INTRODUZIONE AI PICMICRO

Da questo Workshop iniziamo a occuparci di sistemi elettronici programmabili, introducendo una nuova famiglia di dispositivi che si rivela di enorme utilità in tutte le applicazioni in cui sarà richiesta capacità di calcolo: i microcontrollori.

Uno degli elementi fondamentali dei robot è la capacità di acquisire ed elaborare le informazioni. Durante la costruzione di RZB-1 abbiamo visto come questo robot operi effettuando un'elaborazione dei dati di tipo combinatorio, nella quale il comportamento dipende solo ed esclusivamente dallo stato di ingresso dei sensori. In questo Workshop iniziamo a conoscere una famiglia di dispositivi dalle enormi potenzialità, che può offrire spunto per una quantità infinita di impieghi: i **microcontrollori**.

I microcontrollori sono per certi versi molto simili ai microprocessori dei computer, ma integrano memoria e dispositivi di I/O.

COS'È UN MICROCONTROLLORE ›››

L'informatica è oramai parte della nostra vita quotidiana e la sigla CPU (*Central Processing Unit*) è nota praticamente a tutti coloro che possiedono un personal computer. Le **CPU** costituiscono, tecnicamente, il cuore vero e proprio del computer, ossia i componenti adibiti **all'esecuzione dei calcoli e dei programmi**. Ma l'esperienza quotidiana ci insegna che una CPU da sola è di poca utilità senza tutto l'insieme dei dispositivi hardware che la circondano. È così che entrano in gioco schede madri, schede video, moduli RAM e tutti gli altri elementi che, messi assieme, costituiscono la complessa struttura di linee di **comunicazione** e sottosistemi di **memorizzazione** e di **elaborazione** che permettono al computer di operare come sappiamo. Più nello specifico, nei moderni elaboratori possiamo identificare **quattro classi di elementi fondamentali**: la **CPU**, di cui abbiamo discusso in precedenza, la **memoria** (intesa come supporto di memorizzazione dei dati di lavoro a uso della CPU), le **periferiche di I/O**, come la tastiera, la scheda video o le memorie di massa, e i **bus**, ossia i canali digitali attraverso i quali vengono scambiati i dati dai suddetti dispositivi. Immaginiamo ora di incorporare in un unico circuito integrato tutto ciò che serve per il funzionamento di un computer. CPU, memoria di lavoro, memoria di massa e dispositivi di I/O diventano, quindi, parte di un unico microdispositivo caratterizzato dalla possibilità di essere programmato e interfacciato ricorrendo a richieste minimali di hardware supplementare. Ciò che abbiamo di fronte è ciò che viene comunemente chiamato **microcontrollore**. I microcontrollori sono, insomma,

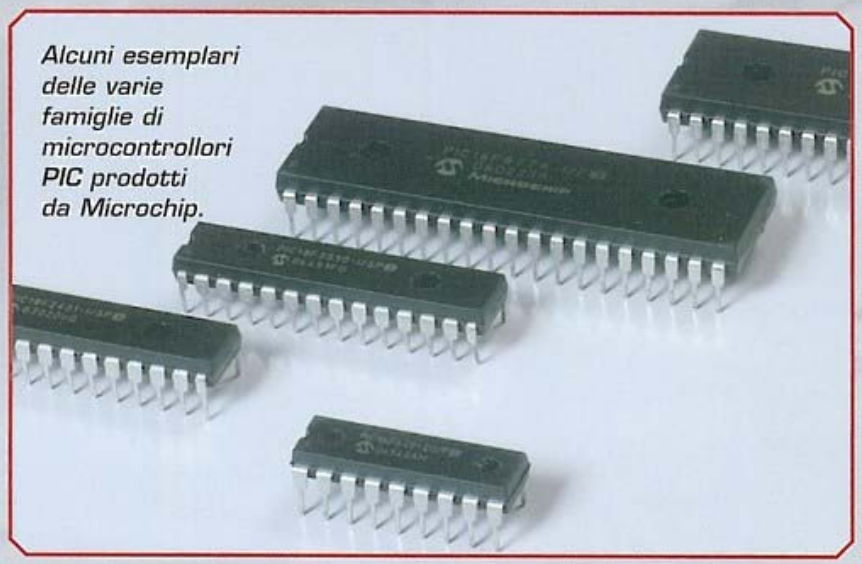
chip programmabili capaci di offrire la versatilità e l'adattabilità di un personal computer, ma con l'ingombro e i consumi di un singolo circuito integrato digitale.

VANTAGGI E LIMITI DEI MICROCONTROLLORI

I microcontrollori (detti anche **MCU**, acronimo dei termini inglesi *Micro Controller Unit*) sono, in sostanza, microscopici computer programmabili caratterizzati da un **costo estremamente ridotto**, dalla **facile reperibilità** e da una **versatilità** quasi infinita, specialmente se ci si sofferma davanti alla miriade di modelli e varianti disponibili sul mercato. Il lato negativo della medaglia, però, è rappresentato dalle performance. Parlare di microprocessori multi-core che lavorano a diversi Gigahertz di velocità o di dischi con capienze del livello dei Terabyte (migliaia di gigabyte) è divenuta oramai una quotidianità. L'idea di avere a disposizione macchine in grado di affrontare miliardi di operazioni al secondo non stupisce più nessuno, tanto meno gli sviluppatori di software che, vista l'attuale situazione di costante incremento della potenza di calcolo, tendono a dedicarsi sempre meno all'importante ruolo di **ottimizzazione del codice** e degli **algoritmi** usati. I microcontrollori, però, non hanno capacità di calcolo paragonabili a quella messa a disposizione dai moderni elaboratori. Per avere un'idea quantitativa di ciò a cui stiamo facendo riferimento puoi osservare la tabella in alto, all'interno della quale sono messe a confronto le prestazioni (in termine di quantitativo di massima) della famiglia di microcontrollori PIC a 8 bit, che utilizzeremo nei prossimi Workshop, con quelle di un personal computer domestico di fascia medio/alta. Come si può osservare, la differenza di risorse hardware è davvero notevole, senza contare che l'architettura RISC (ossia a ridotto numero di istruzioni) dei PIC di fascia bassa non permette di svolgere in modo nativo molte operazioni, come la manipolazione dei numeri decimali (float). In queste condizioni, quindi, la **riduzione degli sprechi di risorse e l'efficienza degli algoritmi** è essenziale per raggiungere buoni risultati.

	COMPUTER	PIC (8 BIT)
Architettura Registri	64 bit	8 bit
Frequenza di lavoro	GHertz (10^9 Hz)	MHz (10^6 Hz)
Memoria di lavoro	Gbyte (2^{20} byte)	Kbyte (2^{10} byte)
Memoria di archiviazione	Tbyte (2^{30} byte)	Kbyte (2^{10}) per la flash; ordine dei 10^2 byte (2^7) per la EEPROM dati

Sopra, una tabella comparativa che mette a confronto le prestazioni di un microcontrollore con quelle di un PC.



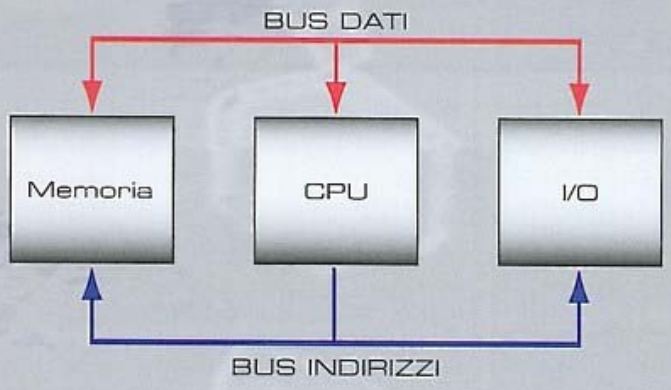
Alcuni esemplari delle varie famiglie di microcontrollori PIC prodotti da Microchip.

I MICROCONTROLLORI PIC

Nelle pagine precedenti abbiamo potuto farci un'idea di cos'è un microcontrollore. Iniziamo a conoscere più da vicino quelle che saranno le MCU che ci accompagneranno nel corso nei nostri esperimenti. Come

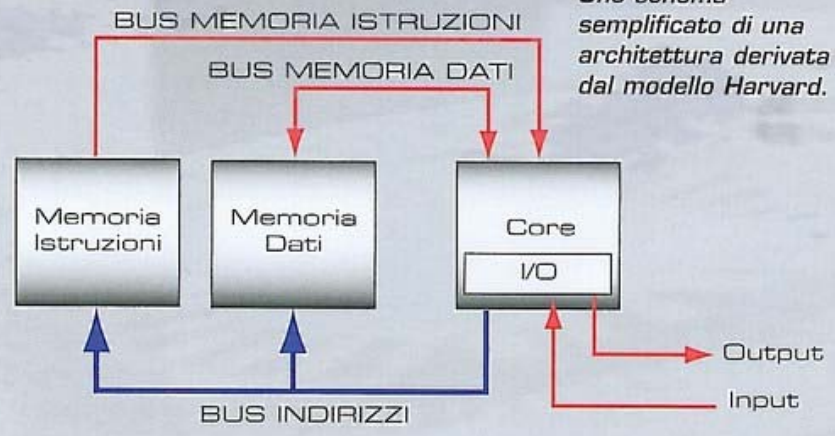
già detto, l'offerta commerciale mette a disposizione dei professionisti e degli hobbisti un'enorme quantità di modelli di microcontrollori. I più famosi e diffusi tra gli appassionati di robotica sono, probabilmente, gli **AVR** (prodotti da **Atmel**) e i **PICmicro** (nome più comunemente abbreviato con la sigla **PIC**), dell'americana **Microchip**. Proprio i PIC ci accompagneranno da questo fascicolo in poi.
Per comprendere meglio come funziona un PIC, iniziamo analizzando la sua struttura interna. Nel secondo paragrafo di questo articolo abbiamo già descritto, a grandi linee, la struttura interna dei moderni elaboratori, struttura che prende il nome di **architettura di Von Neumann**. Come abbiamo visto, in questa particolare architettura (proposta in modo semplificato nello schema in alto a destra) tutti i dispositivi hardware presenti nel computer comunicano attraverso alcuni canali principali (detti **bus**), ognuno dei quali è adibito al trasporto di informazioni specifiche. Nei moderni PC possiamo identificare **tre bus fondamentali**. Tralasciando il **bus di controllo** (non mostrato nello schema), che svolge, come intuibile dal nome, operazioni di controllo sull'hardware, è interessante comprendere i ruoli degli altri due canali rimanenti, chiamati rispettivamente **bus indirizzi** (in blu) e **bus dati** (in rosso). Il bus indirizzi viene utilizzato dalla CPU per 'indirizzare' i dati (sia verso la memoria, sia verso i dispositivi hardware). Allo stesso modo, il bus dati è impiegato per far 'viaggiare' i contenuti informativi tra le diverse periferiche che costituiscono l'elaboratore. In fase di esecuzione, quindi, il programma viene 'allocato' in memoria dal disco, memoria che viene utilizzata dalla CPU anche per archiviare variabili e dati di elaborazione. La CPU, di conseguenza, prendendo il controllo dei bus crea un continuo flusso di dati con la memoria di lavoro, prelevando istruzioni ed effettuando operazioni di lettura e scrittura dei valori delle variabili. La memoria di lavoro diventa, di conseguenza, la discriminante che **limita la dimensione dei programmi e la quantità di dati manipolabili.** L'organizzazione dei PICmicro è, però, molto diversa da quella dei PC.
La loro struttura logica interna è un derivato **dell'architettura Harvard**, caratterizzata dalla **separazione fisica della memoria in memoria riservata alle istruzioni di programma e memoria dedicata ai dati di lavoro**, con una duplicazione dei bus di comunicazione (schema a sinistra). La Harvard

Architettura Von Neumann



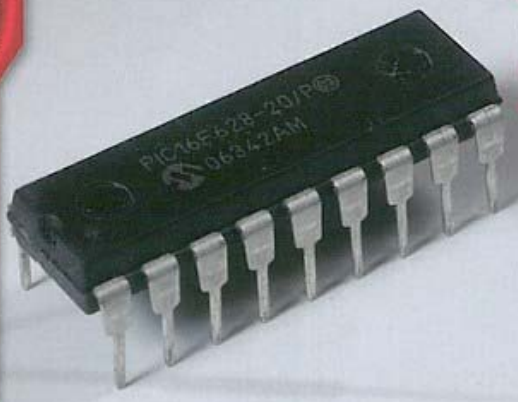
Uno schema semplificato dell'architettura di Von Neumann.

Architettura Harvard

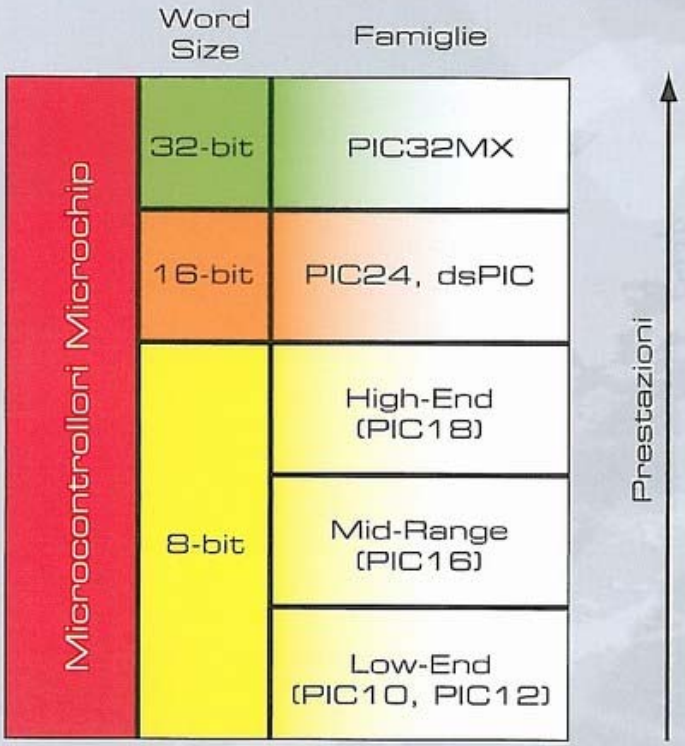


Uno schema semplificato di una architettura derivata dal modello Harvard.

è un tipo di architettura molto particolare, che però può portare alcuni vantaggi, specialmente se applicata a dispositivi con numero di istruzioni ridotto (detti RISC). Il più evidente è identificabile nella possibilità di progettare **bus per il trasferimento dei dati di ampiezza differente da quelli dedicati al trasferimento delle istruzioni.** In alcuni modelli di PIC a 8 bit, ad esempio, il bus dati che collega il core alla memoria di lavoro ha un'ampiezza di 8 bit, mentre il suo equivalente che è posto tra core e memoria istruzioni lavora a 14 bit. Ma in che modo questa soluzione può rendere più efficiente il ciclo operativo del PIC? Senza addentrarci troppo nei particolari (a livello elettronico il problema è tutt'altro che banale), facciamo un breve richiamo al fascicolo 16, nel quale abbiamo parlato per la prima volta di 'opcode'. Abbiamo visto che l'opcode non è altro che una forma di rappresentazione numerica delle istruzioni di basso livello utilizzata nelle architetture di elaborazione digitale. Può capitare che vi siano istruzioni nelle quali è previsto **l'utilizzo di un opcode affiancato da un 'parametro'**, che deve essere compatibile strutturalmente con l'ampiezza del canale riservato ai dati.



Il PIC 16F628 è uno dei microcontrollori a 8 bit più utilizzati a scopo didattico ed è anche quello che ci accompagnerà nei nostri esperimenti iniziali.



Microchip classifica le MCU prodotte in varie classi e sottoclassi, in base ad architettura interna e prestazioni. Nei nostri esperimenti utilizzeremo i PIC a 8 bit.

Se l'ampiezza della linea di trasferimento delle istruzioni fosse analoga a quella del bus dati, per caricare un opcode con il relativo parametro dalla memoria istruzioni sarebbero necessari due step consecutivi: il primo per prelevare l'istruzione vera e propria e il secondo per caricare il parametro associato. **Se si amplia**, però, **il bus di trasferimento delle istruzioni**, portandolo a un numero di bit pari almeno alla somma delle dimensioni di opcode e dati, **è possibile memorizzare in una singola istruzione sia il codice numerico dell'opcode, sia il parametro.** In questa maniera **istruzione e parametro possono essere memorizzate e caricate in un unico passaggio**, ottimizzando il funzionamento.

TANTI MODELLI, MA QUALE FA AL CASO NOSTRO? ▸▸▸

I microcontrollori non sono solo dispositivi in grado di eseguire programmi e calcoli in formato digitale. Integrano, al contrario, una grande varietà di **periferiche** (attivabili via firmware in base alle necessità), che possono rivelarsi di enorme utilità nella realizzazione dei circuiti. Oltre **all'I/O digitale**, possiamo trovare, ad esempio, microcontrollori che dispongono di **convertitori analogico-digitali**, ossia sistemi elettronici, che tramite apposite procedure, consentono di **'rilevare' un valore di tensione da un pin tramutandolo in un numero binario** (un'operazione utilissima per 'leggere', ad esempio, un valore di tensione prodotto da un sensore). Possiamo trovare, inoltre, **comparatori di tensione**,

che permettono di confrontare tensioni su due o più pin del microcontrollore o, ancora, sottosistemi per la **generazione di segnali periodici** (come, ad esempio, i **PWM**) e persino periferiche interne dedicate alle comunicazioni (dalla **comunicazione seriale RS232** dei PC a quelle più orientate ai sistemi elettronici, come il supporto **I²C** o **SPI**). Come scegliere il chip adatto alle nostre esigenze? Attraverso i datasheet, ovviamente, e tramite alcune tabelle messe a disposizione dai produttori, che consentono di confrontare le caratteristiche fisiche e prestazionali dei singoli prodotti. Microchip, ad esempio, produce microcontrollori suddividendoli in varie famiglie. Una prima divisione può essere fatta sul **tipo di architettura**: a 8, 16 o 32 bit. **I microcontrollori a 8 bit rappresentano quelli più semplici ed economici**, ma anche i più adatti ai nostri scopi di sperimentazione a causa del loro costo contenuto e della loro facile reperibilità. L'ulteriore suddivisione della famiglia a 8 bit in **low-end**, **mid-range** e **high-end** è associata alla potenza e alle risorse messe a disposizione dai singoli microcontrollori. Nel box sotto puoi osservare una comparazione tra le caratteristiche tecniche di 7 modelli di PIC (1 low-end di famiglia '12F', 3 mid-range di famiglia '16F' e 3 high-end di famiglia '18F'). Come puoi vedere dai dati presentati, **con il crescere della famiglia, aumentano anche quelle che sono le risorse e le potenzialità di calcolo.** Ad esempio, se nel 12F675 abbiamo a disposizione solo 64 byte di RAM, nel 18F4550 la memoria cresce a 2048 byte, ben 32 volte tanto). Allo stesso modo, anche la frequenza massima di lavoro aumenta salendo di categoria: dai 20 MHz dell'esemplare di serie 12F si passa ai 48 MHz del 18F4550. **Scegliere un PIC significa, innanzitutto, capire di quante e di quali risorse hardware abbiamo bisogno** per non avere risorse sovradimensionate, ma soprattutto per non sprecare soldi. Nel prossimo fascicolo, proseguiremo il nostro percorso alla scoperta dei PIC iniziando a conoscere il microcontrollore **16F628**, ossia il primo chip a cui faremo ricorso per i nostri esperimenti, e cominciando ad affrontare alcune problematiche legate alla programmazione.

UN CONFRONTO TRA PIC›››

In questo box puoi osservare alcuni modelli di PIC di tipo 'flash' a 8 bit appartenenti alle tre sottofamiglie presentate nell'articolo, messi a confronto nelle loro 'funzionalità' e in alcuni parametri operativi. Il micro 16F628 sarà anche il primo che utilizzeremo nei nostri esperimenti.

MODELLO	MEMORIA DI PROGRAMMA (KBYTE)	EEPROM (BYTE)	RAM (BYTE)	Pin I/O	Pin totali	FREQUENZA MASSIMA	PERIFERICHE DI COMUNICAZIONE
12F675	1,75	128	64	6	8	20 MHz	NO
16F628A	3,5	128	224	16	18	20 MHz	USART
16F77	14	0	368	33	40	20 MHz	USART SPI / I2C
16F877A	14	256	368	33	40	20 MHz	USART SPI / I²C
18F2431	16	256	768	33	28	40 MHz	USART SPI / I²C
18F4520	32	256	1536	33	40	40 MHz	USART SPI / I²C
18F4550	32	256	2048	33	40	48 MHz	USART / USB SPI / I²C

GLOSSARIO

LE PAROLE CHIAVE DEI MICROCONTROLLORI

In queste due pagine inseriamo un breve glossario dei termini legati ai microcontrollori che compariranno con maggiore frequenza da questo fascicolo in poi, in modo da metterti a disposizione un punto di riferimento semplice e facile da consultare.

BOOTLOADER: microfirmware che può essere inserito nella memoria istruzioni del microcontrollore per avviare il firmware principale all'accensione del dispositivo. Una delle funzioni tipiche dei bootloader è quella di permettere, unitamente a un apposito client presente su PC, la riprogrammazione della MCU ricorrendo a strumenti alternativi al programmatore hardware (ad esempio, tramite porta seriale RS232 o USB e senza rimuovere il PIC dal circuito).

BROWNOUT: stato di tensione insufficiente al corretto funzionamento del microcontrollore, che può portare al reset automatico della MCU o ad anomalie operative. Può essere causato da sbalzi di tensione imprevisti.

CONVERTITORE ANALOGICO/DIGITALE (ADC): dispositivo elettronico che ha lo scopo di convertire un segnale elettrico continuo (ad esempio una tensione) in uno numerico rapportato a un segnale elettrico costante usato come riferimento.

DSP: acronimo dei termini *Digital Signal Processor*. I DSP sono particolari microcontrollori caratterizzati da elevatissime prestazioni nati per il trattamento digitale in tempo reale dei segnali analogici.

EEPROM DATI: memoria permanente cancellabile e riscrivibile presente all'interno dei PIC, in cui è possibile archiviare dati durante il funzionamento del dispositivo.

FIRMWARE: software compilato ed eseguibile dai sistemi integrati, come i PIC.

FLASH ISTRUZIONI: memoria permanente (normalmente di tipo 'flash') in cui viene scritto in fase di programmazione il firmware del PIC.

FREQUENZA DI CLOCK: indica la frequenza del clock di temporizzazione del microcontrollore. Maggiore è questa frequenza e maggiore è il numero di operazioni elementari svolte dalla MCU (da non confondere con il numero di istruzioni; ad esempio nei PIC ogni istruzione assembler richiede mediamente 4 'colpi' di clock per essere completata).

FUSE: insieme di bit di configurazione dei PIC che vengono settati in fase di programmazione. Inizializzando correttamente questi bit è possibile impostare parametri di funzionamento del PIC quali il tipo di oscillatore utilizzato (XT, HS, RC...), la protezione di determinate aree di memoria (fuse CP), la programmazione a basso voltaggio (fuse LVP) e molto altro.

I2C: sistema di comunicazione master/slave a bus molto utilizzato per scambiare dati tra dispositivi integrati.

ICSP: acronimo di 'In Circuit Serial Programming', una tecnologia che permette di programmare i chip senza rimuoverli dal circuito elettronico in cui sono inseriti.

INTERRUPT: tecnica che permette di rispondere automaticamente a particolari eventi hardware o software 'congelando' lo stato software del microcontrollore per attivare istantaneamente specifiche procedure create dal programmatore.

MCU: acronimo di *Micro Controller Unit*, un'abbreviazione che può essere utilizzata come sinonimo per il termine 'microcontrollore'.

OSCILLATORE: generatore di onda quadra utilizzato per temporizzare il funzionamento delle MCU. I PIC permettono di selezionare diversi tipi di oscillatori, da quelli interni agli integrati a quelli basati su rete RC fino ad arrivare agli oscillatori ad alta frequenza con quarzo piezoelettrico.

OVERFLOW: evento che si verifica in dispositivi quali timer o unità aritmetiche nel momento in cui il risultato di un'operazione (ad esempio un incremento o una somma) genera un risultato numerico che richiede per la sua rappresentazione un numero di bit maggiore di quelli messi a disposizione dall'architettura hardware dei registri di lavoro. In molti dispositivi il verificarsi di un overflow genera un bit di 'carry', ossia di 'riporto' (proprio come avviene in matematica).

PIC: abbreviazione del termine commerciale 'PICmicro' adottato dal gigante dell'elettronica Microchip per indicare i propri microcontrollori digitali.

PORTA: raggruppamento logico/funzionale dei pin all'interno di un microcontrollore. I PIC hanno generalmente a disposizione una o più porte contraddistinte con una lettera dell'alfabeto (PORTA A, PORTA B...).

PRESCALER: divisore di frequenza utilizzato per regolare la velocità di incremento dei timer. Generalmente viene impostato attraverso appositi registri interni al microcontrollore e può assumere valori di divisione uguali alle potenze di 2 (2, 4...).

PROGRAMMATORE: dispositivo hardware da interfacciare a un personal computer (via porta seriale, parallela o USB) che permette, attraverso applicazioni specifiche, di scrivere un firmware nella memoria di programma del dispositivo.

REGISTRO: area di memoria di dimensioni ridottissime (1 byte o poco più) utilizzata per memorizzare le variabili di stato dei microcontrollori o i dati da elaborare.

SET DI ISTRUZIONI: insieme delle istruzioni di base (ossia di basso livello) di un microprocessore. I microcontrollori PIC di fascia bassa hanno un set di istruzioni molto essenziale composto da poco più di 30 istruzioni assembly. Le istruzioni più complesse sono ottenute creando sequenze delle istruzioni di base.

SISTEMI EMBEDDED: sistemi elettronici programmabili in grado di incorporare ed eseguire un software che ne controlla il funzionamento.

TIMER: dispositivo integrato nei PIC con funzionamento autonomo basato su un contatore a incremento automatico sincronizzato con il clock del microcontrollore. I timer sono utilizzati molto frequentemente per la generazione di eventi periodici e per risolvere problematiche di temporizzazione.

USART: standard internazionale per la comunicazione seriale asincrona, utilizzato anche dalle porte seriali dei PC.

WATCHDOG TIMER: sistema di supervisione, attivabile facoltativamente nei PIC, che consente di resettare il dispositivo nel caso in cui si verifichino condizioni di stallo a runtime (deadlock), come loop infiniti. Da attivare solo se necessario.