lingua inglese presenti in questa relazione.
 - Prof.ssa Babaglioni, docente di DIRITTO ed ECONOMIA che ci ha concesso alcune ore di lezione per completare questa relazione.
 - Prof.ssa Assumma, docente di ITALIANO e STORIA che ci ha concesso alcune ore di lezione per completare questa relazione.
 - Prof. Ferrara, docente di Laboratorio di TLC che ci ha fornito alcuni suggerimenti per l'accoppiamento dei motori alle ruote.
 - Prof. Colosio, docente di RELIGIONE, che ci ha concesso alcune ore di lezione per completare questa relazione.
- I Professori delle altre sezioni di Elettronica e Telecomunicazioni:
 - Prof. Belloni, che ha supervisionato la parte finale della stesura del programma.
- Tutti i tecnici e assistenti di Laboratorio del Corso di Elettronica e Telecomunicazioni.
- Un ringraziamento particolare a Luca Merigo, della classe 4^D, che si è appassionato a questo progetto e ha supervisionato la parte finale della stesura del programma.
- Un ringraziamento particolare ai nostri genitori che ci hanno stimolato con idee e suggerimenti per portare a termine il lavoro.

Fonti

Di seguito sono riportati i titoli o i collegamenti ad internet per il materiale aggiuntivo utilizzato per completare al meglio questa relazione.

- Articolo “Costruiamo un Rasaerba Automatico” del periodico “Elettronica IN” del settembre 2008 che ha ispirato il nostro progetto.
- Cenni teorici presenti nei nostri libri di testo.
- Cenni teorici sul funzionamento dei motori in continua da Wikipedia.
- Datasheet e manuali di funzionamento dei vari componenti o programmi utilizzati (riportati nella sezione “Allegati/Appendici”).