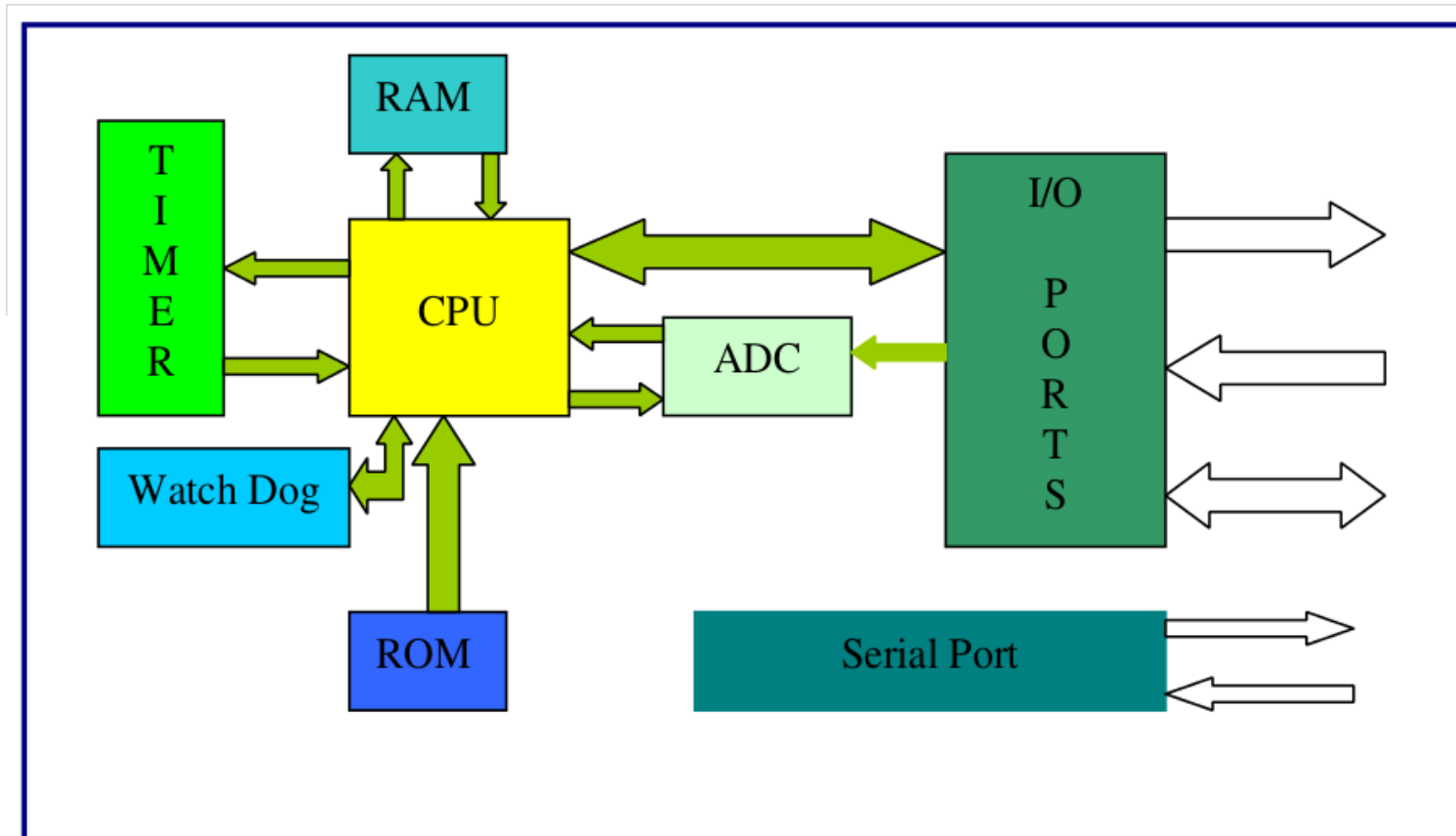


Microcontrollore

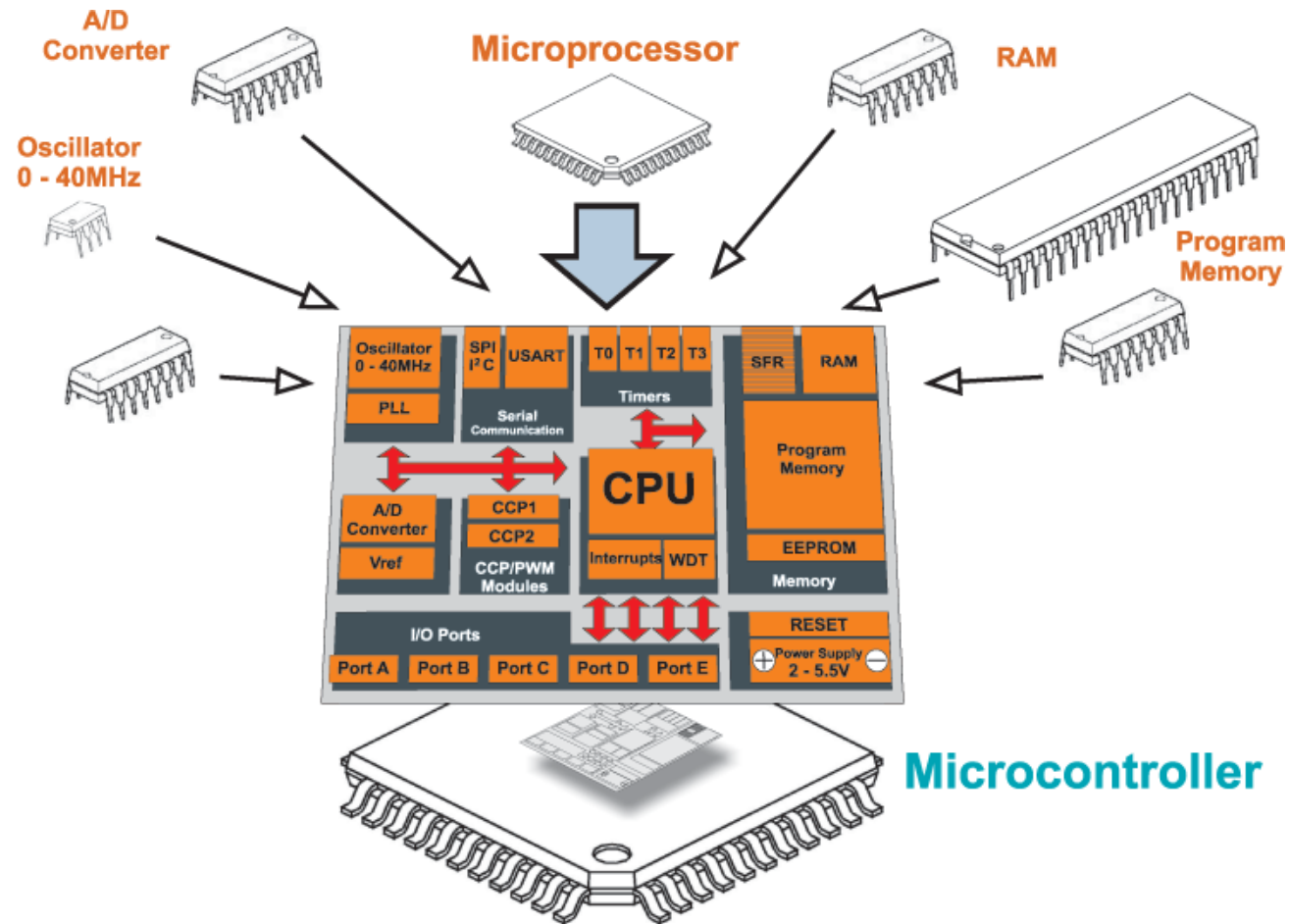
Un **microcontrollore** (*microcontroller* o **MCU**, *MicroController Unit*) è un *single-chip computer*, ovvero un microcalcolatore integrato su un singolo chip. Come suggerisce il nome, il microcontrollore è utilizzato principalmente per realizzare sistemi di controllo digitale e, in particolare, nei dispositivi cosiddetti **embedded**. Si tratta di sistemi elettronici di elaborazione a microprocessore progettati appositamente per una determinata applicazione (*special purpose*) ovvero non riprogrammabili dall'utente per altri scopi. Alcuni esempi di sistemi embedded in cui vengono usati i microcontrollori sono

- controllo di hard disk e di stampanti;
- calcolatrici;
- telefoni cellulari;
- fotocamere;
- autoveicoli;
- sistemi di allarme;
- sistemi di controllo di macchinari industriali;
- elettrodomestici etc etc.

Il microcontrollore si differenzia rispetto al microprocessore in quanto al proprio interno contiene normalmente anche una certa quantità di memoria RAM e di EPROM e vari dispositivi periferici integrati, come timer, convertitori AD etc. Si tratta dunque di un vero e proprio computer completo di tutto ciò che occorre per il suo funzionamento. La figura seguente mostra uno schema della struttura interna di un MCU:



Si noti che generalmente il MCU non viene collegato a chip di memoria esterni (a differenza del microprocessore): l'intero programma di gestione del MCU e i relativi dati devono dunque risiedere sulla memoria interna integrata *on chip*.



A differenza del Personal Computer (PC), che è un dispositivo *general purpose* (cioè di applicazione generale, che può eseguire un gran numero di programmi diversi), i microcontrollori hanno una potenza piuttosto limitata e sono utilizzati in applicazioni specifiche, spesso per eseguire sempre lo stesso identico compito. Per fare un semplice esempio, un microcontrollore utilizzato per controllare le funzioni di un forno a microonde potrebbe semplicemente controllare il livello di temperatura all'interno del forno e il tempo di accensione. Dunque i microcontrollori sono notevolmente più semplici ed economici di un normale PC.

Porte di input e di output

Gran parte dei piedini di un microcontrollore viene utilizzata per comunicare con l'esterno, cioè per ricevere e inviare dati. Ciò è ragionevole dato che i MCU, come si è detto, sono utilizzati prevalentemente nell'ambito dei controlli automatici.

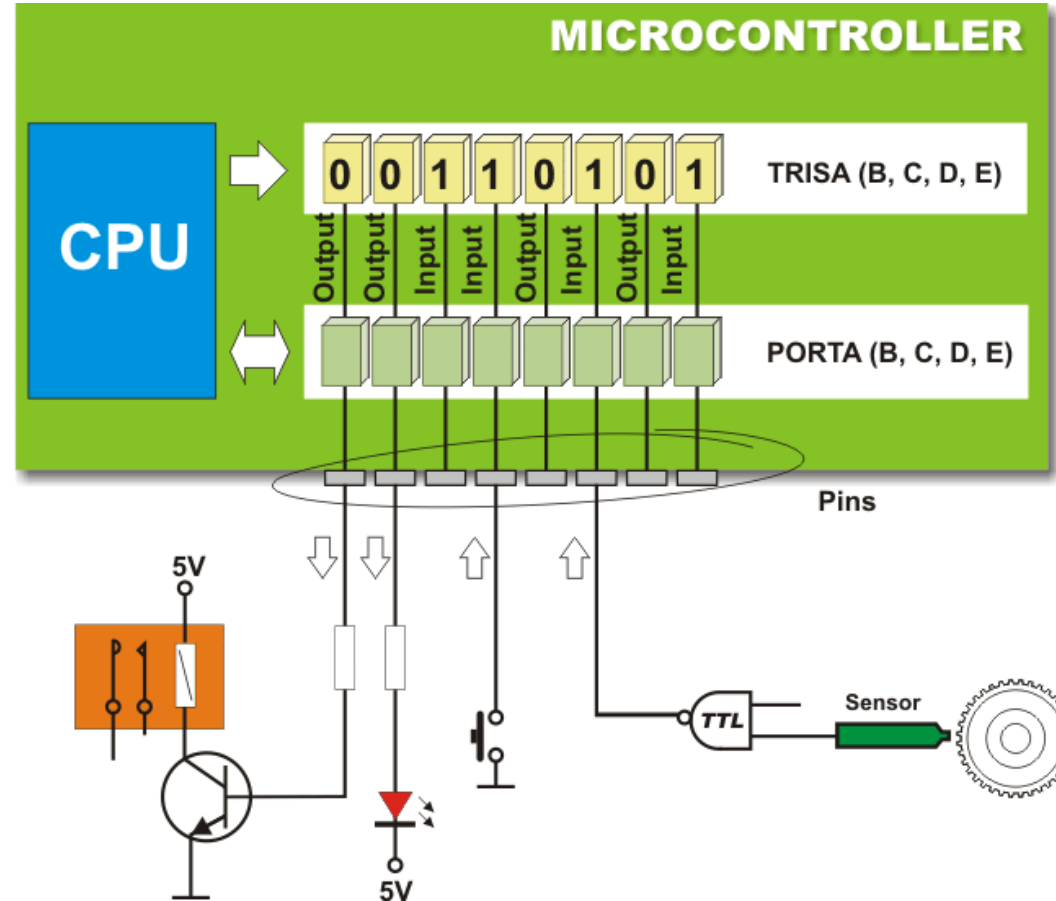
I microcontrollori comunicano verso l'esterno per mezzo di **porte** (*port*). Una porta è semplicemente un gruppo di piedini (PIN) dedicato al collegamento di dispositivi esterni. Spesso i singoli bit di una porta possono essere programmati per funzionare come ingressi (input) o come uscite (output) e la programmazione avviene via software, cioè per mezzo di opportune istruzioni del microcontrollore.

Per esempio scrivendo sulla porta PORTC programmata in uscita (usiamo qui la nomenclatura usata per le porte dei MCU della famiglia PIC) il valore binario b'00000100' (il prefisso b sta ad indicare che si tratta di un valore binario), tutti i pin corrispondenti alla porta vengono resettati a zero tranne il bit 2 (la numerazione convenzionale dei bit parte da destra col bit zero) che viene settato a uno. Se il pin corrispondente al bit 2 di PORTC viene collegato a un LED, è possibile accendere il LED stesso attraverso il microcontrollore.

Se i bit di una porta sono accessibili singolarmente dall'esterno in modo parallelo si parla di **porte parallele**. Viceversa in una **porta seriale** i bit vengono generati o letti in sequenza (come accade in un registro a scorrimento).

Inoltre alcune porte di ingresso possono essere connesse internamente a un convertitore ADC e servono dunque per acquisire e digitalizzare una tensione analogica di ingresso. Si tratta dunque di ingressi di tipo analogico.

La figura seguente mostra come i diversi pin di una stessa porta possano essere programmati per funzionare come input o come output. La programmazione avviene attraverso opportuni registri (chiamati TRIS nel PIC): come si vede in figura, un valore 0 nel registro TRISA indica che il corrispondente pin di PORTA è programmato come output, mentre la presenza di un valore 1 indica che il pin è usato come input

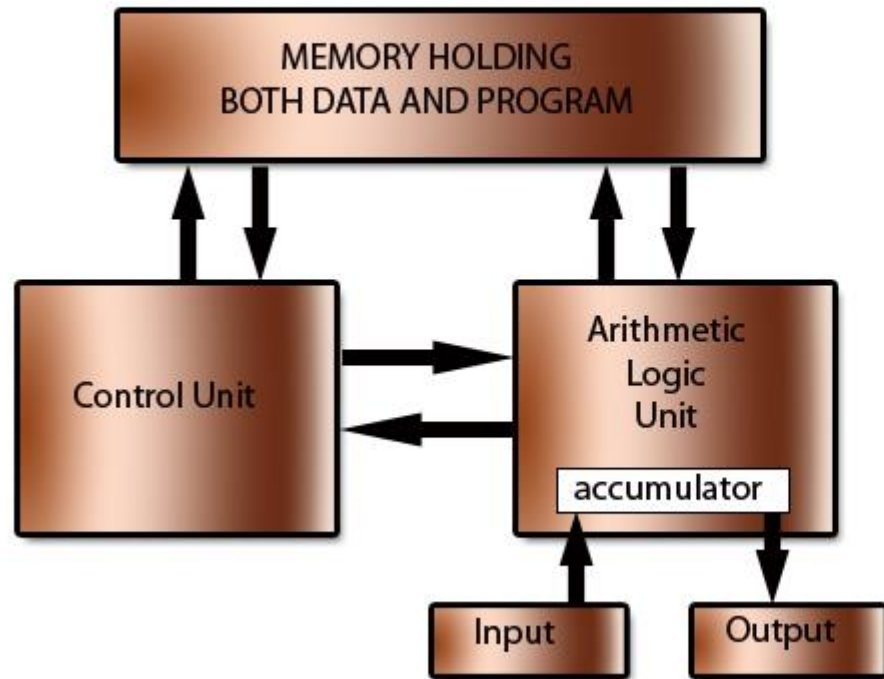


Architettura Harvard

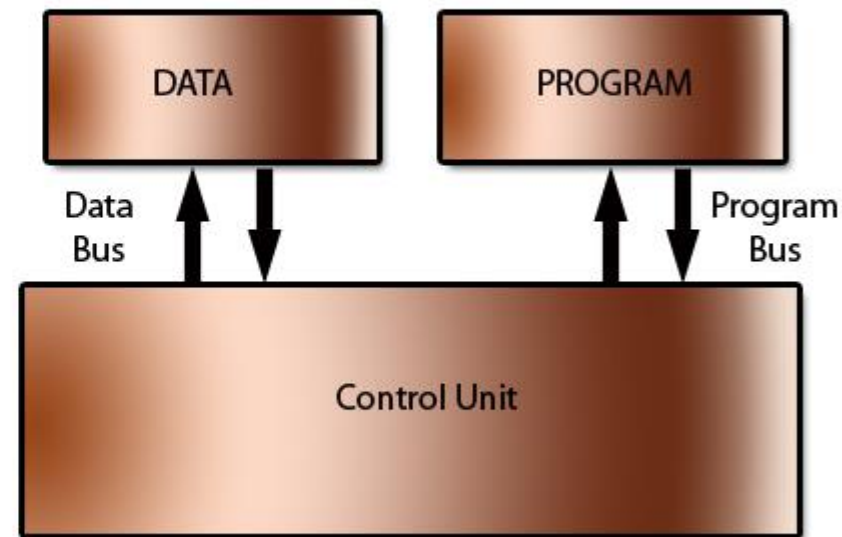
A differenza dei calcolatori classici, i microcontrollori adottano generalmente un'architettura interna detta **Harvard** e caratterizzata dalla separazione fra bus dati e bus istruzioni e, conseguentemente, fra la memoria che contiene i dati e quella contenente le istruzioni. La figura seguente mostra un confronto fra l'[architettura di Von Neumann](#) (comunemente utilizzate nei microprocessori classici) e quella Harvard. Come si può osservare in questo secondo caso esistono due bus separati che collegano l'unità di controllo alla memoria: uno è dedicato ai dati e l'altro alle istruzioni del programma. Invece nella struttura classica di Von Neumann, dati e istruzioni condividono la stessa memoria e lo stesso bus.

Confronto fra l'architettura di Von Neumann (comunemente utilizzate nei microprocessori classici) e quella Harvard

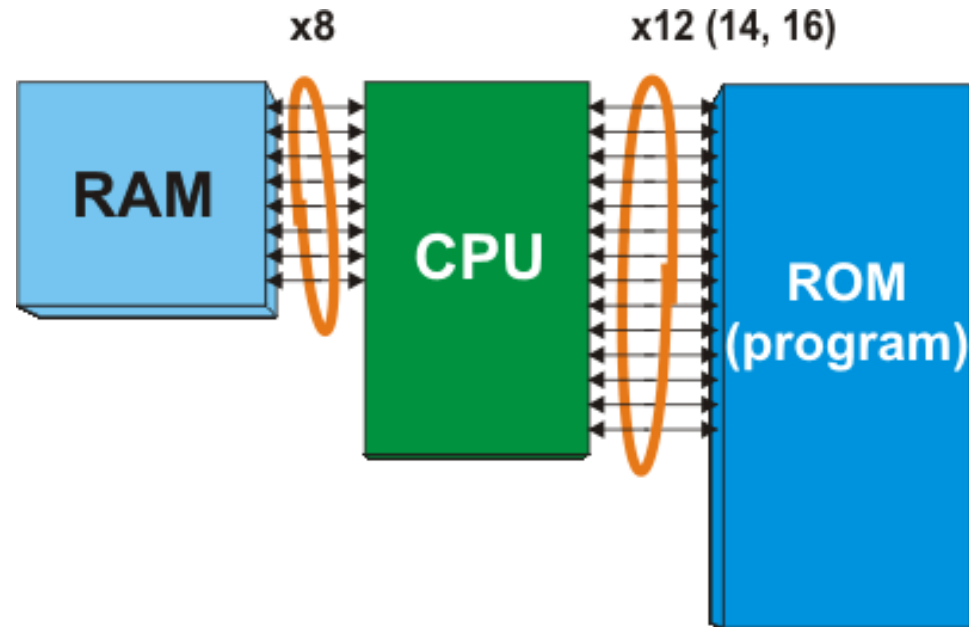
The Von Neumann or Stored Program architecture



The Harvard architecture

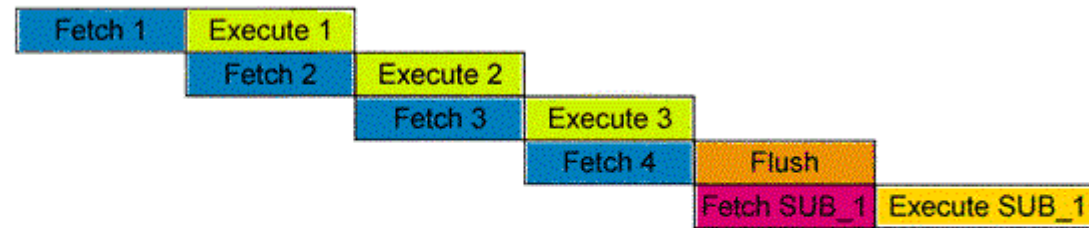


Questa separazione fra i due bus e le due memorie permette ai microcontrollori di avere istruzioni di lunghezza diversa rispetto alla parola dei dati. Per esempio la figura seguente mostra un bus dati a 8 bit e un bus istruzioni che invece può essere a 12 (oppure a 14 o a 16) bit:



La separazione fra dati e istruzioni consente di velocizzare l'esecuzione di un programma eseguendo le istruzioni in parallelo alla fase di caricamento (*fetch*) dell'istruzione successiva (solo le istruzioni di salto causano la perdita di un ciclo). Infatti il bus istruzioni è separato da quello dei dati e dunque eventuali prelievi di dati in memoria (per esempio nel caso di una istruzione con operandi in memoria) non interferiscono col *fetch* delle istruzioni stesse:

```
1. MOVLW 55h  
2. MOVWF PORTB  
3. CALL SUB_1  
4. BSF    PORTA, BIT3
```



RISC

Il set di istruzioni dei microcontrollori è generalmente più limitato rispetto a quello dei normali microprocessori (nei MCU più semplici ci sono solo una trentina di istruzioni diverse). Si parla in questi casi di architettura **RISC** (*Reduced Instruction Set Computer*). Si tratta di una scelta di progetto che privilegia la velocità e la semplicità, rispetto alla varietà e alla versatilità. Il numero di istruzioni ridotto non significa necessariamente minori potenzialità, dal momento che le istruzioni nei RISC tendono a essere più generali, nel senso che una singola istruzione può eseguire compiti piuttosto diversi.

Il set di istruzioni dei microcontrollori è generalmente più limitato rispetto a quello dei normali microprocessori (nei MCU più semplici ci sono solo una trentina di istruzioni diverse). Si parla in questi casi di architettura **RISC** (*Reduced Instruction Set Computer*). Si tratta di una scelta di progetto che privilegia la velocità e la semplicità, rispetto alla varietà e alla versatilità. Il numero di istruzioni ridotto non significa necessariamente minori potenzialità, dal momento che le istruzioni nei RISC tendono a essere più generali, nel senso che una singola istruzione può eseguire compiti piuttosto diversi.

Inoltre nei microcontrollori il tempo di esecuzione è generalmente lo stesso per tutte le istruzioni (di solito 2 o 4 cicli di clock, a seconda del microcontrollore).

Sistema di sviluppo

Un [sistema di sviluppo per MCU](#) consente di scrivere programmi, debuggarli e caricarli nella memoria flash (EPROM) interna al microcontrollore.

Per quanto riguarda la scrittura di programmi, questa può essere fatta usando il linguaggio assembly oppure versioni ridotte di linguaggi ad alto livello (es. il C).

Il programma può quindi essere compilato, testato e debuggato usando opportuni simulatori.

Infine il programma binario in linguaggio macchina può essere caricato sul microcontrollore usando opportuni circuiti di interfaccia che consentono di collegare il MCU con un ingresso USB del PC.