

# Progetto di un sistema di controllo per un robot basato su CAN

Laurea in Ingegneria Informatica ed Elettronica

UNIVERSITÀ DEGLI STUDI DI PERUGIA  
FACOLTÀ DI INGEGNERIA

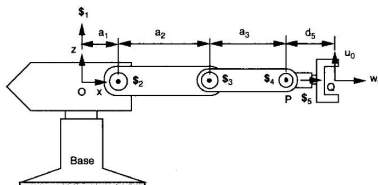
**Candidato:** *Giulio Spinozzi*  
**Relatore:** *Ing. Antonio Ficola*

19 Febbraio 2010

# Contenuti

- 1 Introduzione
- 2 Comunicazione
- 3 Controllo
- 4 Elettronica
- 5 Conclusioni

# Il problema

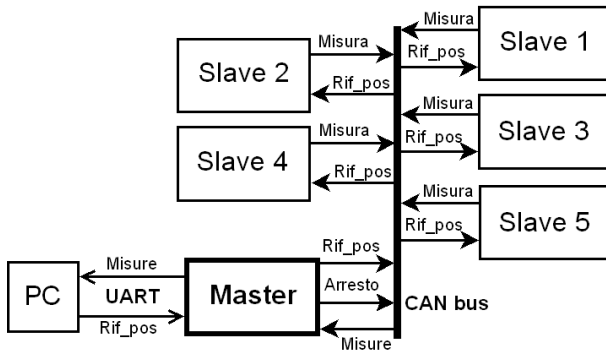


## Cosa si vuole realizzare?

Si vuole controllare i cinque motori dello SCORBOT tramite:

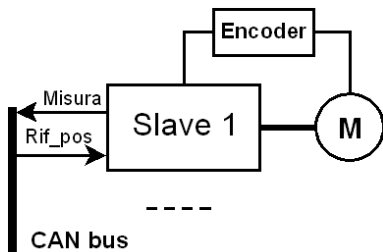
- 1 Una scheda elettronica per ogni motore (**Slave**)
- 2 Una scheda elettronica che scambia dati con un calcolatore (**Master**)

# Il sistema di controllo



- **Set-Point** (*Master -> Slave*), 2 byte
- **Arresto Motori** (*Master -> Slave*), 2 byte
- **Misure** (*Slave -> Master*), 2 byte

# Gli Slave



## Che compiti svolgono?

- 1 Riceve dal *Master* i segnali di **Set-Point** (*Rif\_pos*)
- 2 Controlla il motore (**M**) tramite *encoder incrementale* e modulazione *PWM*
- 3 Invia le **Misure** calcolate al *Master*

# Il Master

## Che compiti svolge?

- ① Scambia messaggi con un calcolatore tramite RS-232 (UART), il quale tramite software elabora i dati.
- ② Invia l'**Arresto dei motori** a tutti gli *Slave*
- ③ Invia il **Set-Point** allo specifico *Slave*
- ④ Riceve da ogni *Slave* le **Misure** calcolate dal ramo di controllo

Scambia dati con il PC con **Baud Rate** di 38.4 kHz

# Il dsPIC30F4011 della Microchip

## Dispositivo digitale programmabile a 16 bit

- Ogni scheda elettronica ne possiede uno
- Memoria Flash, molto rapida
- Molte periferiche, tra cui CAN, UART, PWM, QEI...
- Clock massimo di 30 MIPS
- Gestione degli **interrupt**
- Ambiente di sviluppo dedicato, *MPLAB IDE*
- Programmazione tramite linguaggio C30

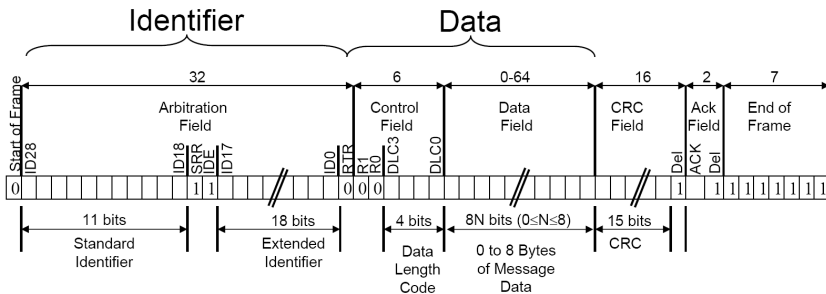
# Perchè il CAN per comunicare tra Master e Slave?

## Protocollo CAN (Controller Area Network) 2.0A

- Protocollo di comunicazione seriale
- Tutti i nodi sono connessi insieme al bus
- Ogni nodo può sia trasmettere che ricevere messaggi
- Velocità di trasferimento massima 1 Mbps
- Si trasmette un messaggio alla volta
- Priorità di messaggi
- Protocollo non orientato agli indirizzi
- Elevata affidabilità del protocollo

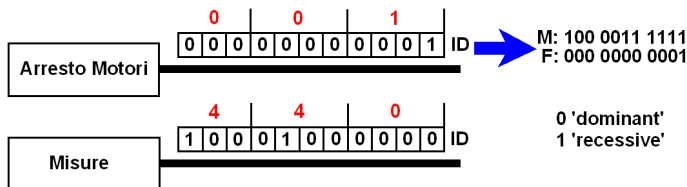


# Struttura dei messaggi



La rete è stata progettata per avere una **Baud Rate** di circa 625 kb/s, in cui il campo *Data* sia sempre di 2 byte. Ogni nodo poi seleziona tramite gli **ID** i messaggi di suo interesse con **maschere** e **filtri**.

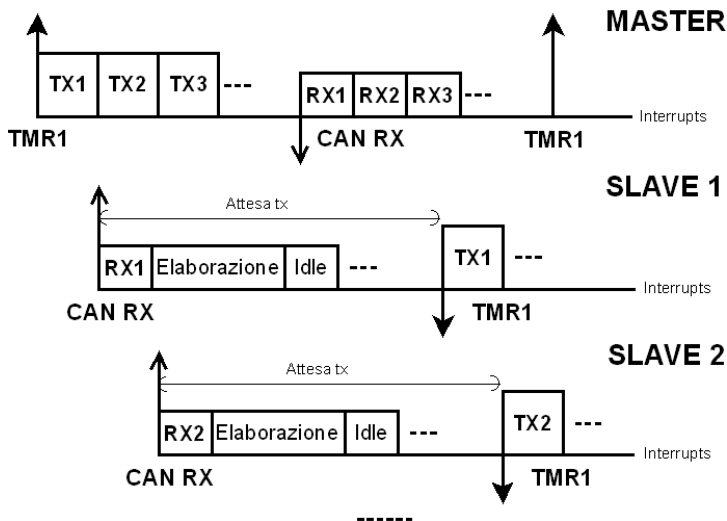
# Arbitraggio nel CAN bus



Poichè nel bus può viaggiare un messaggio alla volta...

- Si sceglie tra i due il messaggio a priorità maggiore, l'**Arresto Motori** (determinato dall'ID)
- Le maschere ed i filtri in ricezione degli *Slave* sono impostati per ricevere il messaggio con quell'ID

# Un problema di sincronia...

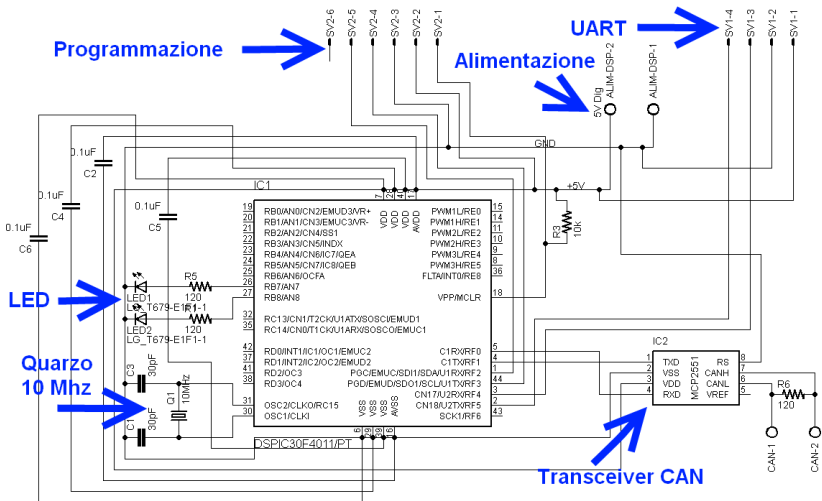


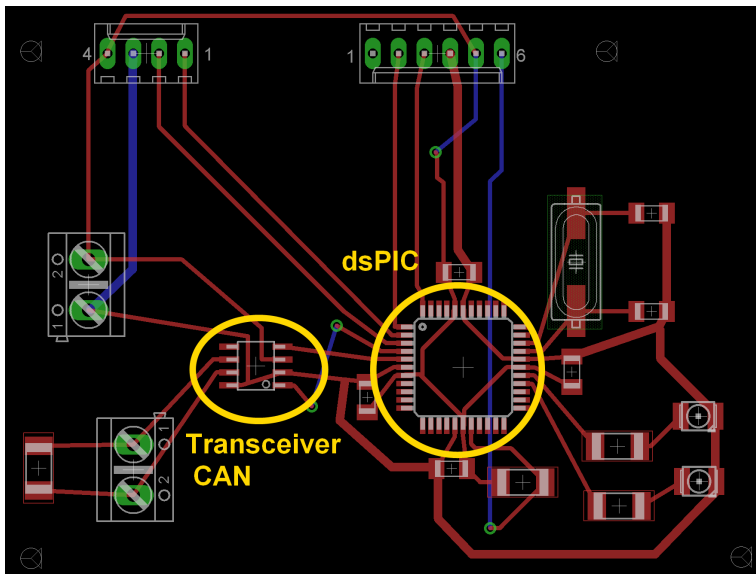
# Controllo nelle schede Slave

## Composto da:

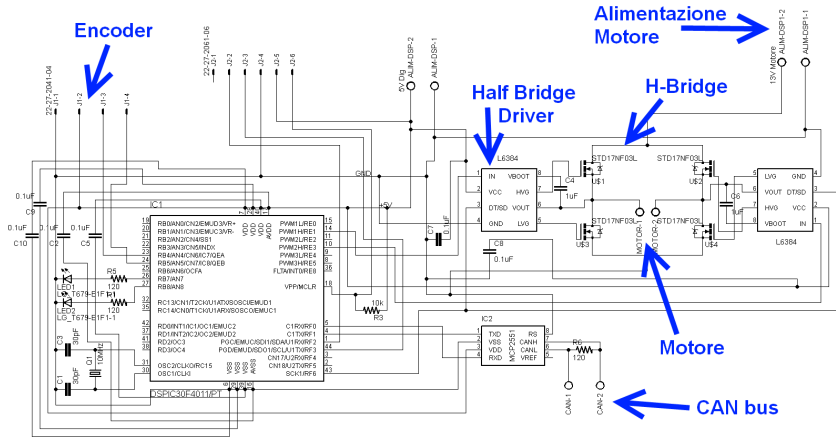
- Interfaccia e lettura della posizione del motore tramite **QEI**, encoder incrementale
- Modulo di potenza costituito da un **Ponte H** di quattro *MOS-FET* N-Channel, e due Half-Bridge driver L6384D
- Pilotaggio del motore tramite modulazione **PWM**
- Regolatore proporzionale nel firmware del *dsPIC*

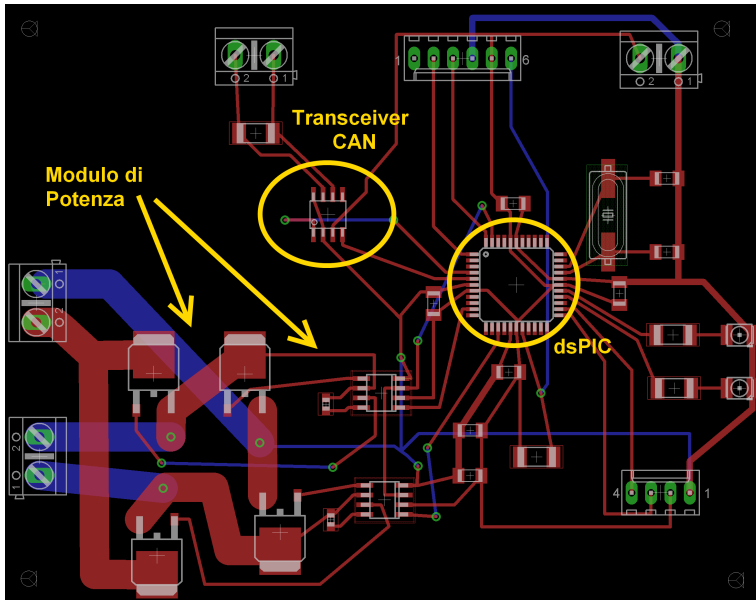
# La scheda Master





# La scheda Slave







# Concludendo...

Il lavoro di tesi si è articolato nelle seguenti fasi:

- ① Analisi e studio dei requisiti
- ② Realizzazione software (firmware)
- ③ Realizzazione hardware (elettronica)
- ④ Test, debug e collaudo

La realizzazione di questi prototipi porta a concludere:

- L'azionamento costruito è un sistema di controllo economico
- Abbastanza efficace

# Sviluppi futuri

## Da completare

- Sviluppare gli altri quattro *Slave*
- Misurare la corrente in entrata al *Ponte H* ed acquisire il dato nel *dsPIC*

## Da migliorare

- Gestione di maggiori messaggi tramite CAN
- Algoritmo di controllo più avanzato