

Istituto **P**rofessionale per L'**i**ndustria L'**A**rtigianato

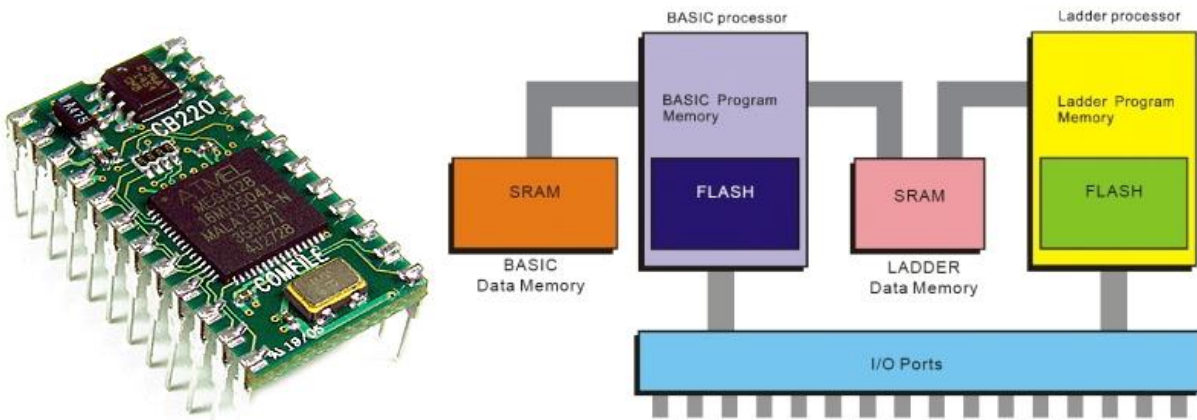
63039 SAN BENEDETTO DEL TRONTO (AP)

Classe 5A\_TIEN.

Anno Scolastico 2013-2014

## MICROCONTROLLORI CUBLOC

# Hardware



Vers. 1.2

Sistemi, automazione e organizzazione della produzione

## 1.1 Microcontrollore

Il **Microcontrollore** o **Microcontroller** o **MCU** è un dispositivo elettronico integrato su singolo chip, nato come evoluzione alternativa al Microprocessore.

**E' progettato per interagire direttamente con il mondo esterno tramite un programma residente nella propria memoria interna e mediante l'uso di *pin* specializzati o configurabili dal programmatore.**

Sono disponibili in 3 fasce di capacità elaborativa (ampiezza del *bus* dati): 8 bit, 16 bit e 32 bit. L'ampia gamma di funzioni di comando e controllo disponibili, sia analogiche che digitali, integrate sullo stesso chip, permette l'impiego delle MCU in sostituzione di schede elettroniche cablate tradizionali ben più complesse e costose.

Per i microcontrollori sono rilasciati sistemi di sviluppo amatoriali e professionali anche in modalità *open source*.

## 1.2 - Caratteristiche di un microcontrollore

L'architettura del Microcontrollore prevede un insieme di moduli fissi, comuni a tutti i modelli, e una serie di possibili estensioni in funzione del costruttore, del prezzo e della fascia applicativa):

- Unità di elaborazione (CPU o Core)
- Memoria di programma (ROM, OTP, EPROM, FLASH)
- Memoria dati (RAM e EEPROM)
- Porte di I/O configurabili
- Gestione Interrupt
- Oscillatore interno o esterno
- Moduli aggiuntivi
  - Contatori e temporizzatori
  - Moduli di comunicazione: (USART, I2C, SPI, USB, Ethernet, IrDA, CAN, Wi-Fi, Zigbee)
  - Interfacce analogiche o tecnologia mista: ADC, DAC, PWM, Comparatori analogici
  - Interfacce di visualizzazione e controllo: (LCD, Touch sensor)

Da quanto fin qui esposto risulta evidente quale sia la differenza tra un microprocessore e un microcontrollore: il primo contiene esclusivamente l'unità centrale di calcolo (CPU) e quindi per poter funzionare necessita di una memoria ROM esterna nella quale viene scritto il programma, di una RAM per i dati e di alcuni integrati per l'interfacciamento; nei microcontrollori tutto ciò è contenuto all'interno di un singolo chip.

Il principio di funzionamento di un microcontrollore è molto semplice, e coincide con quello di un computer o elaboratore elettronico e può essere riassunto in solo tre operazioni eseguite dalla CPU:

- legge l'istruzione contenuta nella memoria programma;
- la interpreta;
- la esegue.

### **Riepilogo**

Il microcontrollore è un dispositivo che raggruppa su un unico chip tutto il necessario per un sistema a microprocessore.

1. CPU (core)
2. Memoria RAM
3. Memoria EPROM, EEPROM, OTP
4. Porte I/O
5. Timers e contatori
6. Porte di comunicazione seriale speciali ("Bus CAN", "Bus I2C")
7. Convertitori A/D

### 1.3 Vantaggi del sistema a microcontrollore

Il successo e l'enorme crescita del Mercato di questi componenti sono dovuti a questi fattori:

1. basso costo (consente di sostituire 1 o più circuiti integrati tradizionali a costo inferiore);
2. ampia scalabilità di prestazioni, di complessità (da 8 pin a 144 pin) e velocità (da 1 MHz a 200 MHz);
3. vasta gamma di dotazioni in periferiche e moduli specializzati;
4. Ridotto il numero di componenti esterni, ovvero semplicità di realizzazione, minore complessità del circuito stampato, minor spazio occupato;
5. facilità di programmazione dovuta anche ai numerosi tool di sviluppo disponibili.
6. Ampia (e spesso libera) disponibilità di librerie, codici di esempio e documentazione
7. riconvertibilità del progetto (riprogrammando il dispositivo);
8. Grande flessibilità applicativa .
9. Brevi tempi di introduzione sul mercato del prodotto finito.
10. risparmio energetico (le versioni CMOS supportano il modo di funzionamento stand-by: è possibile bloccare, *via software*, l'attività della CPU e quindi ottenere correnti di alimentazione molto basse);
11. protezione contro le copiatore (la maggiore parte del single-chip offre la possibilità di proteggere da lettura il programma contenuto nella ROM);

### 1.4 Applicazioni

Le principali applicazioni sono:

- **automotive** (auto e altri mezzi di trasporto), che utilizza decine, in alcuni casi anche centinaia, di componenti per singola unità industriale venduta;
- **telefonia mobile e telecomunicazioni;**
- **prodotti medicali;**
- **sistemi acquisizione ed elaborazione dati** (centraline rilevamento dati ambientali);
- **consumer;**
- **automatismi e Demotica;**
- **sistemi di sicurezza;**
- **elettrodomestici.**

Spesso utilizziamo questi dispositivi senza rendercene conto, come per le smartcard delle carte di credito o per le cartoline musicali di auguri.

### 1.5 DSP

Parallelamente al Microcontrollore, e in continua evoluzione di potenza e di mercato, esistono i DSP (Digital Signal Processor) che incorporano moduli specializzati nel trattamento in digitale di segnali analogici. I campi tipici di utilizzo sono nel controllo di azionamenti (motori), di componenti per auto e avionica, di trattamento di segnali multimediali (codifica/decodifica audio e video, streaming e, campo principe, nella telefonia mobile. Il DSP ha tipicamente una struttura a 32 bit, e prossimamente a 64 bit.

### 1.6 Cenni storici

Il primo computer on-chip ottimizzato per applicazioni di controllo è stato il modello 8048 di Intel, rilasciato nel 1975, con RAM e ROM sullo stesso chip. Questo componente è stato utilizzato in più di un miliardo di tastiere per PC e numerose altre applicazioni.

Nei primi anni di sviluppo del Microcontrollore, la maggior parte dei modelli era commercializzata in due varianti. La più economica era dotata di memoria di programma programmata in fabbrica (ROM) su specifiche del cliente oppure programmabile dall'utente una sola volta (OTP, One Time Programming). La seconda, più costosa, aveva la memoria di programma cancellabile EPROM mediante esposizione a luce ultravioletta del chip tramite la finestrella trasparente che lo sovrastava.

Nel 1993, Microchip ha introdotto il modello di MCU PIC16C84, caratterizzato da memoria programma in EEPROM, ovvero cancellabile elettricamente, che permetteva sia lo sviluppo veloce del prototipo del prodotto finito, sia la modifica del Firmware a circuito montato (In-System Programming). La semplificazione del contenitore (package), senza finestrella in quarzo, ha contribuito a ridurre il costo finale del componente.

Nello stesso anno, Atmel ha rilasciato il primo MCU che utilizzava una memoria di tipo Flash, ancora più semplice e veloce da programmare/modificare, più compatta e con un ciclo di vita (cancellazioni) molto più elevato. Questo ha segnato l'inizio del massiccio utilizzo del Microcontrollore nelle più disparate applicazioni.

## 2.1 Cubloc – Controllore programmabile

Il CUBLOC, un PLC in miniatura, programmabile in Basic come un microcontrollore e in Ladder come un PLC.

Il PLC è un dispositivo dotato di molti ingressi e uscite. I segnali da esso accettati sono di tipo digitale (Vero o Falso) e di tipo analogico (tensioni variabili nel tempo). Tali valori sono processati dalla CPU interna che li elabora attraverso un programma. La sua dimensione è spesso notevole e per il suo funzionamento sono necessari numerosi cablaggi con le periferiche esterne. Si utilizza quasi sempre in rack (Contenitore metallico), dotato di numerosi slot e alloggiamenti di schede da cablare opportunamente.





Il CUBLOC è un Controllore Programmabile, prodotto dalla Comfile Technology, differente dal tradizionale PLC: quest'ultimo infatti è un dispositivo programmabile che necessita di numerosi cablaggi e collegamenti esterni per funzionare.

Il CUBLOC si può considerare come un piccolo PLC “On-Chip”, ma con tutte le sue funzionalità, per fornire al programmatore più libertà e flessibilità nelle applicazioni create. E' un modulo “ibrido” pensato anche per i principianti, dando loro modo di creare le proprie applicazioni in maniera del tutto semplice.

Allo stato attuale, la Comfile Technology offre diversi modelli (CB210, CB220, CB280, CB290, CB320, CB380, CB400 e CB405) utilizzabili per qualunque applicazione e dotati di caratteristiche molto interessanti. Nella tabella 1 e e nella tabella 2 sono riportate le caratteristiche di tutti i modelli attualmente disponibili.

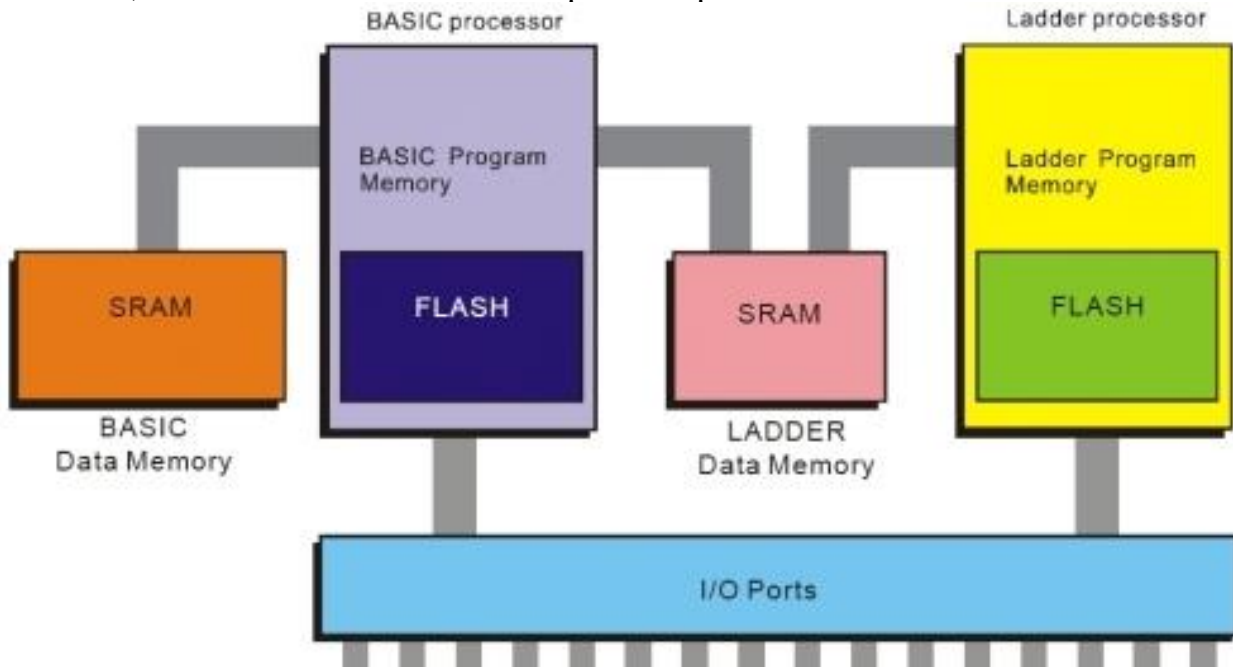
# CUBLOC Specifications

	CB210	CB220	CB280	CB290
<b>Picture</b>				
<b>Program Memory</b>	80KB	80KB	80KB	80KB
<b>Data Memory</b>	2KB(BASIC)+ 1KB(Ladder Logic)	2KB(BASIC)+ 1KB(Ladder Logic)	2KB(BASIC)+ 1KB(Ladder Logic)	24KB(BASIC)+ 4KB(Ladder Logic)
<b>EEPROM</b>	4KB EEPROM	4KB EEPROM	4KB EEPROM	4KB EEPROM
<b>Program Speed</b>	36,000 inst./sec	36,000 inst./sec	36,000 inst./sec	36,000 inst./sec
<b>General Purpose I/O</b>	20 I/O lines (5V TTL) (input/output configurable)	16 I/O lines (5V TTL) (input/output configurable) + Spare I/O 6 (5V TTL)	49 I/O lines (5V TTL) (input/output configurable)	91 I/O lines (5V TTL) (33 input only + 32 output only + 26 input/output configurable)
<b>Serial Ports</b>	1 serial port (Channel 1: TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps)	2 serial ports (Channel 0: RS232C 12V, Channel 1: TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps	2 serial ports (Channel 0: RS232C 12V, Channel 1: RS232C 12V & TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps	2 serial ports (Channel 0: RS232C 12V, Channel 1: RS232C 12V & TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps
<b>Analog Inputs</b>	6 Channel 10-bit ADCs	8 Channel 10-bit ADCs	8 Channel 10-bit ADCs	8 channel 10-bit ADCs
<b>Analog Outputs</b>	- 3 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz	- 3 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz	- 6 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz	- 6 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz
<b>External Interrupts</b>	None	4 channels (in spare I/O)	4 Channels	4 Channels
<b>High Speed Counters</b>	None	2 Channel 32-bit Counters (up to 2Mhz)	2 Channel 32-bit Counters (up to 2Mhz)	2 Channel 32-bit Counters (up to 2Mhz)
<b>Power</b>	DC 9V to 12V, 100mA or From USB	5 to 12V, 40mA (ports unloaded)	5V, 40mA (ports unloaded)	5V, 70mA (ports unloaded)
<b>RTC</b>	No	No	No	Yes
<b>Data Memory Backup</b>	None	None	None	Optional
<b>Operating Temp.</b>	-40 °C to 120 °C	-40 °C to 120 °C	-40 °C to 120 °C	-40 °C to 120 °C
<b>Package</b>	I/O Board	24-pin DIP 600mil	64-pin Module	108-pin Module
<b>Size</b>	2.9"L x 2.0"W x 0.4"H (75 x 53 x 12 mm)	1.2"L x 0.6"W x 0.4"H (30 x 15.3 x 11 mm)	1.4"L x 1"W x 0.4"H (35 x 25.4 x 11 mm)	2.4"L x 1.9"W x 0.5"H (59.4 x 47.8 x 13 mm)

	<b>CB320</b>	<b>CB380</b>	<b>CB405</b>	<b>CB400</b>
<b>Picture</b>				
<b>Program Memory</b>	200KB	200KB	200KB	200KB
<b>Data Memory</b>	6KB(BASIC)+ 1KB(Ladder Logic)	6KB(BASIC)+ 1KB(Ladder Logic)	51KB(BASIC)+4KB(Ladder Logic)+55KB(Heap)	6KB(BASIC)+ 1KB(Ladder Logic)
<b>EEPROM</b>	4KB EEPROM	4KB EEPROM	4KB EEPROM	4KB EEPROM
<b>Program Speed</b>	36,000 inst./sec	36,000 inst./sec	36,000 inst./sec	36,000 inst./sec
<b>General Purpose I/O</b>	16 I/O lines (5V TTL) (input/output configurable) + Spare I/O 6 (5V TTL)	49 I/O lines (5V TTL) (input/output configurable)	64 I/O lines (5V TTL) (input/output configurable)	83 I/O lines (5V TTL) (input/output configurable)
<b>Serial Ports</b>	2 serial ports (Channel 0: RS232C 12V, Channel 1: TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps	2 serial ports (Channel 0: RS232C 12V, Channel 1: RS232C 12V & TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps	4 serial ports (Channel 0: RS232C 12V, Channel 1 to 3: RS232C TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps	4 serial ports (Channel 0: RS232C 12V, Channel 1 to 3: RS232C TTL 5V) - Configurable Baud rates: 2400bps to 230,400 bps
<b>Analog Inputs</b>	8 Channel 10-bit ADCs	8 Channel 10-bit ADCs	16 channel 10-bit ADCs	16 Channel 10-bit ADCs
<b>Analog Outputs</b>	- 6 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz	- 6 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz	- 12 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz	- 12 Channel 16-bit PWMs (DACs) - Frequency: 35hz to 1.5Mhz
<b>External Interrupts</b>	4 Channels (in Spare I/O)	4 Channels	4 Channels	4 Channels
<b>High Speed Counters</b>	2 Channel 32-bit Counters (up to 2Mhz)	2 Channel 32-bit Counters (up to 2Mhz)	2 Channel 32-bit Counters (up to 2Mhz)	2 Channel 32-bit Counters (up to 2Mhz)
<b>Power</b>	5 to 12V, 40mA (ports unloaded)	5V, 40mA (ports unloaded)	5V, 50mA (ports unloaded)	5V, 40mA (ports unloaded)
<b>RTC</b>	No	No	No	No
<b>Data Memory Backup</b>	None	None	Optional	None
<b>Operating Temp.</b>	-40 °C to 120 °C	-40 °C to 120 °C	-40 °C to 120 °C	-40 °C to 120 °C
<b>Package</b>	24-pin DIP 600mil	64-pin Module	80-pin Module	108-pin Module
<b>Size</b>	1.2"L x 0.6"W x 0.4"H (30 x 15.3 x 11 mm)	1.4"L x 1"W x 0.4"H (35 x 25.4 x 11 mm)	2.4"L x 1.9"W x 0.5"H (59.4 x 47.8 x 13 mm)	2.4"L x 1.9"W x 0.5"H (59.4 x 47.8 x 13 mm)

## STRUTTURA INTERNA DEL CUBLOC

Il CUBLOC è un dispositivo ibrido che implementa un controllore Ladder e un processore Basic, garantendo che i programmi eseguiti dai suddetti siano totalmente indipendenti, tanto da poter parlare, in questo caso, di un vero e proprio multitasking (esecuzione contemporanea di programmi task-differenti), la cui struttura intera è riprodotta in Fig. 1. L'interprete Basic ed il processore Ladder condividono la memoria flash per la memorizzazione dei programmi utente; anche le porte di I/O sono condivise fra i due controller, consentendo libero accesso a questa da parte di entrambi.



L'interprete Basic dispone, inoltre di una memoria dati riservata, a differenza del processore Ladder che condivide la sua memoria dati Basic: ciò consente lo scambio continuo di informazioni e la sincronizzazione tra il codice Basic e quello Ladder..

I dati creati in Basic possono essere gestiti solo da tale linguaggio, mentre i dati creati in Ladder possono essere manipolati utilizzando entrambi i linguaggi.

La memoria programma Basic e Ladder condivide la stessa Flash Memory, per un totale di 80 KB, ma modelli futuri garantiranno un aumento sostanziale di tale capacità.

Il chip interno del CUBLOC è basato su microcontrollore Atmel a 8 bit RISC, in fig.2 è riportato lo schema a blocchi interno.

La memoria di programma, di tipo Flash, è complessivamente pari ad 80kb per ogni modello, eccetto che per il CB405, che rende disponibili all'utente ben 200k.

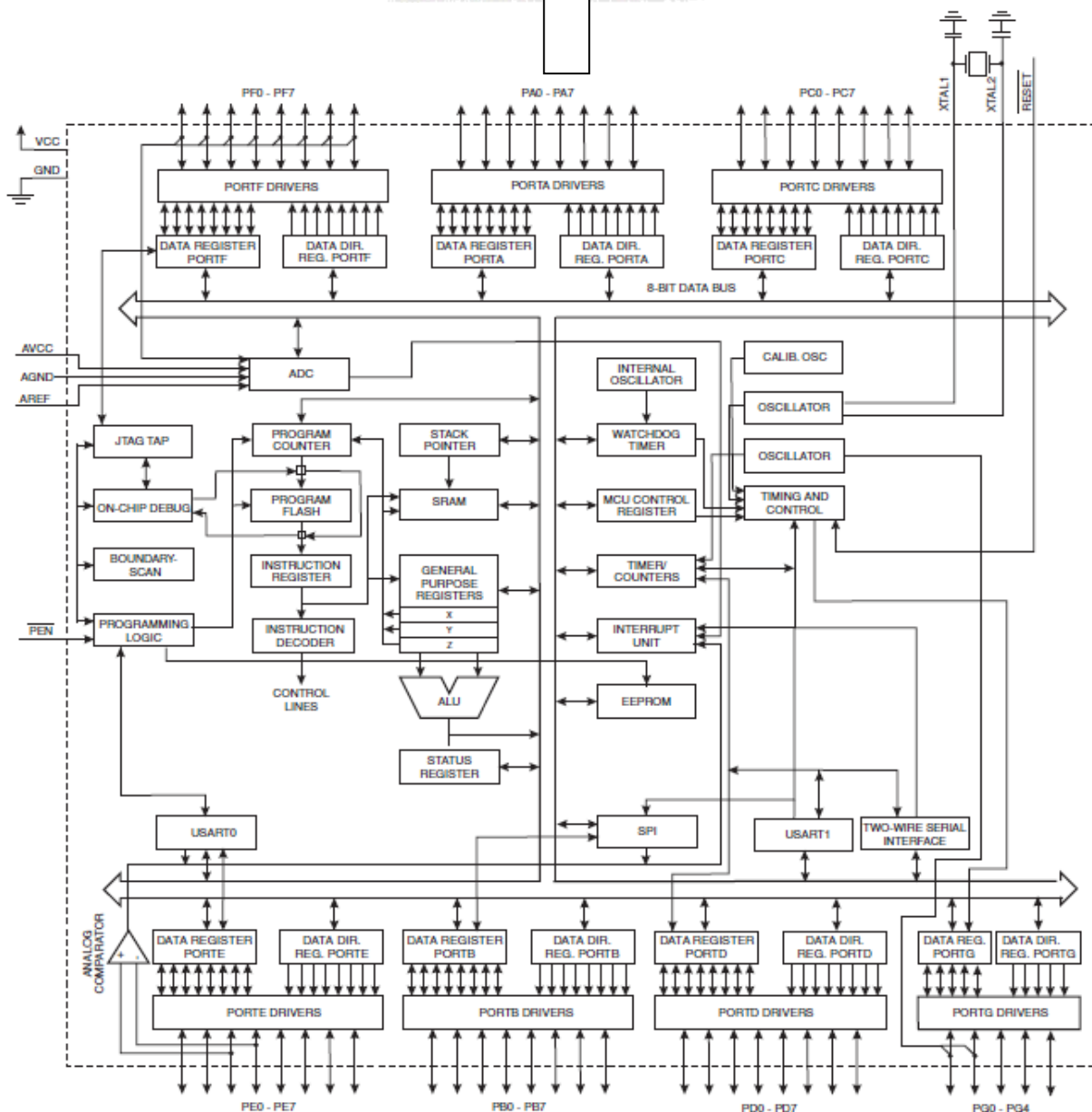
In tutti i modelli sono presenti ulteriori 4kb di memoria EEPROM per la memorizzazione di dati ritentivi (cioè che vengono mantenuti anche se viene a mancare la tensione di alimentazione) oltre ad una memoria RAM statica utilizzabile per memorizzare dei dati non ritentivi (come la variabili di programma), di dimensioni legate al modello.

Il modello CB405 dispone anche di 55kb di memoria Heap, gestibile con specifiche istruzioni per la memorizzazione di strutture di dati dinamiche. La velocità di esecuzione di programma dichiarata è di 36.000 istruzioni per secondo, che condizionano il tempo di scansione del programma Ladder, generalmente pari a 10 millisecondi ma che possono essere ridotti in alcuni casi fino a 500 microsecondi attivando una particolare modalità di funzionamento software/hardware chiamata "turbo".

Per quanto riguarda le interfacce disponibili, tutti i modelli sono muniti di almeno due porte seriali (una a livello RS232Ced una TTL), di almeno otto canali di ingresso analogici con ADC a 10 bit, dai tre ai dodici canali di uscita analogici di tipo PWM e di due contatori



hardware ad alta velocità (2 Mhz). Tutti i modello, ad eccezione del CB220, comprendono quattro canali interrupt, che possono essere utilizzati, oltre per le applicazioni tipiche dell'evento di interrupt, anche per l'interazione tra codice Ladder e Basic. Nel modello CB290, infine, è presente anche un modulo RTC (REAL TIME CLOCK). Per quando riguarda la tensione di alimentazione, il CB220 è l'unico a comprendere un regolatore che accetta in ingresso un valore compreso fra 5V e 12V; tutti gli altri dispositivi devono essere rigorosamente alimentati con una tensione di 5V con una tolleranza del 10%. L'assorbimento di corrente con porte scollegate può variare dai 40mA ai 70mA, a seconda del modello di CUBLOC utilizzato.



## COME PROGRAMMARE IL CUBLOC

Il CUBLOC, nonostante le sue ridotte dimensioni, consente l'utilizzo di due linguaggi di programmazione:

- il **Ladder** (tipicamente utilizzato nel mondo dei PLC)
- il **Basic**, linguaggio che non ha certo bisogno di presentazioni.

### Ladder

Il Ladder è il linguaggio principale utilizzato nell'automazione industriale.

La sua struttura è, formata da una organizzazione grafica. Occorre infatti “disegnare” il proprio programma, in maniera corretta, per poterlo eseguire.

Il disegno segue lo schema elettrico.

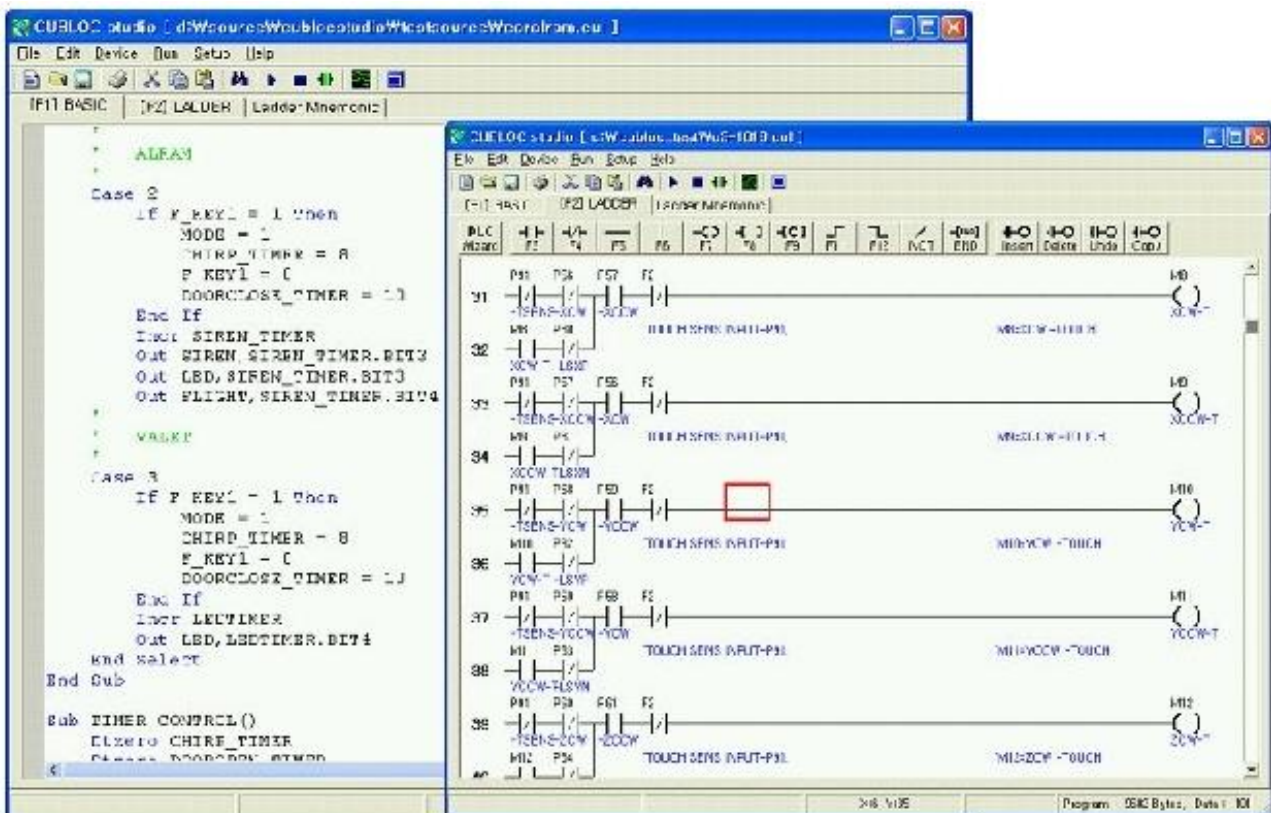
La sua implementazione nei Plc, e adesso nei CUBLOC, rende molto semplice e versatile la programmazione di prototipi industriali ed automatici.

### Basic

Il Basic è un linguaggio molto antico, implementato essenzialmente sui personal computer.

Solo da poco tempo ha visto anche l'applicazione sui microcontrollori, per dare modo ai principianti di programmare questi dispositivi, pur non conoscendone le caratteristiche interne.

E' composto da un set di parole chiave, comandi e funzioni che, strettamente collegate in un file sorgente, danno luogo al funzionamento completo di analisi. Utilizzato assieme al Ladder permette il raggiungimento di ottimi risultati.



### Quindi: Ladder o Basic?

Come detto prima il Ladder è un linguaggio puramente grafico, costituito da blocchi elementari.

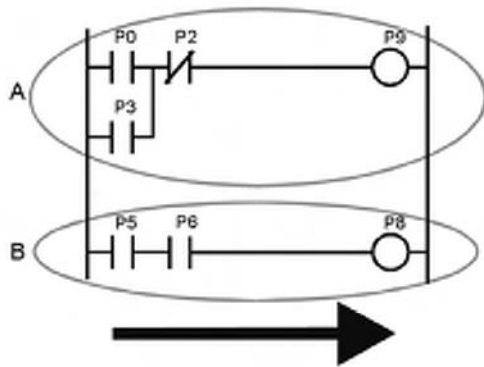


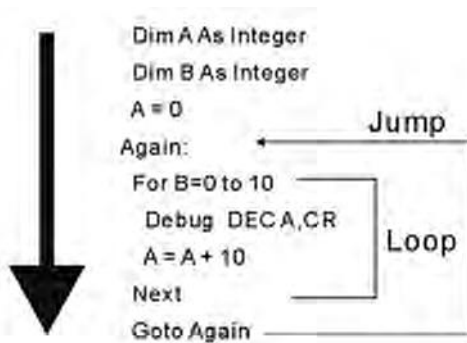
Fig. 6 – Ladder Multitasking

Il suo maggior vantaggio consiste nella possibilità di “processare” i circuiti in parallelo, ovvero in **multitasking**, permettendo di controllare nello stesso istante più configurazioni. Tale risultato è realmente difficile da ottenere con il Basic, quale linguaggio puramente “sequenziale”.

In figura 6 possiamo seguire l’esecuzione contemporanea di due diversi circuiti

“indipendenti”, programmati in linguaggio Ladder.

Il Basic è essenzialmente iterativo, ossia le varie istruzioni sono eseguite una dopo l’altra, in rapida successione, come visibile in figura 7. I maggiori benefici



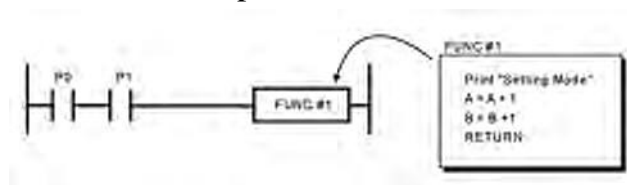
derivanti dalla programmazione del CUBLOC possono essere ottenuti sia avendo un’esperienza della programmazione dei Plc, sia di quella dei microcontrollori.

Un altro vantaggio pratico nell’uso del Ladder, è che esso processa gli Input in un lasso di tempo molto breve, garantendo la loro puntuale rilevazione. In questo modo assicura sempre l’acquisizione degli ingressi, molto più del linguaggio Basic. Inoltre il Ladder è

Fig. 7 – Basic - Sequenziale

orientato agli oggetti, non essendo un completo linguaggio di programmazione. Infatti per applicazioni molto complesse, esso porta alla luce i suoi limiti. Proprio per questo motivo è implementato al Basic, con cui possono essere sviluppate routine ben più complesse. Infine il linguaggio Basic è molto simile all’inglese parlato. Infatti utilizza parole chiave di uso corrente, permettendo così anche agli sviluppatori meno pratici di creare le proprie applicazioni in poco tempo.

Abbiamo detto prima che solamente il Ladder lavora in ambiente multitasking, mentre il



Basic esegue solamente un’istruzione per volta. E’ però possibile rendere il Basic continuamente “residente” implementandolo in un contesto Ladder, costituendo in questo modo una sorta di blocco ad esso proprietario (Fig.8).

Figura 8 - Ladder e Basic multitasking

D’altro canto il Basic possiede una velocità d’esecuzione più alta per la maggior parte delle applicazioni. In ogni caso, ripetiamo, i migliori risultati si ottengono implementando entrambi i linguaggi.

## **Interazione tra i due linguaggi**

Il multitasking del Ladder Logic e la sequenzialità del Basic hanno sicuramente vantaggi che trovano sbocco nella programmazione strutturata. Infatti il Ladder consente processi che al Basic sono letteralmente impossibili, per contro il Basic consente la creazione di semplici routine che il Ladder non può prevedere.

Tanto il Basic quanto il Ladder condividono la stessa memoria flash che ospita i programmi, per un massimo di 80KB di codice. Tanto il Basic che il Ladder possono occupare per intero tale spazio, ma il totale della capacità utilizzata non

matico esistono molte soluzioni, che risolvono praticamente qualunque esigenza. Ma ai fini del presente tutorial, abbiamo ritenuto conveniente utilizzare, sia per motivi pratici ed economici, la seguente componentistica.

può superare gli 80KB. Esistono molte modalità per far interagire i due linguaggi tra loro e con- dividere, a volte, le stesse variabili. Occorre però fare i conti con qualche piccola debolezza, soprattutto quando si prevede il multitasking promiscuo, eliminata la quale si può disporre di un sistema veramente affidabile. Per esempio, una routine scritta in Basic per la sua successiva “incapsulazione” nel Ladder non può essere efficiente come la stessa routine indipendente e autonoma. Ma anche questo aspetto sarà ampia- mente approfondito in seguito.

## **GLI STRUMENTI NECESSARI**

Per poter programmare e verificare un CUBLOC, occorrono alcune risorse hardware e software. Sul mercato elettronico e quello infor-

### **Study Board 1**

È una potente scheda per lo studio e la sperimen- tazione dei CUBLOC. Serve per la programmazio- ne e per il testing delle applicazioni create, senza la necessità di rimuovere, di volta in volta, il chip. Viene fornita completa di manuale, cavo seriale e cavetti di connessione per la breadboard.