



Robotic Rapid Prototyping

Manolo Garabini, Manuel Catalano, Lorenzo Malagia, Manuel Bonilla, Carlos Rosales, Fabio Bonomo, Alessandro Settimi e Antonio Bicchi



ROBOTICS RAPID PROTOTYPING

Software

- Firmware: software “lato” scheda
- Comunicazione HW - PC
 - Interfaccia fisica
 - Protocollo
- Software di interfaccia “lato” computer (librerie)
- Ambienti di sviluppo di alto livello
 - Simulink\Matlab
 - LabView
 - Software proprietari

Software

```
1 //===== includes
2
3 #include "qbmoveAPI/qbmove_communications.h"
4
5 //===== your code
6
7 int main() {
8     //Open port
9     openRS485(&comm_settings, port);
10
11     //Retrieve measurements
12     commGetMeasurements(&comm_settings, device_id, measurements);
13
14     //Retrieve currents
15     commGetCurrents(&comm_settings, device_id, currents);
16
17     //Set new position
18     commSetInputs(&comm_settings, device_id, inputs);
19
20     return 1;
21 }
22
```

In electronic systems and computing, firmware is the combination of persistent memory and program code and data stored in it. [1]

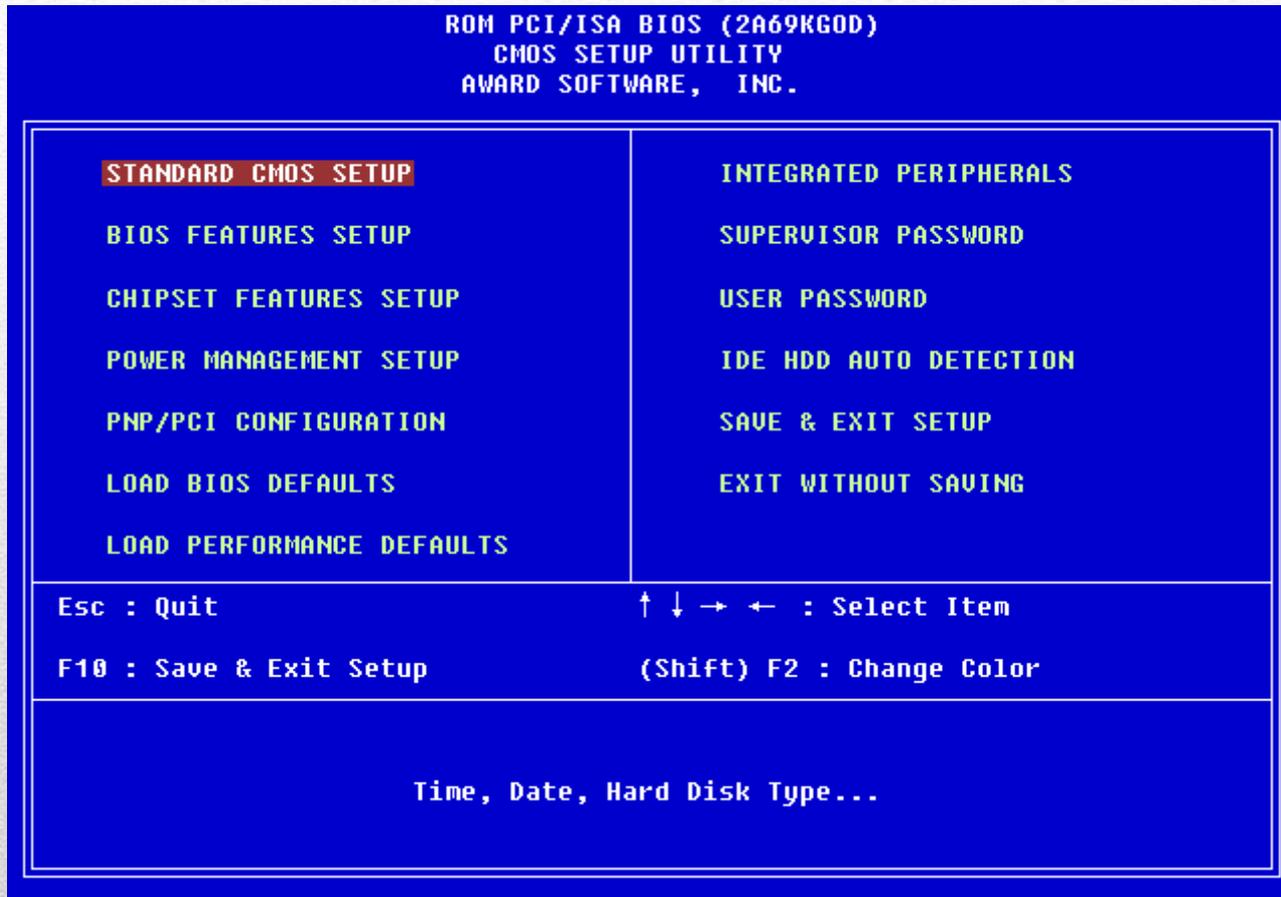
[1] The Authoritative Dictionary of IEEE Standards Terms. p. 438.
doi:10.1109/IEEESTD.2000.322230. ISBN 0738126012.

un programma (sequenza di istruzioni) integrato in un componente elettronico (integrati, schede elettroniche, periferiche). Il suo scopo è quello di avviare il componente stesso e consentirgli di interagire con altri componenti hardware tramite **protocolli di comunicazione** e/o interfacce di programmazione.

Deriva da "**firm**" (stabile) e "**ware**" (componente), indica che il programma non è immediatamente modificabile dall'utente finale (risiede in una memoria non volatile), e che si tratta del punto di incontro fra componenti logiche e fisiche, ossia fra hardware e software.

Firmware – Che cosa è?

- Il BIOS nei PC



Firmware – Un esempio

- Schede “**chiuse**” (Phidget):
firmware proprietario e
completamente
non modificabile
- Schede **Open-Source**
(Arduino): *firmware*
parzialmente o interamente
modificabile

Firmware

- Bootloader
- Gestione hardware e comunicazione (se necessaria) lato scheda
- ...

In qbControl

- Bootloader
- Implementazione Comunicazione MASTER-SLAVE
 - Ricevi pacchetti (e.g. inputs motori)
 - Invia pacchetti (e.g. misure)
- Lettura sensori
- Controllo motori
 - INPUT: Tensione al motore [V] (PWM diretto)
 - INPUT: Posizione di riferimento [$^{\circ}$] (PWM + PID)
 - INPUT: Coppia (Corrente) [Nm] ([A]) (PWM + PID)

Firmware - funzioni

- FPGA (Field Programmable Gate Array)

Contiene componenti di logica programmabile (Logic Blocks) e interconnessioni riconfigurabili che permettono di collegarli tra di loro, dalle porte AND OR NOT a reti logiche complesse

Firmware - FPGA

- Modifica fisicamente l'hardware (crea e distrugge fisicamente collegamenti elettrici all'interno di una matrice di elementi base ...)

Firmware – FPGA: Esempio

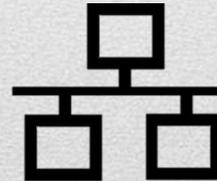
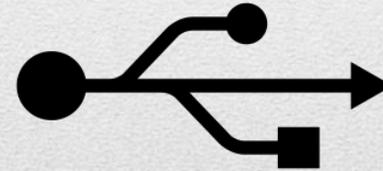
- Assembler
- C e affini
- Grafico (VPL Visual Programmin Language)

Firmware - linguaggi



COMUNICAZIONE

- UART - interfaccia seriale (con RS485 fino a 35 Mb/s)
- Wireless (fino a 600 Mb/s)
 - Infrarosso
 - ZigBee
 - Bluetooth
 - WiFi
- USB (fino a 3.2 Gb/s con USB 3.0)
- Ethernet (1-10 Gb/s)
- PCI-Express (1-31 GB/s)



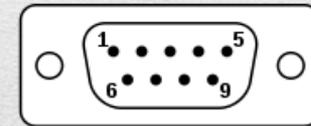
Comunicazione – Interfacce fisiche

UART - Universal Asynchronous Receiver/Transmitter

- Il mezzo fisico può essere costituito da 2, 3 o 5 fili
 - 2 fili: comunicazione half-duplex o simplex
 - 3 fili: comunicazione full-duplex
 - 5 fili: controllo di flusso hardware

Il controllo di flusso evita il congestionamento della rete

- La porta fisica:
 - nei portatili non più in uso
 - nei desktop in via di estinzione
 - nei PC industriali ancora utilizzata



Curiosità

Perchè nella porta sono presenti 9 pin?

Fu introdotta per essere usata con i modem...

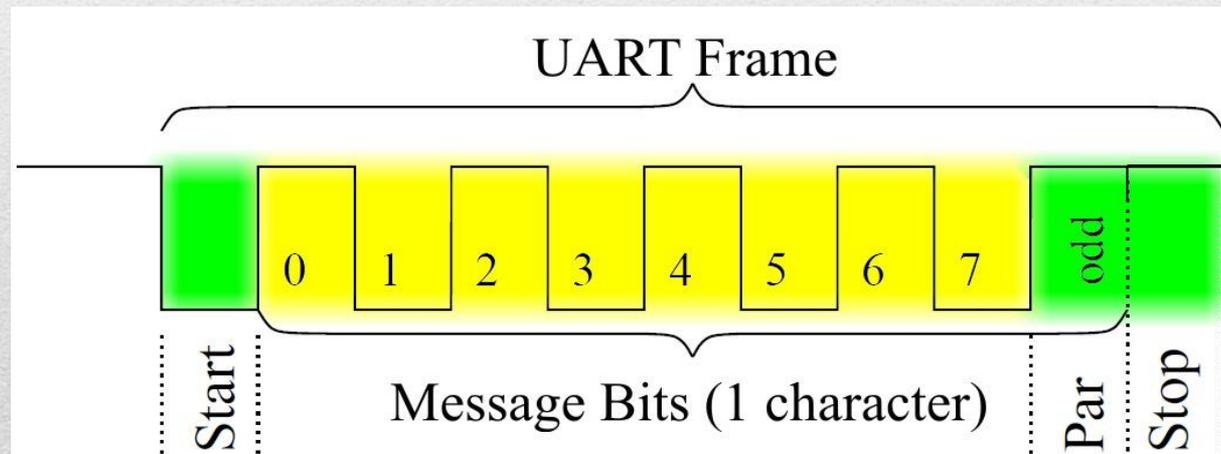
Riiiiing

Comunicazione – UART

Il protocollo definisce i livelli di tensione e la modalità di comunicazione.

Nel protocollo UART si trasmette byte per byte.

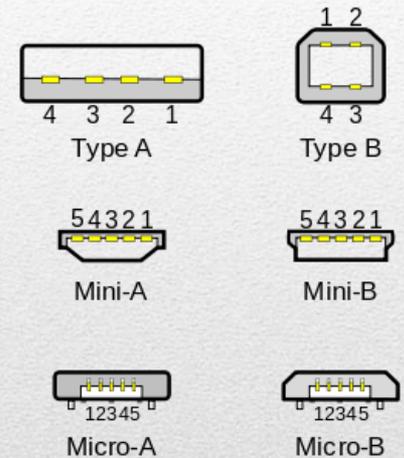
Ognuno è preceduto da un bit di start e seguito da almeno un bit di stop. Eventualmente c'è anche un bit di parità che funge da verifica.



Comunicazione – Protocollo UART

USB – Universal Serial Bus

- Sviluppata per standardizzare le connessioni delle periferiche al PC (si trova in tutti i dispositivi)
- Mezzo fisico costituito da 4 (o 5) fili
- E' in grado di alimentare dispositivi a 5 V con modeste richieste di potenza (500 mA da PC fino a 5 A). Non ha isolamento elettrico di sicurezza.
- Struttura pacchetto standardizzata



Comunicazione – USB

Il protocollo è molto più complesso e avanzato, ma essendo uno standard non è praticamente mai necessario scrivere le proprie funzioni per comunicare.

Molte volte fa solo da tramite ed è completamente trasparente all'utente finale che la vede in effetti come semplice porta seriale o altri tipi di comunicazione.



Comunicazione – Protocollo USB

Ethernet– Universal Serial Bus

- Sviluppata per reti locali
- Mezzo fisico costituito da 8 fili
- E' in grado di alimentare dispositivi a 48 V fino a 600 mA pari ad una potenza massima di circa 28 W.(Solo con Power-Over-Ethernet). Fornisce un isolamento elettrico di sicurezza fino a 1500 V a 50 Hz per 60 Secondi
- Struttura pacchetto definita



Comunicazione – Ethernet

Lo standard Ethernet definisce solo il livello fisico e il livello MAC (Media AccessControl) per la gestione di nodi multipli.

Al di sopra del pacchetto ethernet sono definiti altri standard come TCP/IP, UDP, ... che definiscono tutto il resto del modello OSI

Nel caso dell'ethernet le prestazioni della rete sono estremamente dipendenti dalla scelta del tipo di protocollo che deve essere valutata in base all'applicazione

Comunicazione – Protocollo Ethernet

L'integrazione tra la scheda che si occupa della comunicazione e la scheda madre del PC è al massimo livello, quindi la velocità di comunicazione che si ottiene è molto più elevata rispetto ad altre interfacce.



<http://www.advantech.it/>



National Instruments

Di contro presenta questi svantaggi:

- Costo elevato
- Non è possibile utilizzarle con un portatile
- La porta di uscita dalla scheda di comunicazione è proprietaria
- “Non Plug and Play”

Comunicazione – PCI express

*Master/slave is a model of communication where one device or process has **unidirectional control** over one or more other devices. In some systems a master is elected from a group of eligible devices, with **the other devices acting in the role of slaves**.*

Comunicazione – MASTER/SLAVE



SOFTWARE LATO PC

Insieme di istruzioni che consentono ad un PC di interfacciarsi con il Firmware e di svolgere operazioni non sempre realizzabili lato Firmware come:

- Data logging
- Controlli avanzati
- Teleoperazione o Controllo remoto
- Post-processing dati
- ...

Rappresenta la componente software del sistema robotico che più comunemente l'utente modifica per ottenere l'applicazione voluta

SW lato PC

- GUI (software proprietario)
- Librerie (e.g. C, Java, Python, ...)
- Librerie per ambienti avanzati
 - Simulink
 - LabView
 - ...

SW lato PC - strumenti

- Phidgets (Libraries)
 - Librerie C, C++, VisulaStudio, e molte altre
 - Librerie Matlab/Simulink
 - Librerie LabView
 - ...
- qbControl
 - Librerie C
 - Librerie Simulink
 - Strumenti da Terminale

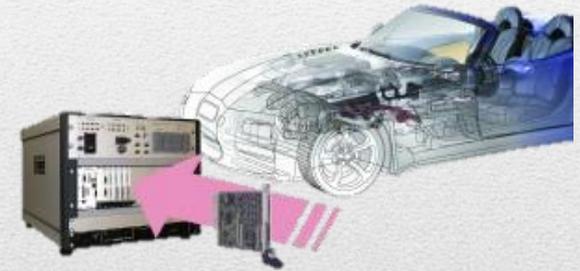
SW lato PC - esempi



ROBOTIC RAPID PROTOTYPING

Hw-in-the-Loop

- La simulazione Hardware-in-the-loop (HIL) è una tecnica utilizzata nello sviluppo e nel test di complessi sistemi embedded e real-time. La simulazione HIL permette di testare il software di controllo sull'hardware che costituirà il controllore, simulando l'impianto che deve essere controllato.
- Una simulazione HIL deve includere l'emulazione di sensori e attuatori i quali costituiscono la vera e propria interfaccia tra il sistema di controllo che si sta testando e la simulazione dell'impianto vero e proprio
- Durante il test il controllore embedded invia i segnali di controllo agli attuatori producendo cambiamenti nelle variabili di stato dell'impianto simulato



[T. Hwang, J. Rohl, K. Park, J. Hwang, K. H. Lee, K. Lee, S.-J. Lee, and Y.-J. Kim, "Development of HIL Systems for active Brake Control Systems", SICE-ICASE International Joint Conference, 2006.]

Hardware In the Loop

Ambiente grafico per la modellazione analisi e simulazione di sistemi dinamici

- Lineari – Non lineari
- Tempo continuo – Tempo discreto

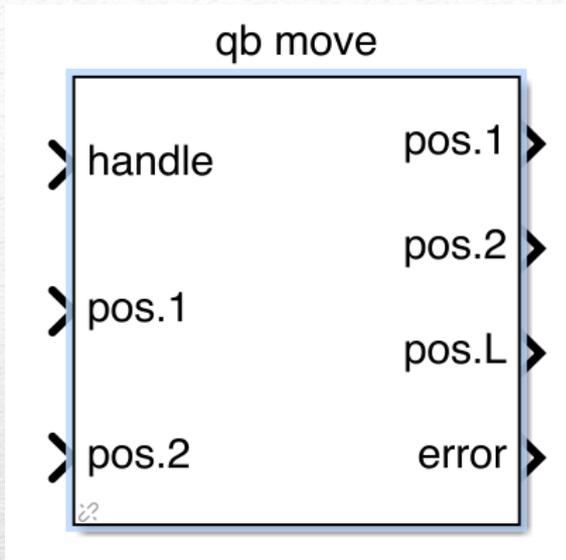
Integrato in MATLAB

Include un elevato numero di librerie con funzioni (più o meno complesse) già implementate

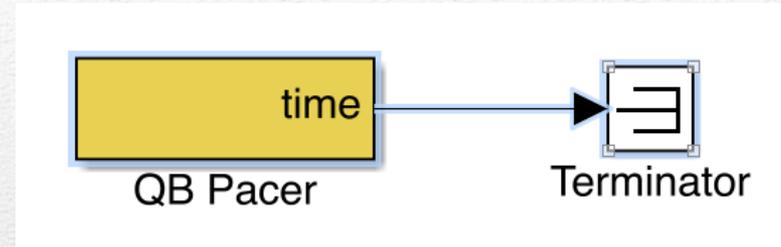
[[Simulink Getting Started](#)]



Simulink



Qb control Library
 Interfaccia Simulink della
 scheda di controllo



Qb Pacer

Fa si che il tempo di simulazione
 sia “sincronizzato” con il tempo
 reale

qbmove per Simulink

Simulation time

Start time: 0.0

Stop time: 10.0

Solver options

Type: Fixed-step

Solver: ode3 (Bogacki-Shampine)

Fixed-step size (fundamental sample time):

0.002

Tasking and sample time options

Periodic sample time constraint:

Unconstrained

Tasking mode for periodic sample times:

Auto

Automatically handle rate transition for data transfer

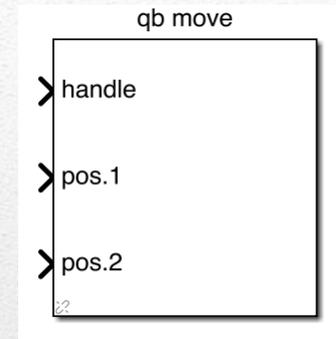
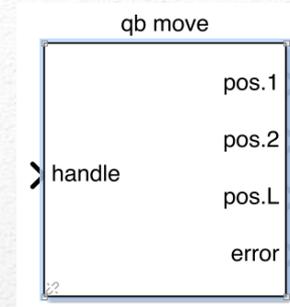
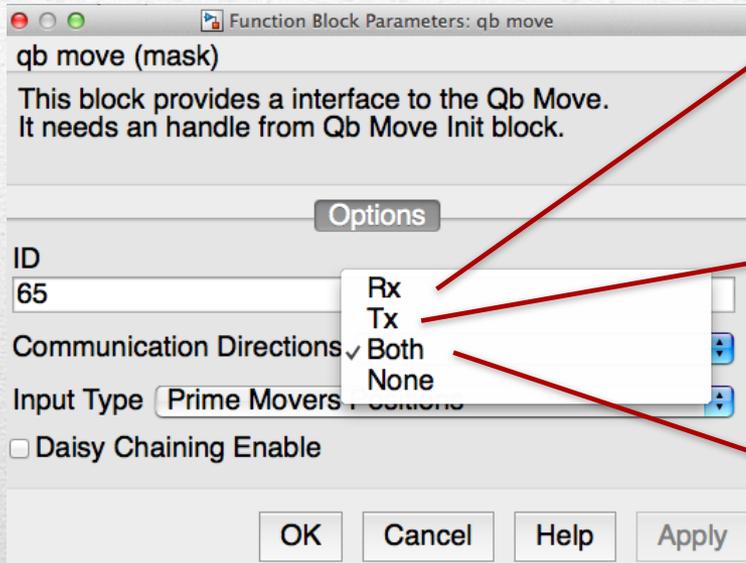
Higher priority value indicates higher task priority

- Integrazione a Passo Fisso
- Sample time non inferiore a 2 ms

Impostazioni preliminari

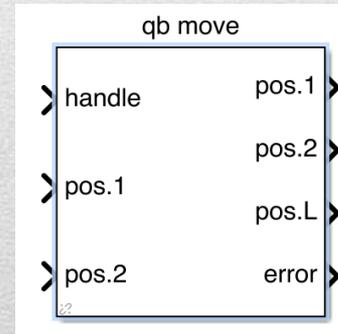
Modalità Rx

La scheda funziona come una scheda di acquisizione



Modalità Tx

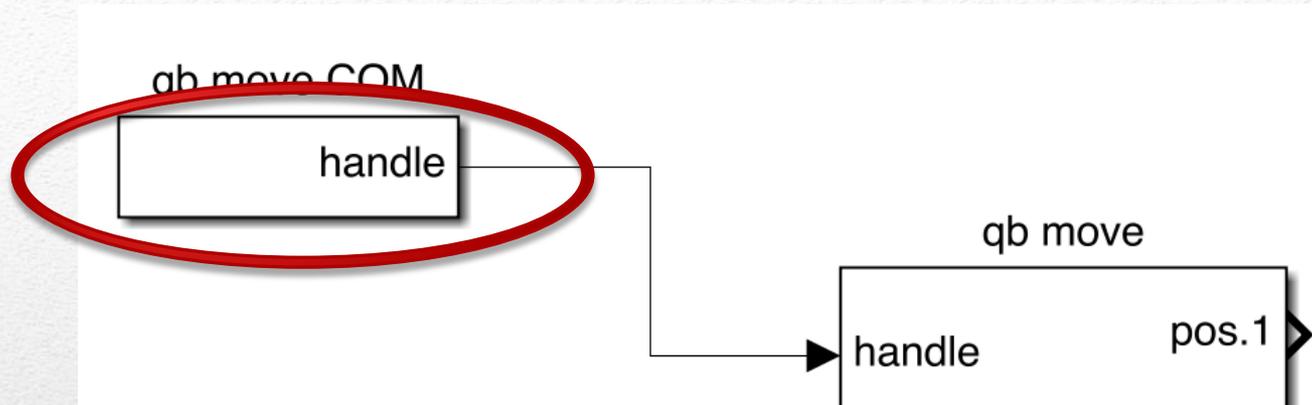
La scheda è vista come un servocomando



Modalità Both

La scheda gestisce gli ingressi e consente la lettura di sensori

qbmove library - configurazione



qbmove init block

consente la comunicazione tra il PC e la scheda di controllo.

Il settaggio dipende dal sistema operativo utilizzato e identifica la porta seriale attraverso cui il PC comunica con la scheda

- MAC OS X: l'interfaccia è di solito chiamata "tty.usbserial-XX" (dove XX è un numero che identifica l'ID della scheda)
- LINUX: l'interfaccia è di solito chiamata "tty.USB0"
- WINDOWS: l'interfaccia è di solito chiamata "COMXX" (dove XX è il più piccolo intero corrispondente ad una porta libera)

qbmove library Inputs - handle



DEMO TIME

Hw-in-the-Loop
