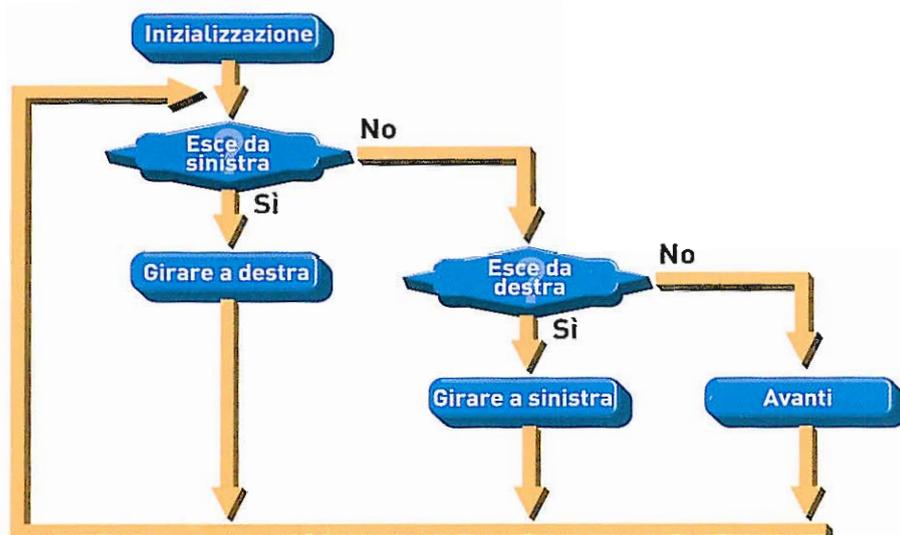


Programma per prendere le decisioni

Un microrobot funziona in base al programma scritto in memoria, che gli indica cosa fare con gli stimoli esterni ricevuti dall'ambiente tramite i sensori. Se noi ci fermiamo qui, il microrobot rimane uno strumento reattivo, dato che reagisce rispondendo sempre allo stesso modo a fronte dello stesso valore del sensore.

Nel caso di un microrobot inseguitore tipico, che percorre la linea utilizzando due unici sensori, si vede come l'organigramma che segue risponda esclusivamente ai valori di ingresso.



I microrobot inseguitori sono dispositivi reattivi.

Proveremo ora a impostare un modo di simulare questa intelligenza, utile in alcuni casi.

Presentazione del problema

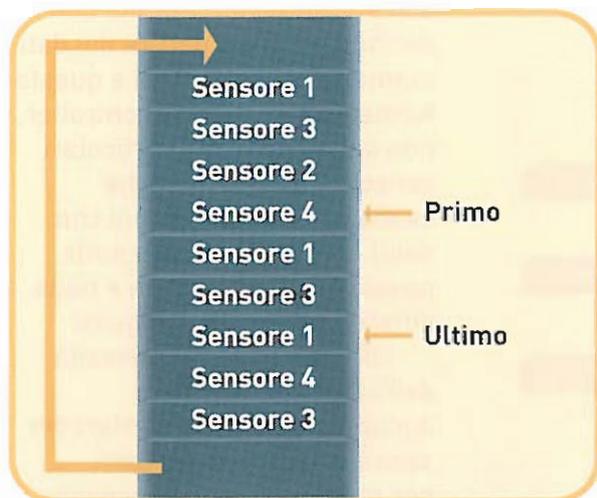
Un microrobot reattivo non può essere definito intelligente, infatti non prende mai l'iniziativa nelle sue azioni, non "pensa".

Idea dell'algoritmo

Invece di reagire solamente in base all'attivazione dell'ultimo

sensore, si può riempire un registro con gli ultimi "N" sensori attivati, in modo che la decisione dipenda da questi sensori come se si trattasse di una memoria degli ultimi eventi accaduti. Essendo l'informazione più completa, la decisione può essere più complessa ed elaborata.

Ogni volta che si attiva un nuovo sensore, lo si deve scrivere nella memoria come ultimo arrivato, un'opzione è scriverlo in modo consecutivo e cambiare gli indici del primo e dell'ultimo. I vantaggi di questo metodo, si traducono in un minor impiego di tempo, dato che il programma è più semplice, e l'ideale per non avere restrizioni di memoria è immaginarla come uno spazio circolare che una volta



Scrittura degli ultimi dati con indice.

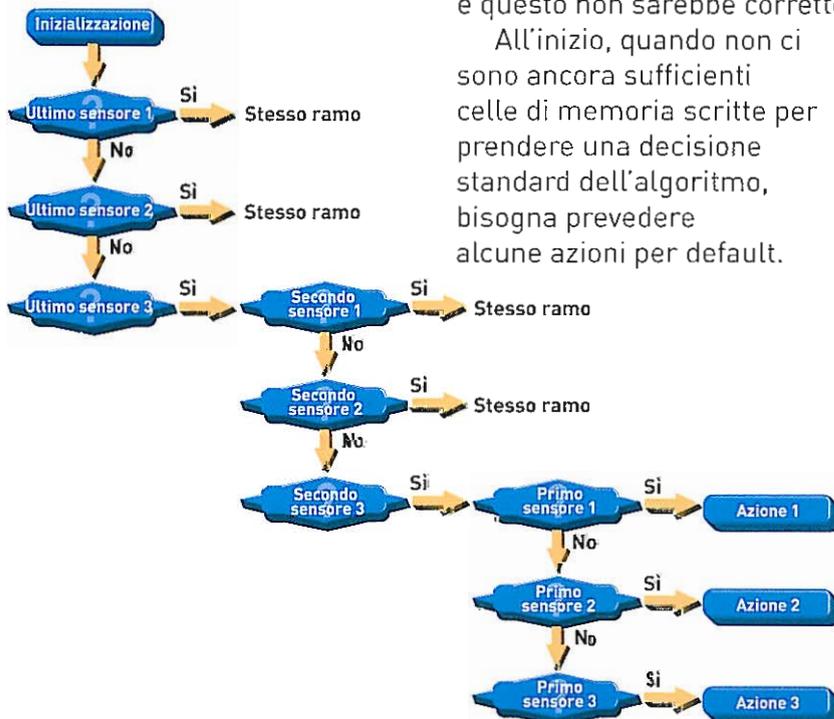


Scrittura degli ultimi dati in posizioni fisse della memoria.

completato riprende dall'inizio. È anche possibile muovere i dati facendo sparire il primo come se si trattasse di uno stack FIFO, in questo caso avremo un maggior tempo di elaborazione però una minor occupazione di memoria. Per questo algoritmo è meglio non utilizzare la memoria EEPROM per scrivere i

dati: nel caso di un errore di alimentazione o di un reset, infatti, ci potrebbero essere intervalli di tempo in cui si producono eventi che non sono acquisiti, quindi le decisioni prese dal microrobot potrebbero utilizzare le ultime celle di memoria scritte, ma che in realtà non sono le ultime, e questo non sarebbe corretto.

All'inizio, quando non ci sono ancora sufficienti celle di memoria scritte per prendere una decisione standard dell'algoritmo, bisogna prevedere alcune azioni per default.



Organigramma tipo di un programma con "memoria".

I modi di utilizzo dei dati scritti saranno molteplici, così come le applicazioni esistenti.

Per cui sarà possibile rilevare quali siano stati gli ultimi "N" sensori attivati o che l'ordine di questa attivazione influisca anche sulla decisione della prossima azione.

In ogni caso gli organigrammi dei programmi risultanti saranno delle istruzioni "if" annidate o alternative tipo "case" a seconda del linguaggio. Quante più possibilità di azioni permette il programma, più intelligente sembrerà la macchina. L'organigramma per non rendere difficile la visione dello schema rappresenta l'ultima delle alternative possibili. Le opzioni segnate come stesso ramo saranno alternative dello stesso tipo di quella mostrata dove si analizzano uno per uno tutti i percorsi. Come si può osservare nell'organigramma scelto influisce l'ordine con cui si attivano i sensori.

Sensori e attuatori

Dato che la parte importante dell'algoritmo è costituita dalle decisioni prese a partire dai dati memorizzati nella RAM e questa forma parte dal microcontroller, non sono necessari particolari sensori o attuatori, anche se sia il numero degli uni che degli altri, si ripercuote sulle possibilità da realizzare e nelle differenti azioni da eseguire.

Influenzerà la complessità dell'algoritmo anche il numero degli ultimi valori dei sensori da tenere presenti per raggiungere una decisione.

