



capace di modificare parametri o istruzioni per migliorare il proprio adattamento all'ambiente.

Immaginate un'azienda che si dedichi alla costruzione di quattro articoli diversi con caratteristiche simili. Ogni articolo utilizzerà parecchi sensori e attuatori, con il loro programma particolare.

L'azienda potrebbe tenere degli stock per ogni articolo e scrivere un programma diverso per ognuno di essi, però questo potrebbe risultare antieconomico. Potrebbe invece decidere di realizzare i suoi prodotti in modo standard, per poter cambiare un sensore/attuatore a seconda della domanda, mediante un unico programma da configurare per l'utente al momento della vendita.

Schema elettronico

Nella figura è presentato un possibile schema elettronico, dove i sensori e gli attuatori possono cambiare e sarà il venditore del prodotto che accoppierà gli uni o gli altri al momento della vendita.

Gli unici fissi saranno gli interruttori di configurazione, posti in una posizione protetta. A seconda del valore che avranno la prima volta che si inizializza il microcontroller, determineranno il

programma da seguire successivamente, e quindi determineranno il tipo di articolo.

Organigramma e programma commentato

Di seguito presenteremo due organigrammi, uno del programma principale e l'altro della routine di scrittura della FLASH. Il primo riquadro è quello di inizializzazione, tuttavia si passerà da questo punto solo la prima volta che si esegue il programma: l'istruzione che verrà modificata per iniziare da differenti punti a seconda della configurazione, infatti, è quella dell'indirizzo 0, che farà deviare verso differenti indirizzi di memoria. Per questo motivo, nell'inizializzazione non si definisce nulla, ma ogni frammento di codice conterrà una parte propria di inizializzazione particolare.

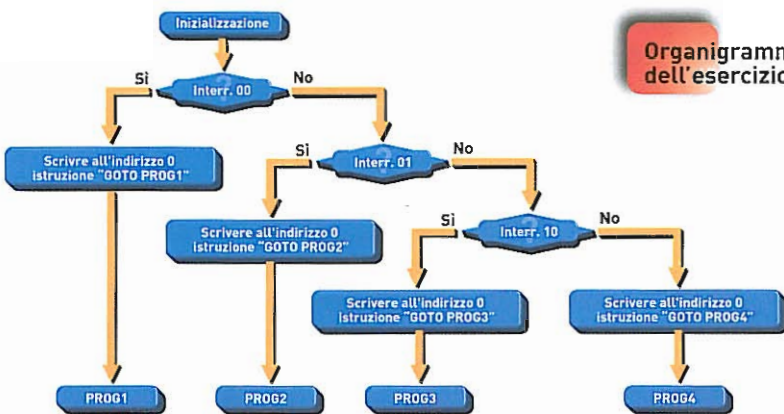
La prima cosa da determinare sarà quindi il valore degli interruttori di configurazione. A seconda del risultato, nell'indirizzo 0 della memoria di programma si scriverà l'istruzione "goto", con il nome di etichetta corrispondente, in modo che quando si esegue nuovamente il programma si vada direttamente là senza la necessità di passare per la verifica



Organigramma di scrittura della FLASH di codice.

degli interruttori; il programma corrispondente viene eseguito già dalla prima volta. Il codice "in eccesso" continuerà a esistere nella memoria anche nelle successive esecuzioni, solo che non si farà accesso a esso. Il secondo organigramma, trattandosi di una subroutine, non ha bisogno di un riquadro di inizializzazione, le operazioni sono simili a quelle della scrittura di una EEPROM, solo che l'indirizzo e il dato sono formati da due registri per volta, e non è necessario entrare in "sleep" dopo una richiesta di scrittura; la memoria stessa non continuerà il programma sino a che non avrà terminato la scrittura, dopodiché dovranno passare due cicli di istruzione senza fare nulla dato che si saltano le istruzioni che arrivano durante questo tempo.

L'indirizzo nel nostro caso sarà lo 0, e il dato il codice dell'istruzione corrispondente.



Organigramma dell'esercizio proposto.