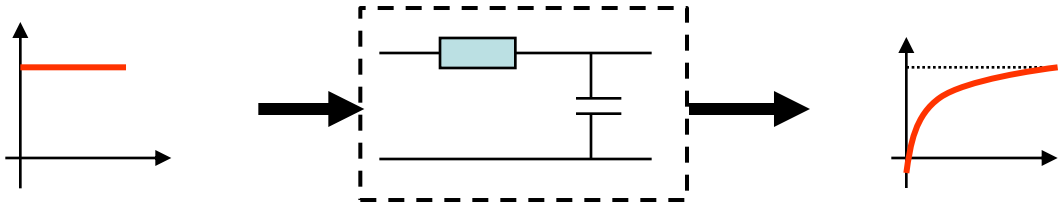
CLASSIFICAZIONE DEI SISTEMI

SISTEMI CONTINUI

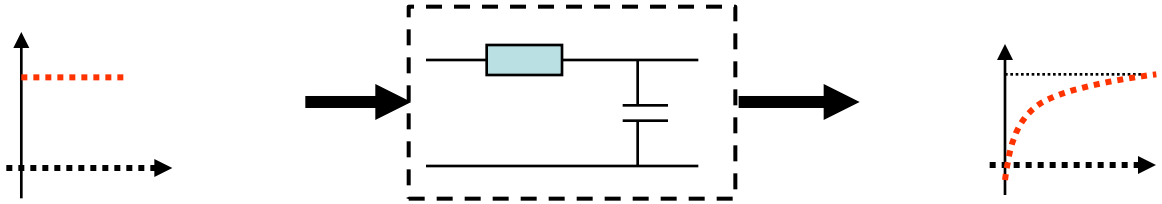
Si tratta di sistemi caratterizzati da variabili continue.

Esempio: Circuito elettrico ohmico - capacitivo



Le variabili evolvono con continuità (***Sistemi continui***) e sono continuamente osservate, cioè il tempo è rappresentato da una semiretta continua (*a* ***tempo continuo***).

Altro **esempio**: Circuito elettrico ohmico - capacitivo



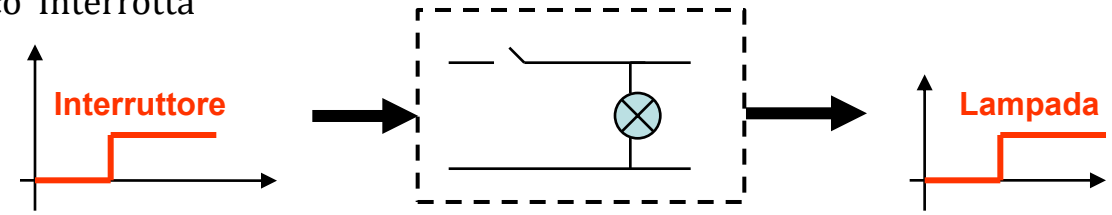
Le variabili sono continue (***Sistemi continui***), ma i loro valori non sono rilevati con continuità, bensì a intervalli di tempo (*a* ***tempo discreto***).

NB: Rientrano in questa classe tutti i sistemi continui controllati mediante controllori digitali.

SISTEMI DISCRETI

Sono sistemi caratterizzati da variabili discrete**.**

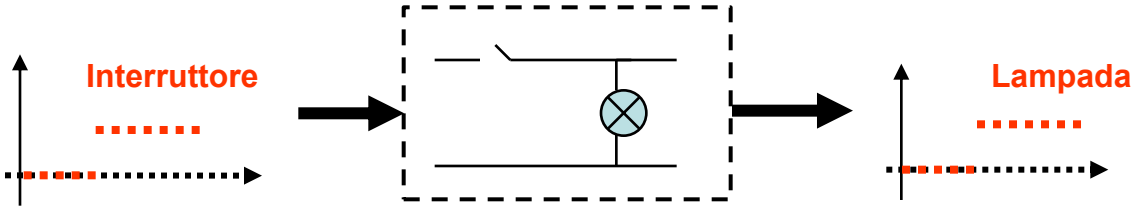
Esempio: Circuito elettrico 'Interrotta'



Le variabili interruttore e lampada possono assumere solo due configurazioni (alto/basso, on/off, aperto/chiuso, acceso/spento) (***Sistema discreto***).

Le loro configurazioni sono rilevate con continuità nel tempo (***a tempo continuo***).

Altro **esempio**: Circuito elettrico 'Interrotta'



Le variabili interruttore e lampada sono discrete (***Sistema discreto***).

Le loro configurazioni sono rilevate a intervalli di tempo (***a tempo discreto***).

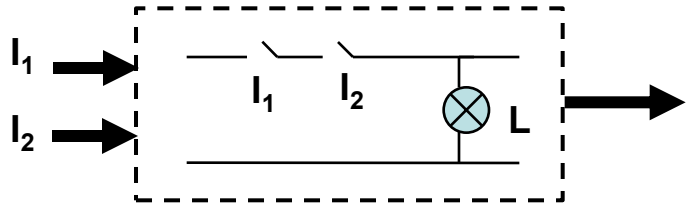
NB: La classificazione del sistema come sistema discreto e quindi dei componenti **Interruttore** e **Lampada** come componenti discreti risponde alle necessità di chi realizza o utilizza il sistema.

Ma per l'industria che costruisce gli interruttori e le lampade, questi non sono affatto componenti discreti. La **transizione da uno stato all'altro** non è istantanea, ma è caratterizzata da una **progressione continua** di cui tener conto in sede di costruzione del componente.

I sistemi discreti a tempo discreto si dividono in due classi:

COMBINATORI

Sistemi in cui le *uscite* dipendono dal valore attuale degli ingressi.



Sono rappresentati mediante tavole di verità

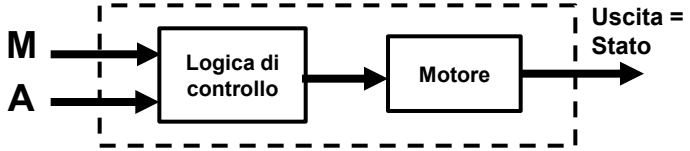
I_1	I_2	L
A	A	OFF
A	C	OFF
C	A	OFF
C	C	ON

NB: sono sistemi in cui *non vi è memoria* della configurazione degli ingressi negli istanti precedenti quello attuale.

SEQUENZIALI

Sistemi in cui le *uscite* dipendono dal valore attuale degli ingressi e dello stato.

Esempio: Comando con pulsanti di Marcia/Arresto di un motore



NB: non è una tavola di verità, in quanto la verità, cioè l'uscita, è condizionata dal valore dello stato, cioè dagli ingressi precedenti.

Situazioni anomale: si è in presenza di comandi contraddittori. La soluzione scelta è di lasciare il motore nello stato in cui è; ma si potrebbe anche decidere di fermare il motore, se vantaggioso per la sicurezza.

M	A	Stato attuale	Stato futuro
R	R	Fermo	Fermo
P	R	Fermo	**Marcia**
R	**P**	Fermo	Fermo
P	**P**	Fermo	Fermo
R	R	Marcia	Marcia
P	R	Marcia	Marcia
R	**P**	Marcia	**Arresto**
P	**P**	Marcia	Marcia

I sistemi **discreti** a **tempo discreto** ***sequenziali*** vengono chiamati anche **SISTEMI A STATI FINITI**.

SISTEMI A STATI FINITI

(AUTOMI)

Gli automi a stati finiti sono sistemi ***dinamici***, ***stazionari***, con *ingressi*, *stato*, *uscite* e *tempo* ***discreti***

Sono ***dinamici*** i sistemi dotati di memoria (Es: flip-flop JK).

La memoria è costituita dallo ***stato*** del sistema.

Definizione: lo *stato* rappresenta una situazione in cui il sistema si trova per effetto delle configurazioni degli ingressi negli istanti precedente

Sono ***stazionari*** i sistemi i cui parametri restano costanti nel tempo.

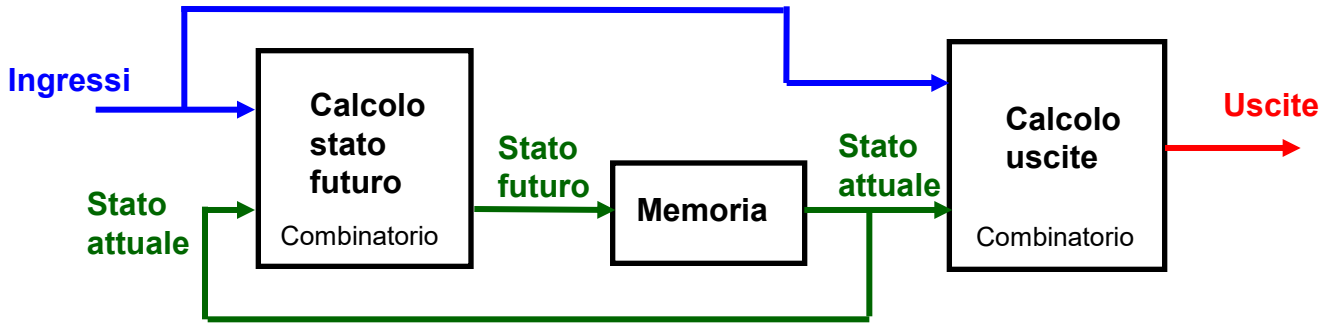
NB: i sistemi a stati finiti sono una classe di sistemi molto importante in cui rientrano

la logica di controllo delle macchine utensili,
il funzionamento di un sistema operativo per PC
Il funzionamento di una lavatrice
……..

AUTOMA DI MEALY

Nel 1955 **Mealy** propose il seguente schema a blocchi per rappresentare un automa a stati finiti:

Modello di Mealy



Nel modello di Mealy l'**uscita** dipende dalla elaborazione, da parte di una rete combinatoria, degli **ingressi** e dello **stato attuale**.

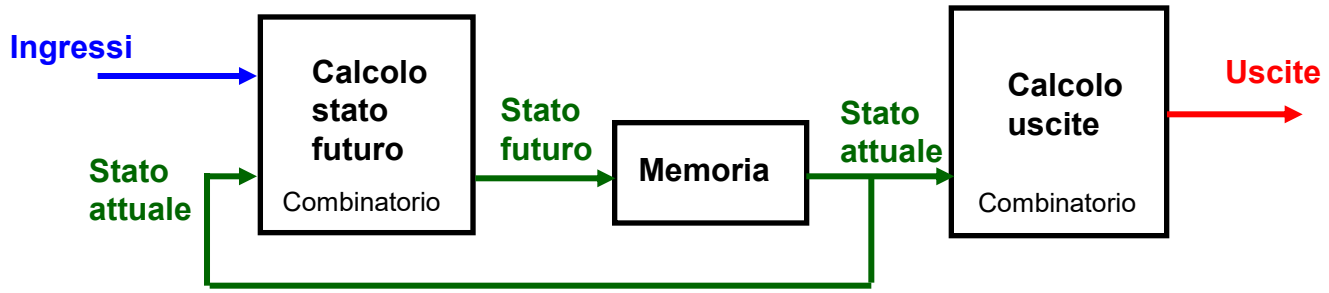
Le uscite commutano *durante la transizione* dallo stato attuale allo stato futuro (sia per i sistemi sincroni che asincroni).

G. H. Mealy propose il suo modello nel trattato *A Method for Synthesizing Sequential Circuits* del 1955

AUTOMA DI MOORE

Nel 1956 **Moore** propose un modello alternativo a quello di Mealy:

Modello di MOORE



Nel modello di Moore l'**uscita** dipende dalla elaborazione, da parte di una rete combinatoria, del solo **stato attuale**.

Edward Forrest Moore (1925 – 2003)**,** statunitense, docente di matematica e informatica. Nel 1956 propose il suo modello con una pubblicazione sull'American Scientific: *Gedanken-experiments on Sequential Machines*

CONFRONTO tra l'automa di **Mealy** e l'automa di **Moore**

La sintesi di un sistema a stati finiti può essere realizzata sia con il modello di Mealy che con quello di Moore.

Tuttavia cambiano la complessità e le prestazioni del sistema costruito.

Automi di Mealy

- Sono più veloci, gli ingressi giungono direttamente sul circuito delle uscite.
- Hanno un minor numero di stati, *potendo* associare le uscite alle transizioni
- Sono possibili uscite spurie transitorie dovute a percorsi diversi, con differenti ritardi di propagazione, cui gli ingressi sono soggetti.

Automi di Moore

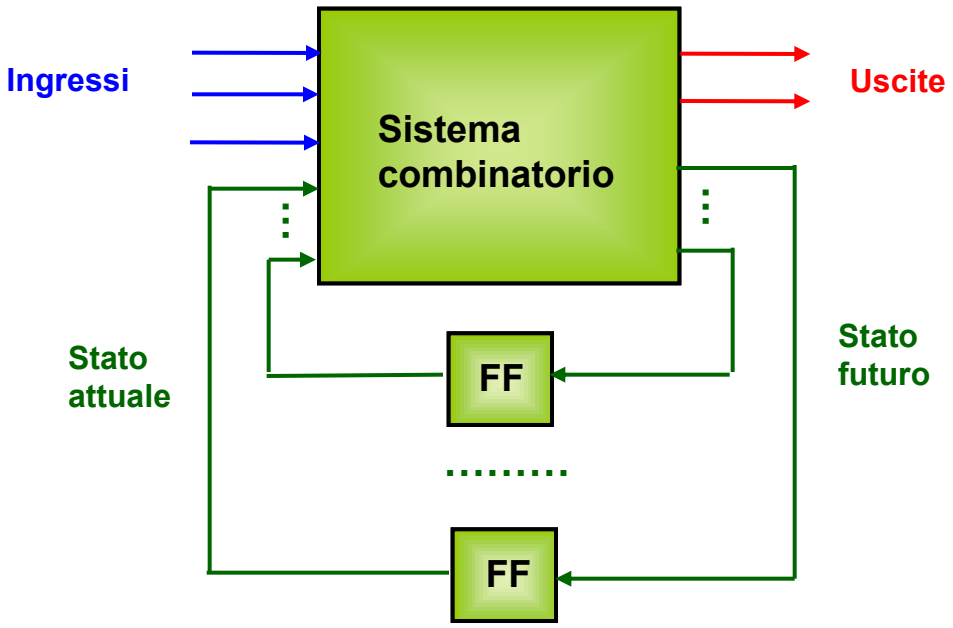
- Sono meno veloci, gli ingressi causano direttamente solo il cambiamento dello stato; le uscite si aggiornano dopo che il nuovo stato si è stabilizzato.
- Hanno un maggior numero di stati, *dovendo* associare le uscite agli stati
- Si ottiene un maggior controllo sull'evoluzione della macchina, grazie all'aggiornamento delle uscite subordinato al raggiungimento del nuovo stato

Sono più facilmente testabili.

MODELLO DI HUFFMAN

Huffman propose un modello generale per rappresentare un automa: un unico sottosistema combinatorio retroazionato con elementi di memoria:

Modello generale di HUFFMAN



Gli *elementi di memoria* sono dei **flip flop** (sincroni o asincroni): uno per ogni variabile di stato.

David A. Huffman (1925 – 1999), pioniere dell'informatica USA. Ha fornito importanti contribuiti alla teoria degli automi a stati finiti, dei circuiti di commutazione, della compressione delle informazioni digitali (codifica Huffman).

MODELLI DI RAPPRESENTAZIONE DEI SISTEMI A STATI FINITI

Il funzionamento di un automa può essere illustrato (descritto) mediante:

- **TABELLA** di *transizione degli stati* e **TABELLA** delle uscite
- **DIAGRAMMI** di *transizione degli stati*

ESEMPIO: riconoscitore di sequenza per sblocco meccanismo

Descrizione: il sistema deve riportarsi allo stato iniziale ogni volta che riceve in ingresso un carattere non valido.

Configurazioni valide per l'ingresso: {A, B, C, D, E, F}

Configurazioni previste per l'uscita: {blocco, sblocco}

Sequenza valida: **BDA**

La corrispondenza della logica di controllo (sistema) al modello di Mealy oppure di Moore è spesso una scelta del progettista.

AUTOMA DI MEALY

La tabella di transizione degli stati descrive l'evoluzione dello stato del sistema in funzione dello ***stato attuale*** e dell'***ingresso:***

Tabella di transizione degli stati

STATO ATTUALE	INGRESSI			
		STATO FUTURO		

STATO ATTUALE	INGRESSI					
	A	B	C	D	E	F
X0	X0	X1	X0	X0	X0	X0
X1	X0	X0	X0	X2	X0	X0
X2	X0	X0	X0	X0	X0	X0

NB: con **X1** si tiene conto (si memorizza) che il primo carattere utile è stato acquisito.

Poiché le uscite dipendono oltre che dallo stato, anche dagli ingressi, per la loro indicazione si deve costruire la tabella delle uscite:

tabella delle uscite

STATO ATTUALE	INGRESSI			
		USCITE		

STATO ATTUALE	INGRESSI					
	A	B	C	D	E	F
X0						
X1			*Comando di blocco*			
X2						

Comando di sblocco

AUTOMA DI MOORE

Poiché le uscite sono definite in corrispondenza degli stati, esse sono indicate nelle stesse celle degli stati futuri.

Tabella di transizione degli stati e delle uscite

STATO ATTUALE	INGRESSI			
	STATO FUTURO / USCITE			

Le uscite commutano in corrispondenza del *raggiungimento* dello stato futuro.

Esercizio precedente:

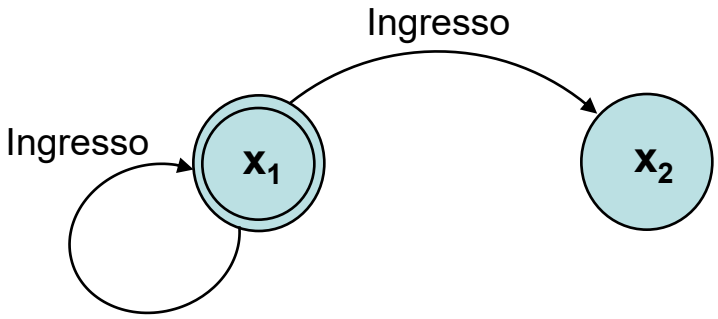
STATO ATTUALE	INGRESSI					
	A	B	C	D	E	F
X0	X0 / Blocco	X1 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco
X1	X0 / Blocco	X0 / Blocco	X0 / Blocco	X2 / Blocco	X0 / Blocco	X0 / Blocco
X2	X3 / Sblocco	X0 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco
X3	X0 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco	X0 / Blocco

NB: con il modello di Moore si rende necessario un elemento di memoria in più: **X3**, cui associare il comando di sblocco.

DIAGRAMMA DI TRANSIZIONE DEGLI STATI

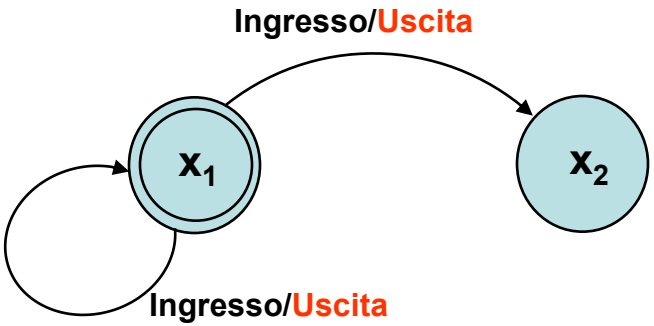
Il funzionamento di un automa può essere descritto anche con i ***diagrammi di transizione degli stati*** (o grafi).

Diagramma di transizione degli stati

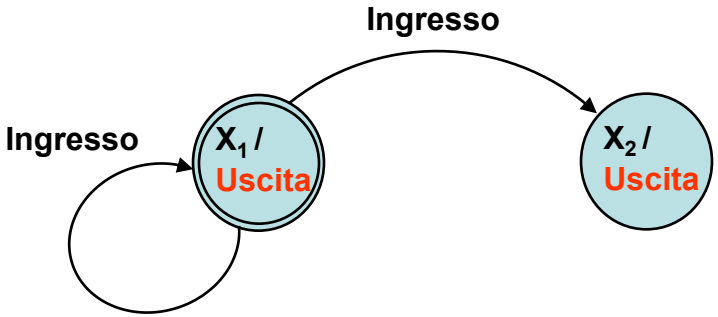


- A ogni nodo è associato uno stato (x)
- A ogni arco è associato un ingresso, la *freccia* indica la direzione della transizione di stato causata dall'ingresso specificato
- Autoanello: transizione che parte e giunge sullo stesso stato
- Lo stato iniziale viene rappresentato con un doppio cerchio.

AUTOMA DI MEALY



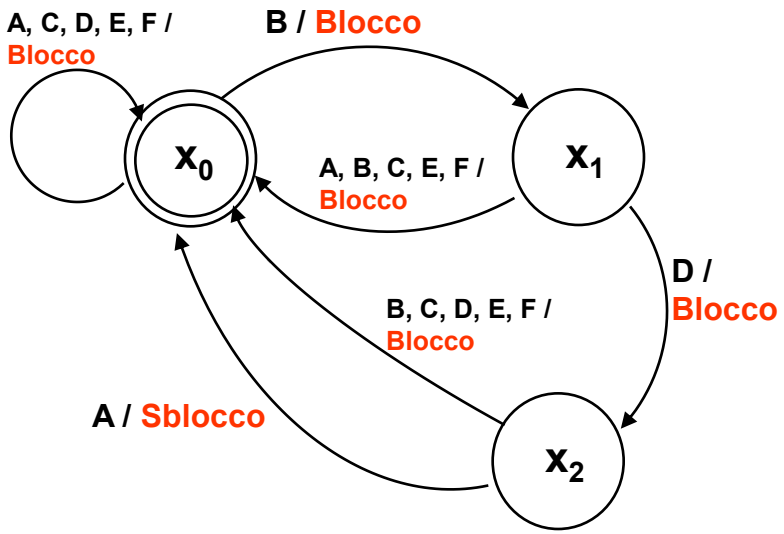
AUTOMA DI MOORE



Esercizio precedente:

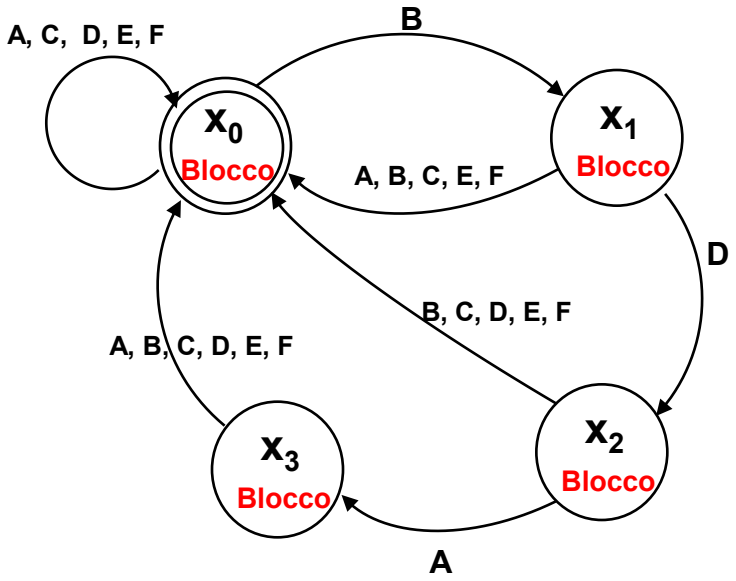
AUTOMA DI MEALY

INGRESSI / USCITE



AUTOMA DI MOORE

INGRESSI



PASSI (Algoritmo) PER LO STUDIO DEGLI AUTOMI

- Descrizione del sistema: occorre disporre di una **dettagliata descrizione** del funzionamento del sistema
- Definizione degli **ingressi**, degli **stati** e delle **uscite** (attribuzione dei simboli e esame delle configurazioni possibili)
- Individuazione di uno **stato iniziale** da cui cominciare la costruzione della tabella o del diagramma
- Costruzione della tabella o del diagramma.

NB: si può cominciare da uno stato iniziale qualsiasi, normalmente si sceglie quello più comodo.

ESEMPIO **Comando di marcia / arresto di un motore**

Descrizione del sistema

Il sistema di controllo è molto semplice: mediante due *pulsanti* si comanda la marcia e l'arresto di un motore.

NB: occorrerebbe prevedere un terzo pulsante che ponga termine al controllo.

Definizione degli ingressi

Gli ingressi sono due pulsanti *normalmente aperti* (**Marcia, Arresto**), che possono assumere solo due configurazioni (P: premuto, R: a riposo).

Definizione degli stati

Lo stato del sistema può essere rappresentato dal ***contatto*** *di potenza* oppure dalla ***condizione*** *del motore*: **F** = fermo, **M** = in marcia.

Definizione delle uscite

L'uscita è rappresentata dal *comando* (**COM**) che si vuole esercitare sul motore. E' un segnale a due livelli (*avvio motore, arresto motore*).

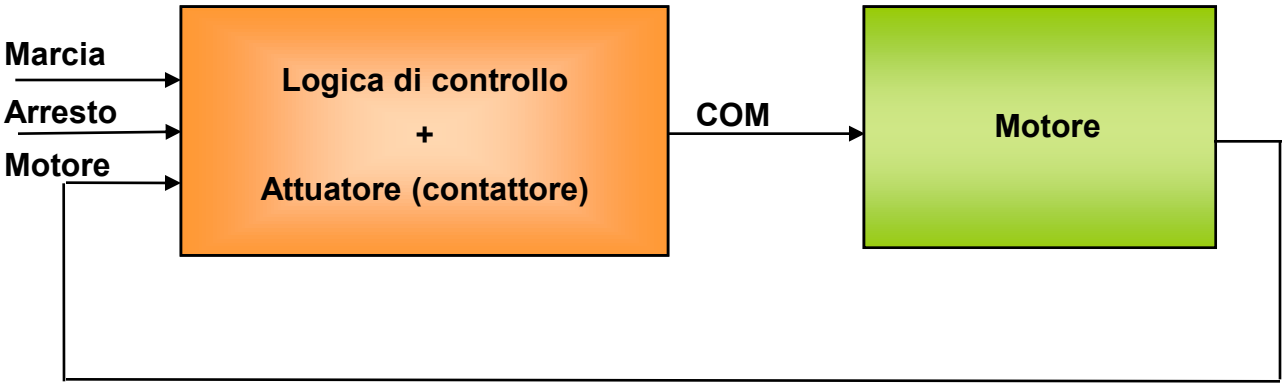
Può essere associato allo stato: **uscite = stato.** Di conseguenza: *modello di Moore.*

Individuazione stato iniziale

Si deve decidere se cominciare con il motore fermo o in marcia. *Ipotesi*: motore **fermo**.

Scrivere il software di controllo.

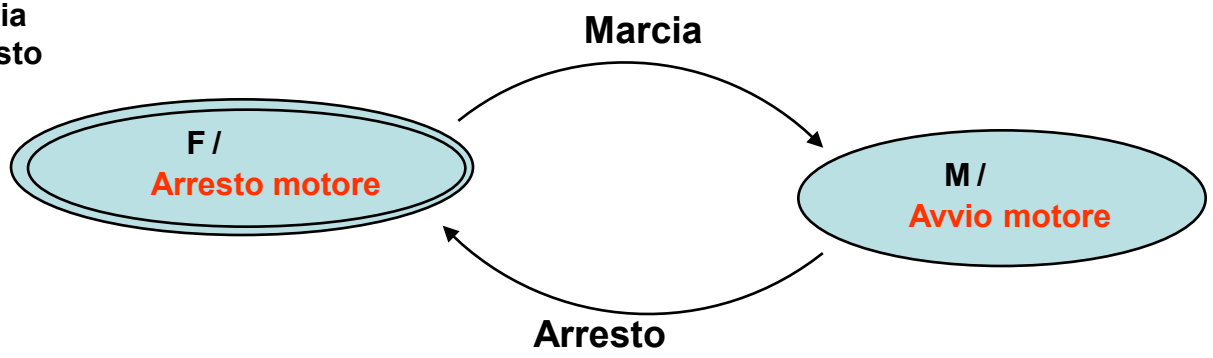
Schema generale



Descrizione della logica di controllo mediante: **diagramma di transizioni degli stati.**

Automa di Moore.

Ingressi: **Marcia**
Arresto



ESEMPIO: COMANDO PUNTO LUCE MEDIANTE RELE' INTERRUTTORE

Descrizione del sistema

Il sistema è costituito da **due** *pulsanti*, **un** *relè interruttore* (con due posizioni di lavoro), **una** *lampada.*

Premendo indifferentemente uno dei due pulsanti lo stato dei contatti del relè interruttore commuta, e di conseguenza commuta anche lo stato della lampada.

Definizione degli ingressi

Gli ingressi sono due pulsanti *normalmente aperti* (**S1, S2**), che possono assumere solo due configurazioni (**P**: premuto, **R**: a riposo). *(su scheda Velleman: Rilasciato = lettura **0**)*

Definizione degli stati

Lo stato del sistema può essere rappresentato dal contatto del relè oppure dallo stato della lampada: contatto aperto = Lampada spenta, contatto chiuso = Lampada accesa. Simbolo **L** = (ON, OFF).

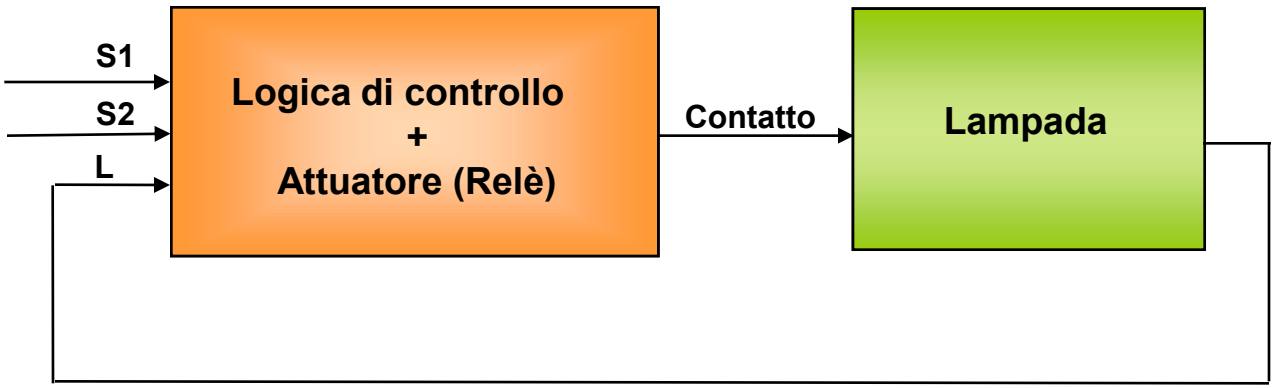
Definizione delle uscite

L'uscita della logica di controllo è rappresentata da un *comando impulsivo* **COM**, che fa commutare il relè.

Individuazione stato iniziale

Si deve decidere se cominciare con il contatto aperto o chiuso. *Ipotesi*: contatto aperto.

Schema generale

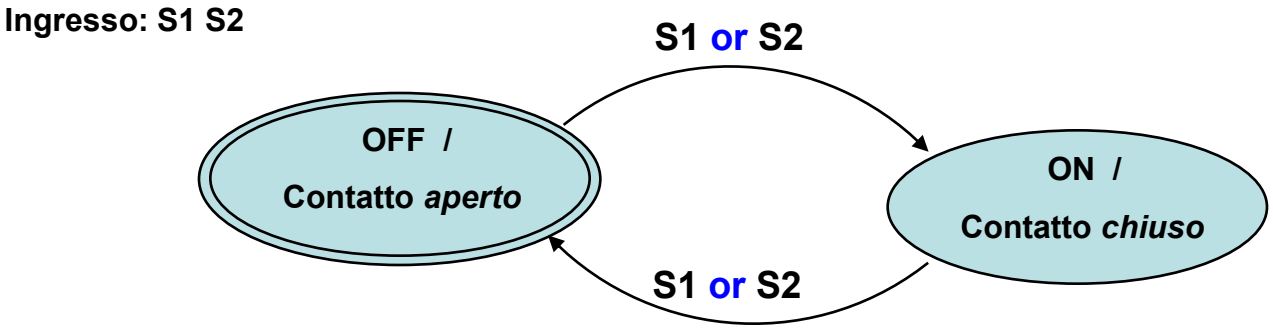


Soluzione

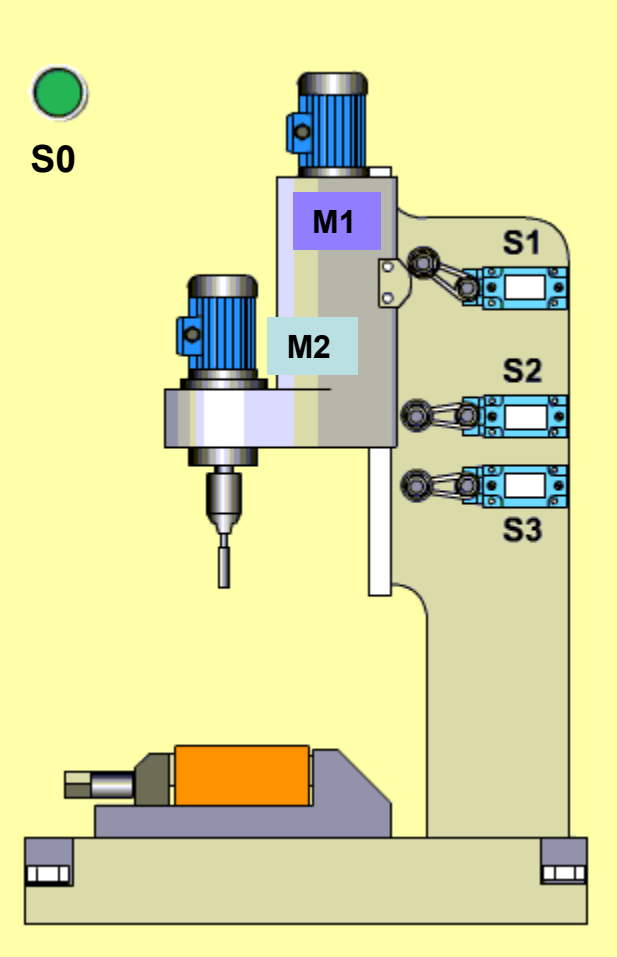
Costruzione del diagramma di transizione degli stati

Occorre scegliere quale rappresentazione utilizzare: Mealy o Moore?

Il sistema è molto semplice; l'*uscita* (il contatto di potenza) può essere fatta corrispondere allo *stato.* In questo caso la rappresentazione è quello di **Moore**.



ESEMPIO: CONTROLLO TRAPANO AUTOMATICO



Descrizione del sistema

Premendo il pulsante S0 si alimenta il motore M1.

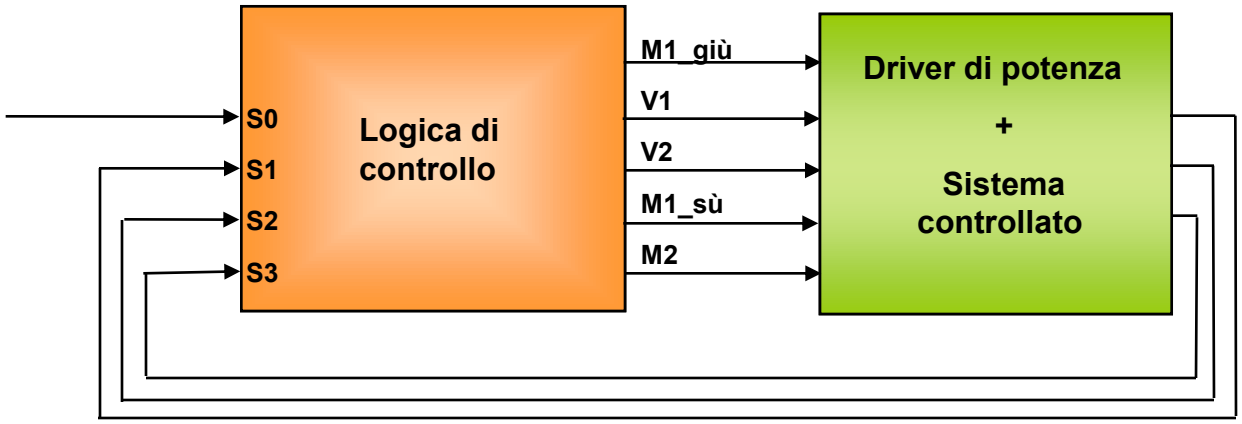
Il carrello porta utensile scende alla velocità v1.

L'attivazione del finecorsa S2 modifica la velocità di discesa del carrello v2 < v1 e attiva il motore M2.

L'attivazione del finecorsa S3 disattiva il motore M2 a fa salire alla velocità v1 il carrello.

L'attivazione del finecorsa S1 arresta il motore M1.

Schema generale



Definizione degli ingressi

Gli ingressi sono costituiti dal pulsante S0 e dai finecorsa S1, S2, S3. ***(tutti NO)***

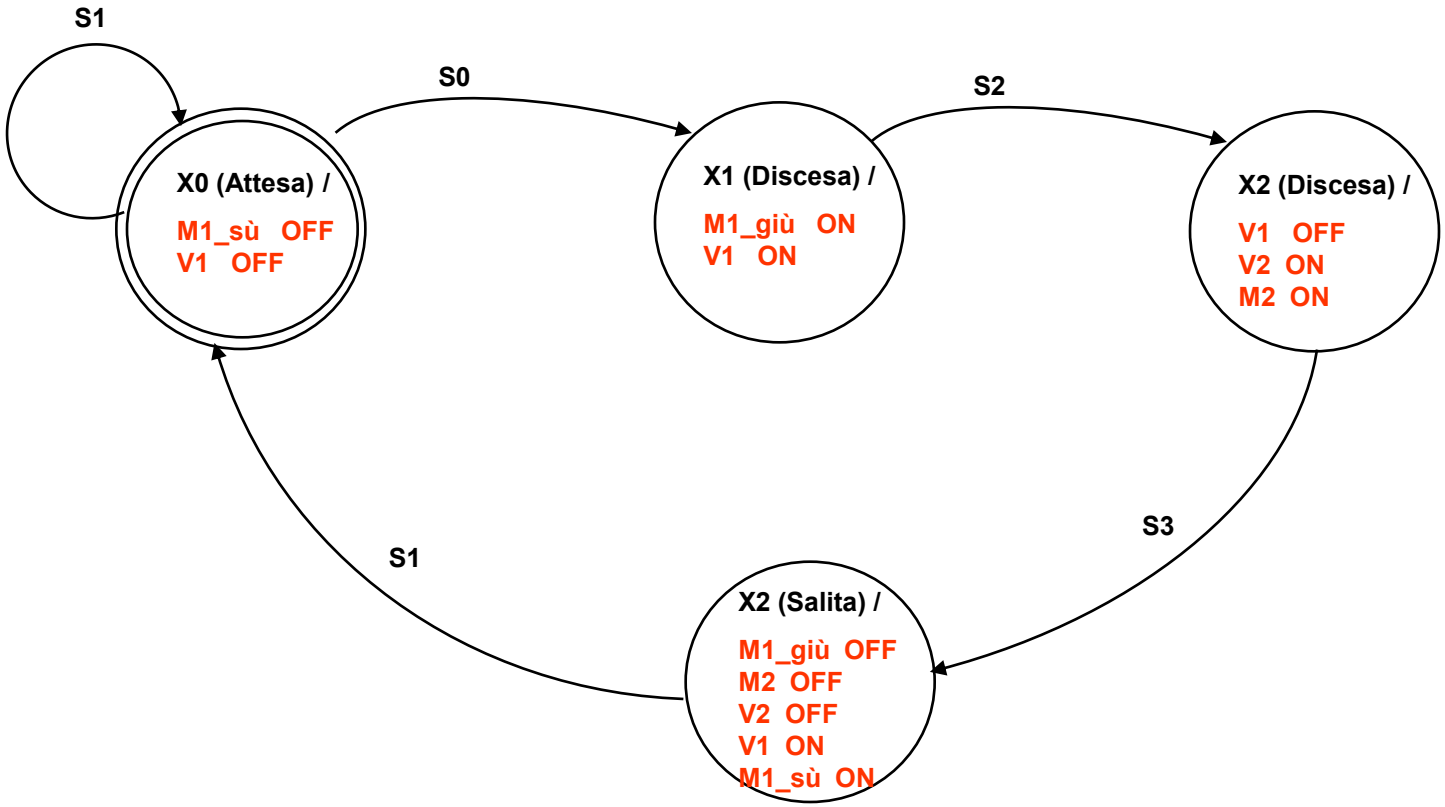
Si tratta di segnali che possono assumere solo due configurazioni:

Pulsante: **S0 = R**ilasciato, **P**remuto *(su scheda Velleman: Rilasciato = lettura **0**)*

Finecorsa: **S1, S2, S3 = R**ilasciato, **P**remuto

Soluzione: **Diagramma di transizione degli stati**

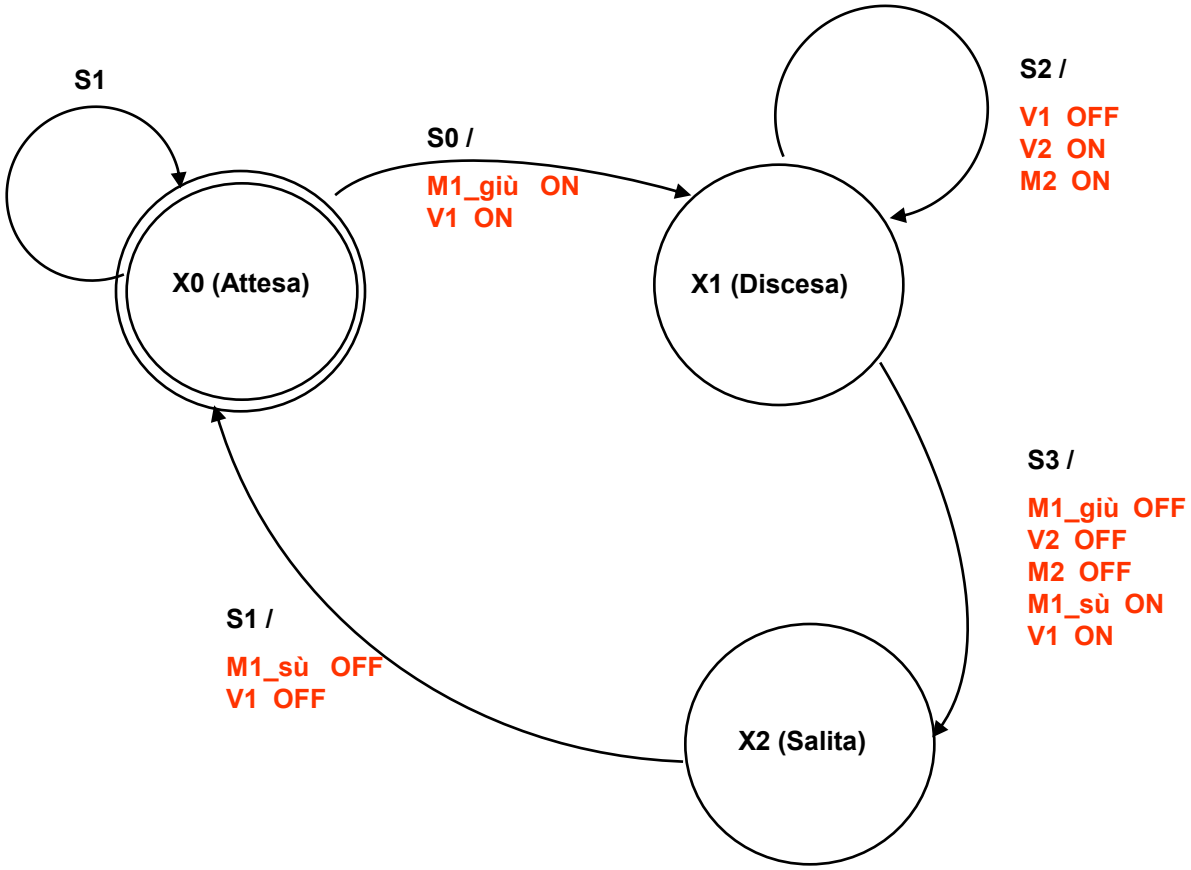
modello di Moore



NB: si dovrebbero assumere opportune precauzioni per evitare l'invio di comandi contraddittori (interblocco)

Soluzione: **Diagramma di transizione degli stati**

modello di Mealy



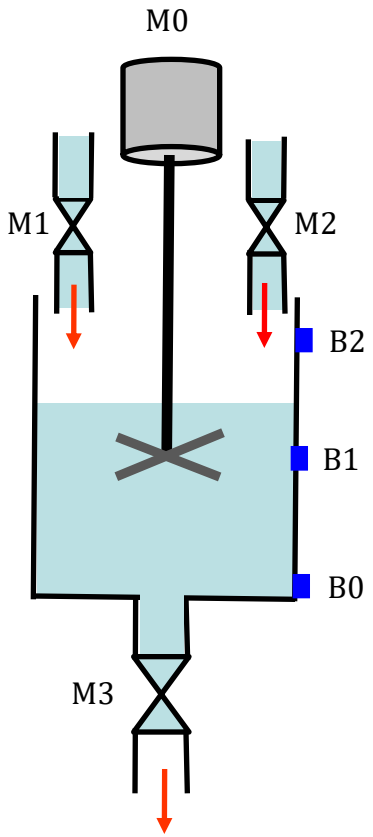
ESEMPIO: MISCELATORE

Descrizione del sistema

Il sistema miscela due liquidi.

Componenti:

- pulsanti S1, S2 normalmente aperti
- elettrovalvole M1, M2, M3 normalmente chiuse
- sensori di livello B0, B1, B2 normalmente aperti (asciutti), si chiudono in presenza del liquido
- motore M0



Premendo S1 si avvia il ciclo, previa verifica di serbatoio vuoto (altrimenti provvedere allo svuotamento).

Fasi del ciclo:

- con serbatoio vuoto (B0, B1, B2 asciutti) , elettrovalvole M2 e M3 chiuse, motore fermo: si fa entrare il 1° liquido aprendo l'elettrovalvola M1;
- durante il riempimento si attiva B0: nessun intervento del controllo
- attivazione di B1: chiusura di M1, apertura di M2 (entrata del 2° liquido) e avvio del motore
- attivazione di B2: chiusura di M2 e apertura di M3 (inizio scarico)
- disattivazione di B2: nessun intervento di controllo
- disattivazione di B1: arresto motore M0
- disattivazione di B0: chiusura M3 e ripetizione del ciclo.

NB: con serbatoio vuoto, premendo S2 il sistema si porta allo stato iniziale di riposo.

Definizione degli ingressi

Gli ingressi sono rappresentati dal *pulsante S1 normalmente aperto* e i *tre sensori di livello normalmente aperti*.

Definizione degli stati

L'identificazione degli stati dipende dal modello:

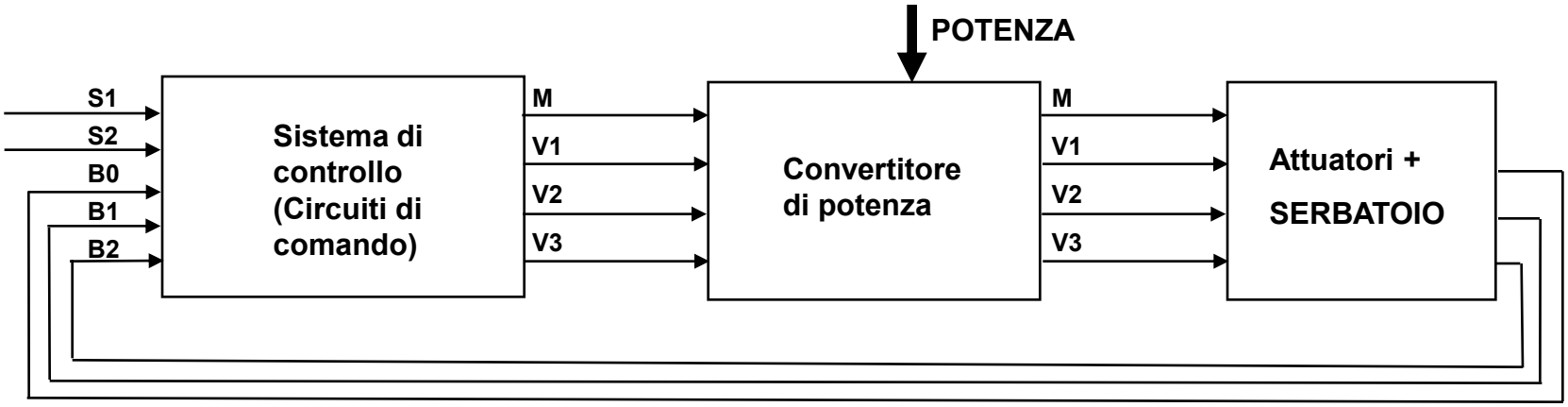
- Moore: ad ogni uscita diversa occorre associare uno stato
- Mealy: possibile accorpare più stati

Definizione delle uscite

Le uscite rappresentano i comandi che il sistema di controllo deve emettere: il motore e le tre elettrovalvole.

Individuazione stato iniziale

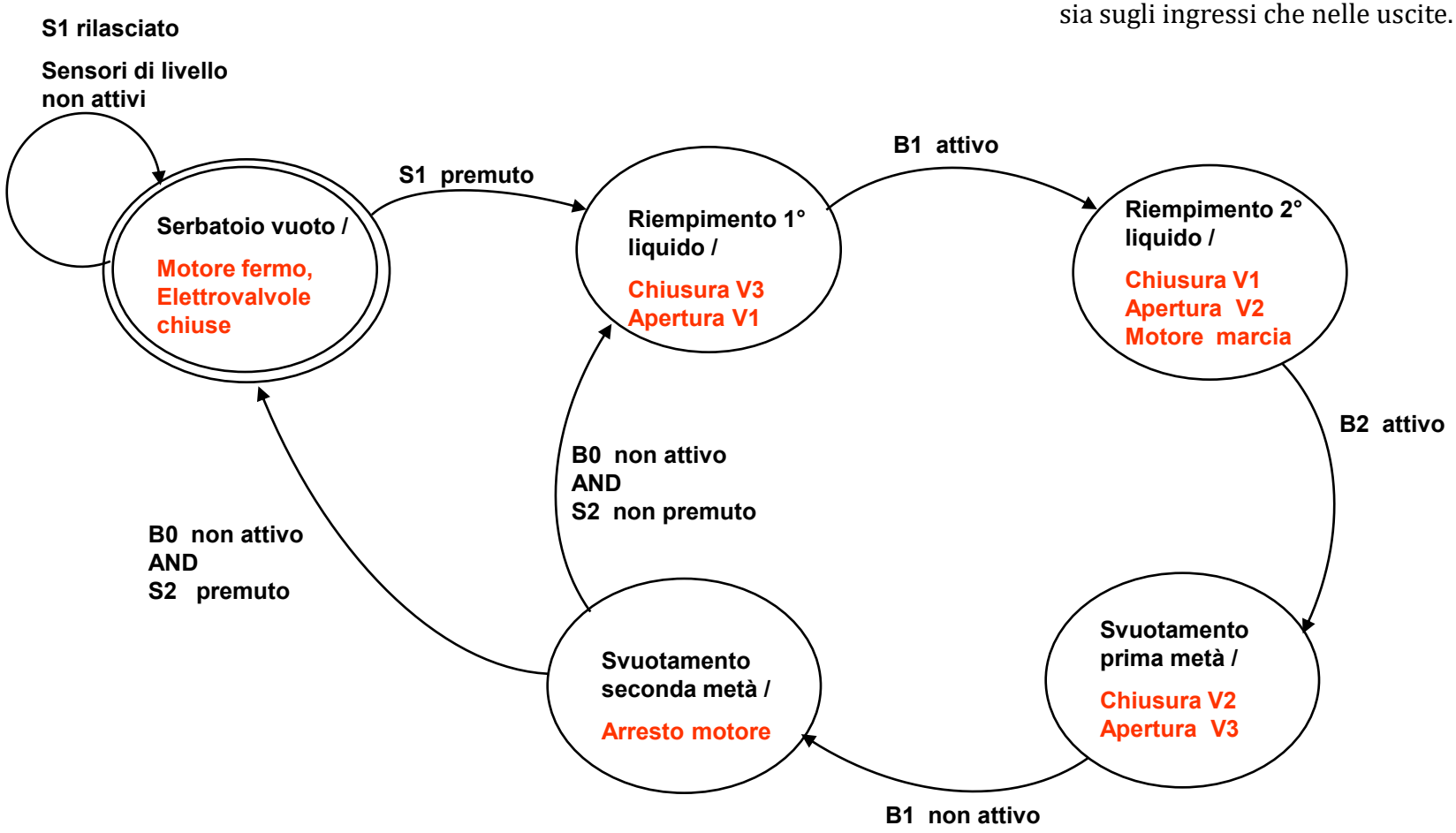
Si deve prevedere uno ***stato di attesa***, da cui uscire con l'attivazione del pulsante S1.



Soluzione: **modello di Moore**

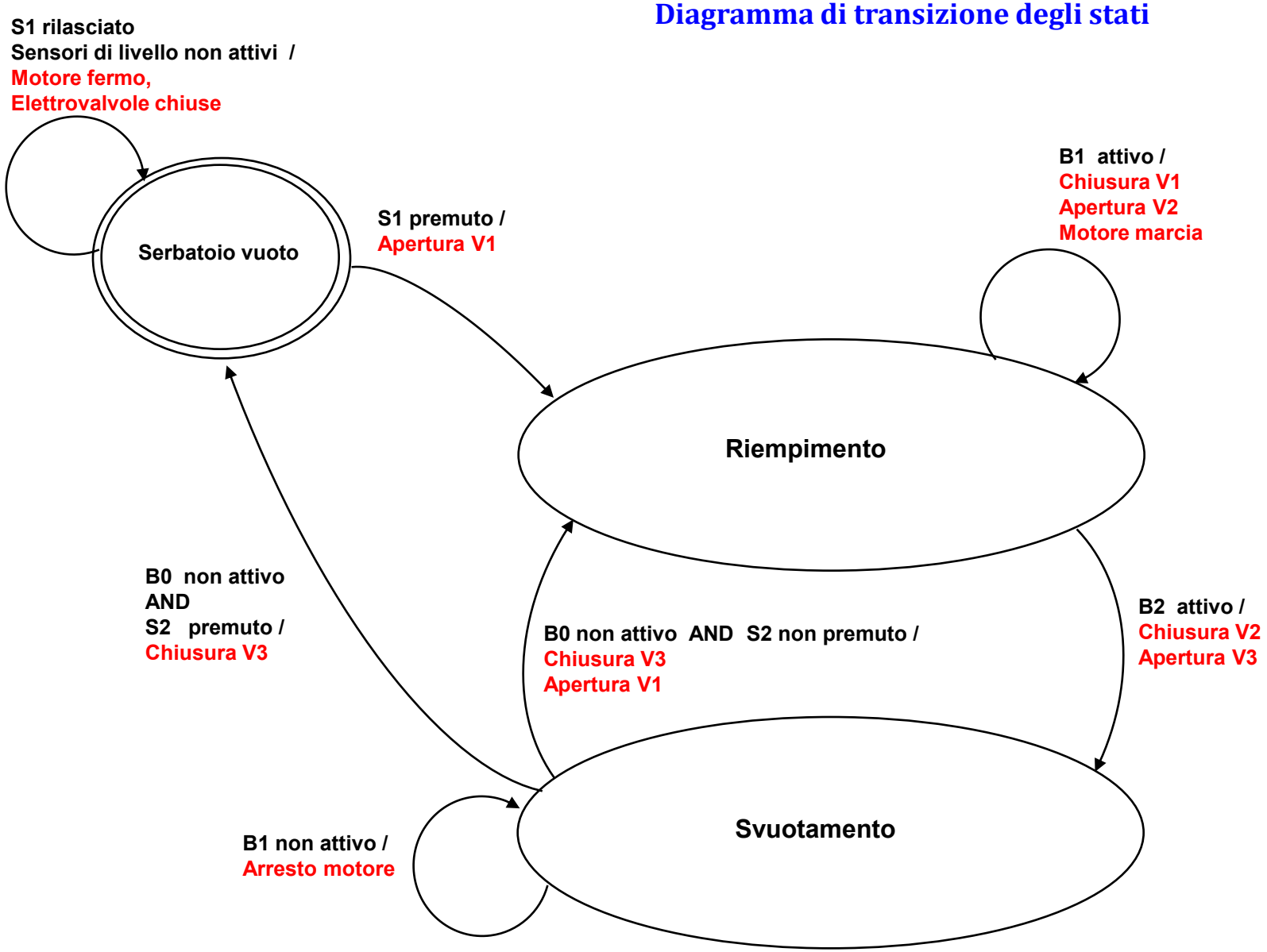
Diagramma di transizione degli stati

NB: nel diagramma sono indicate solo le variazioni che intervengono sia sugli ingressi che nelle uscite.



Soluzione: **modello di Mealy**

Diagramma di transizione degli stati



Implementazione SOFTWARE dei sistemi di controllo

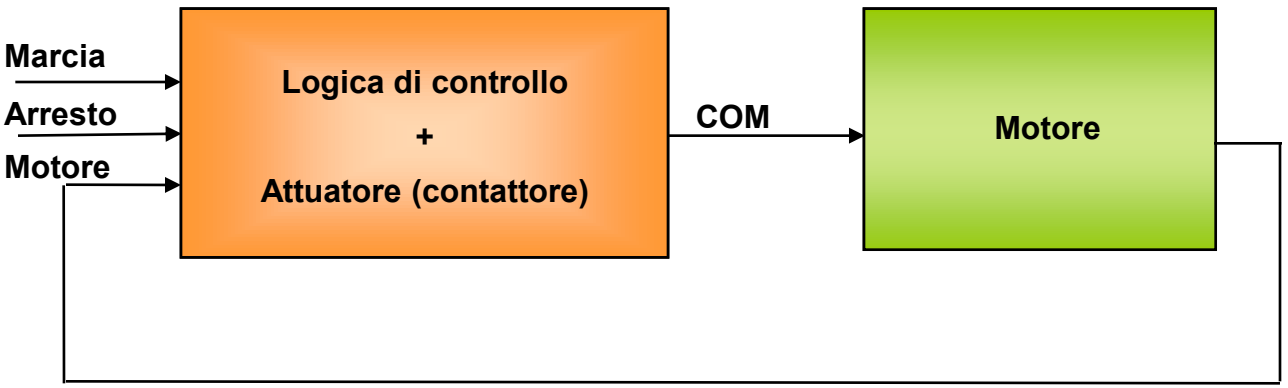
Il sistema di controllo, descritto mediante *diagramma di transizione degli stati* o mediante *tabelle di transizione*, può essere implementato (realizzato) con

- ***hardware cablato*** per la specifica applicazione
- ***hardware programmabile***, scrivendo uno specifico software di controllo.

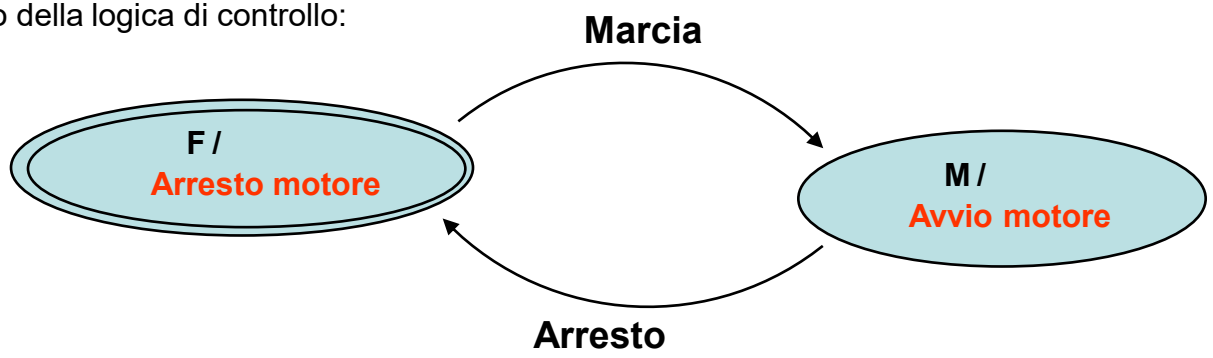
In questo corso si approfondirà lo studio della scrittura del **software di controllo**.

Il linguaggio di programmazione è il C ANSI, con compilatore Dev-C++.

ESEMPIO **Comando di marcia/arresto di un motore**



Funzionamento della logica di controllo:



Il **software di controllo** presenta fondamentalmente due sezioni:

- sezione di inizializzazione: in cui scrivere il codice da eseguire una sola volta, all'inizio
- sezione sotto scansione ciclica: in cui si scrive il codice di controllo, la cui esecuzione è ripetuta ciclicamente.

Esempio di software di controllo:

```
1  /*  CONTROLLO MARCIA-ARRESTO DI UN MOTORE
2
3      Modello di MOORE  -  sistema ASINCRONO
4
5     Indirizzi delle linee di ingresso e uscita:
6              marcia:   ingresso digitale 1
7              arresto:   ingresso digitale 2
8              FineControllo:   ingresso digitale 3
9          NB: comando motore:   uscita digitale 1          */
10
11 #include <stdio.h>
12 #include "LibreriaK8055.c"
13
14
15 int marcia, arresto, FineControllo;
16 char stato;
17 void connessione();         /*  Connessione alla scheda USB K8055 Velleman */
18 void inizializzazione();     /*  Stato e uscita iniziali  */
19
20 int main()
21  {connessione();
22   inizializzazione();
23    while(!FineControllo)
24       {/* Lettura ingressi */
25        marcia = ReadDigitalChannel(1);
26        arresto = ReadDigitalChannel(2);
27        /* Aggiornamento dello stato e delle uscite  */
28        switch(stato)
29             {case 'F': ClearDigitalChannel(1);     /* Uscita associata allo stato 'F' */
30                        if(marcia == 1)             /* Aggiornamento dello stato */
31                              {stato = 'M'; }
32                        break;
```

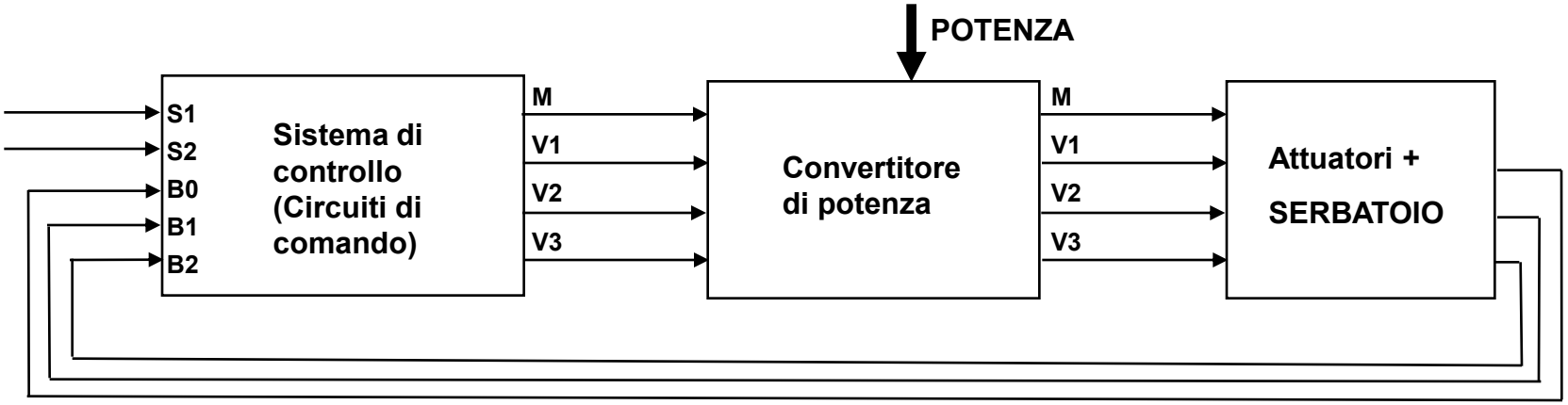
Sezione di inizializzazione

Continua ...

```
33              case 'M': SetDigitalChannel(1);
34                        if(arresto == 1)
35                              {stato = 'F';}
36               }
37          FineControllo = ReadDigitalChannel(3);
38          }
39     CloseDevice(0);
40     }
41
42 /*************************************************************************/
43 void connessione()
44     {int h;
45      avvio();
46      h = OpenDevice(0);
47      if(h == 0)
48          {printf("Scheda 0 connessa");
49           printf("\n\n"); }
50        else
51          {printf("Scheda 0 non connessa");
52           printf("\n\n"); }
53      }
54
55 /*************************************************************************/
56 void inizializzazione()      /*  Stato e uscita iniziali  */
57     {stato = 'F';              /*  Stato: motore fermo  */
58      ClearDigitalChannel(1);     /*  Uscita: linea di comando del motore  */
59      }
60 /*************************************************************************/
61
```

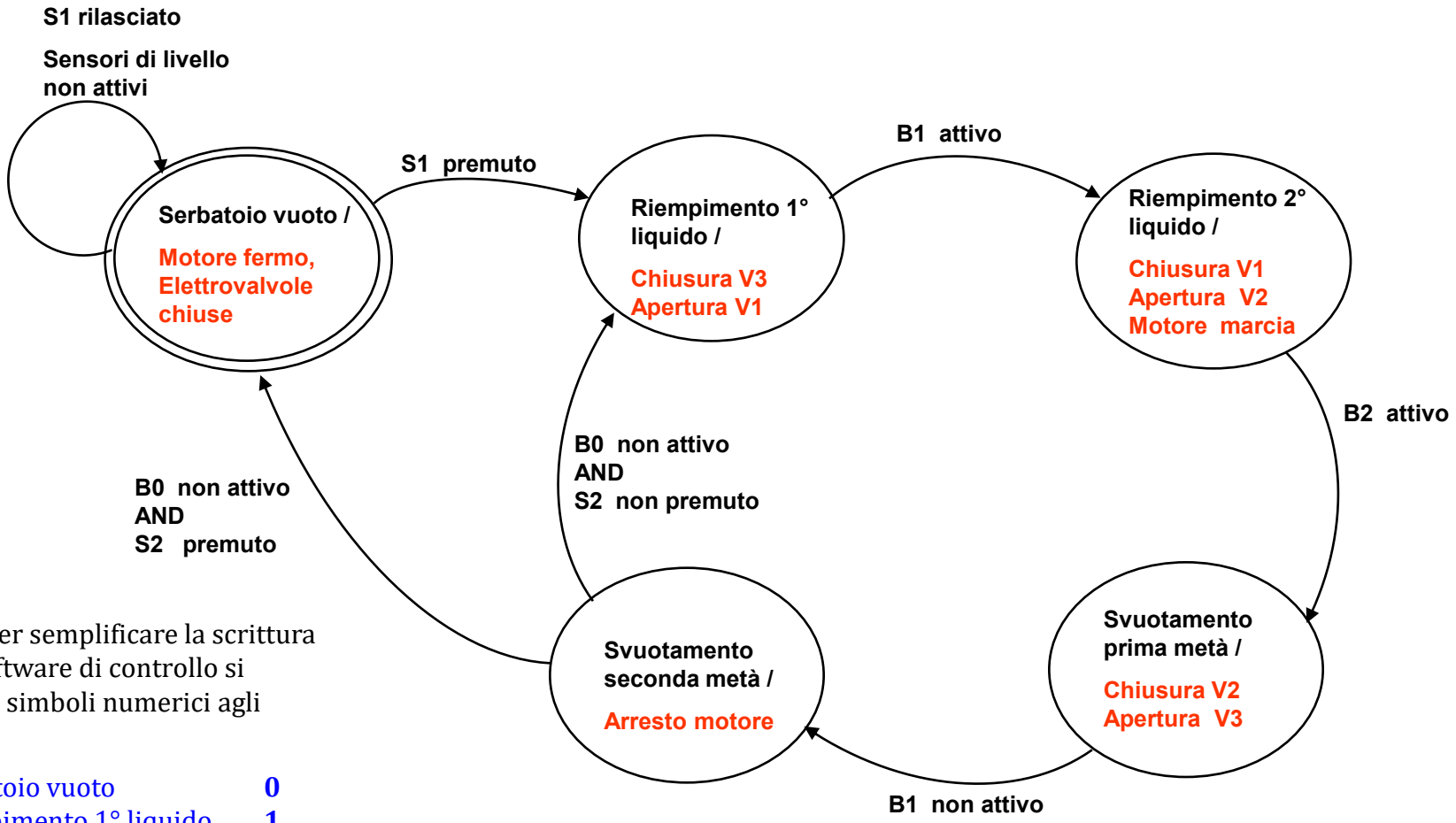
ESEMPIO **Miscelatore** *(descritto in precedenza)*

Scrivere il software di controllo.



Soluzione : **modello di Moore**

Diagramma di transizione degli stati



NB: per semplificare la scrittura del software di controllo si danno simboli numerici agli stati:

Serbatoio vuoto	**0**
Riempimento 1° liquido	**1**
Riempimento 2° liquido	**2**
Svuotamento prima metà	**3**
Svuotamento seconda metà	**4**

NB: nel **software** i sensori **B0, B1, B2** sono rappresentati con le variabili **SL0, SL1** e **SL2**

Esempio di software di controllo:

```
/*  Controllo MISCELATORE

    Modello di MOORE  -  sistema ASINCRONO

   Indirizzi delle linee di ingresso e uscita:
            S1:    ingresso digitale 1
            SL0:   ingresso digitale 2
            SL1:   ingresso digitale 3
            SL2:   ingresso digitale 4
            S2:    ingresso digitale 5
            V1:       uscita digitale 1
            V2:       uscita digitale 2
            V3:       uscita digitale 3
            motore:   uscita digitale 4                    */

#include <stdio.h>
#include "LibreriaK8055.c"

int S1, SL0, SL1, SL2, S2;
int stato;
void connessione();         /*  Connessione alla scheda USB K8055 Velleman */
void inizializzazione();     /*  Stato e uscita iniziali  */

int main()
 { connessione();
   inizializzazione();
   while(!stop)
      {/* Lettura ingressi */
       S1 = ReadDigitalChannel(1);
       SL0 = ReadDigitalChannel(2);
       SL1 = ReadDigitalChannel(3);
       SL2 = ReadDigitalChannel(4);
       S2 = ReadDigitalChannel(5);
```

Continua ...

Segue ...

```
34          /* Aggiornamento dello stato e delle uscite  */
35          switch(stato)
36               {case 0: if(S1)
37                          {stato = 1;}
38                        break;
39                case 1: ClearDigitalChannel(3);     /*  chiusura V3  */
40                        SetDigitalChannel(1);       /*  Apertura V1  */
41                        if(SL1)
42                          {stato = 2;}
43                        break;
44                case 2: ClearDigitalChannel(1);     /*  Chiusura V1  */
45                        SetDigitalChannel(2);       /*  Apertura V2  */
46                        SetDigitalChannel(4);       /*  Avvio motore  */
47                        if(SL2)
48                          {stato = 3;}
49                        break;
50                case 3: ClearDigitalChannel(2);     /*  Chiusura V2  */
51                        SetDigitalChannel(3);       /*  Apertura V3  */
52                        if(!SL1)
53                          {stato = 4;}
54                        break;
55                case 4: ClearDigitalChannel(4);     /*  Arresto motore  */
56                        if(!SL0 && !S2)
57                                {stato = 1;}        /*  ripartenza del ciclo */
58                        if(!SL0 && S2)
59                                {stato = 0;}        /*  stato di attesa  */
60                 }
61           S2 = ReadDigitalChannel(5);
62          }
63     CloseDevice(0);
64     }
```

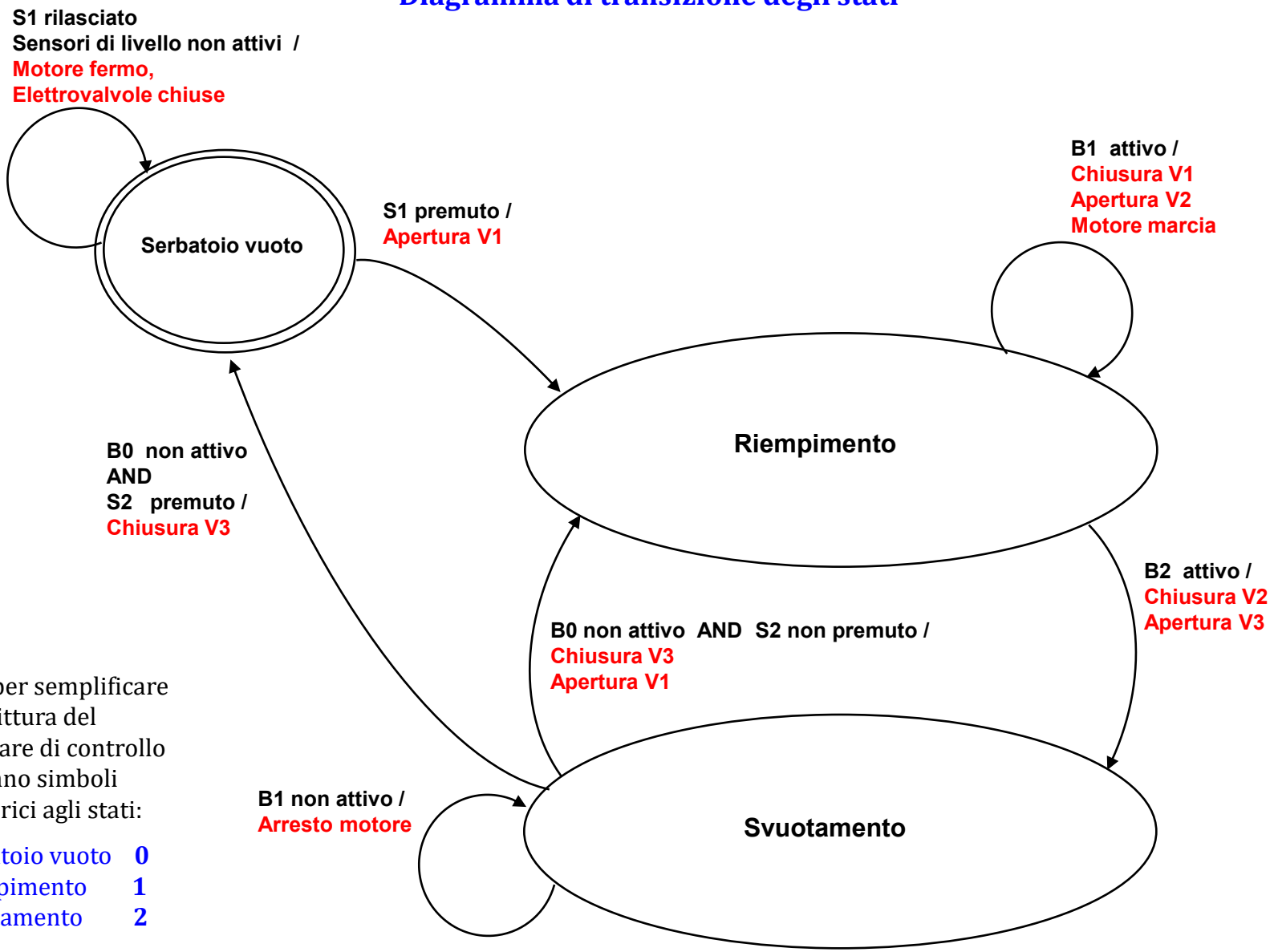
Continua ...

Segue ...

```
66  /*************************************************************************/
67  void connessione()
68     {int h;
69      avvio();
70      h = OpenDevice(0);
71      if(h == 0)
72          {printf("Scheda 0 connessa");
73           printf("\n\n"); }
74        else
75          {printf("Scheda 0 non connessa");
76           printf("\n\n"); }
77      }
78
79  /*************************************************************************/
80  void inizializzazione()      /*  Stato e uscita iniziali  */
81     {stato = 0;               /*  Serbatoio vuoto, elettrovalvole chiuse, motore fermo  */
82      SL0 = ReadDigitalChannel(2);
83      if(SL0)
84         printf("Attenzione!! \nLiquido nel serbatoio. Provvedere allo svuotamento.");
85      }
86  /*************************************************************************/
```

Soluzione : **modello di Mealy**

Diagramma di transizione degli stati



NB: per semplificare la scrittura del software di controllo si danno simboli numerici agli stati:

Serbatoio vuoto	**0**
Riempimento	**1**
Svuotamento	**2**

Esempio di software di controllo:

```
/*  Controllo MISCELATORE

    Modello di MEALY  -  sistema ASINCRONO

   Indirizzi delle linee di ingresso e uscita:
            S1:    ingresso digitale 1
            SL0:   ingresso digitale 2
            SL1:   ingresso digitale 3
            SL2:   ingresso digitale 4
            S2:    ingresso digitale 5
            V1:       uscita digitale 1
            V2:       uscita digitale 2
            V3:       uscita digitale 3
            motore:   uscita digitale 4                    */

#include <stdio.h>
#include "LibreriaK8055.c"

int S1, SL0, SL1, SL2, S2;
int stato;
void connessione();         /*  Connessione alla scheda USB K8055 Velleman */
void inizializzazione();     /*  Stato e uscita iniziali  */

int main()
 { connessione();
   inizializzazione();
   while(!stop)
      {/* Lettura ingressi */
       S1 = ReadDigitalChannel(1);
       SL0 = ReadDigitalChannel(2);
       SL1 = ReadDigitalChannel(3);
       SL2 = ReadDigitalChannel(4);
       S2 = ReadDigitalChannel(5);
```

Continua ...

Segue ...

```
34          /* Aggiornamento dello stato e delle uscite  */
35          switch(stato)
36              {case 0: if(S1)
37                         {SetDigitalChannel(1);      /*  Apertura V1  */
38                          stato = 1;}
39                       break;
40               case 1: if(SL1)
41                         {ClearDigitalChannel(1);    /*  Chiusura V1  */
42                          SetDigitalChannel(2);      /*  Apertura V2  */
43                          SetDigitalChannel(4); }    /*  Avvio motore  */
44                       if(SL2)
45                         {ClearDigitalChannel(2);    /*  Chiusura V2  */
46                          SetDigitalChannel(3);      /*  Apertura V3  */
47                          stato = 2;}
48                       break;
49               case 2: if(!SL1)
50                         {ClearDigitalChannel(4); }   /*  Arresto motore  */
51                       if(!SL0 && !S2)
52                         {ClearDigitalChannel(3);    /*  Chiusura V3  */
53                          SetDigitalChannel(1);      /*  Apertura V1  */
54                          stato = 1;}                /*  Ripartenza del ciclo  */
55                       if(!SL0 && S2)
56                         {ClearDigitalChannel(3);    /*  Chiusura V3  */
57                          stato = 0;}                /*  stato di attesa  */
58                 }
59           S2 = ReadDigitalChannel(5);
60          }
61      CloseDevice(0);
62      }
```

Continua ...

Segue ...

```
64  /*************************************************************************/
65  void connessione()
66     {int h;
67      avvio();
68      h = OpenDevice(0);
69      if(h == 0)
70          {printf("Scheda 0 connessa");
71           printf("\n\n"); }
72        else
73          {printf("Scheda 0 non connessa");
74           printf("\n\n"); }
75       }
76
77  /*************************************************************************/
78  void inizializzazione()      /*  Stato e uscita iniziali  */
79     {stato = 0;               /*  Serbatoio vuoto, elettrovalvole chiuse, motore fermo  */
80      SL0 = ReadDigitalChannel(2);
81      if(SL0)                          /* Controllo iniziale serbatoio vuoto */
82         printf("Attenzione!! \nLiquido nel serbatoio. Provvedere allo svuotamento.");
83       }
84  /*************************************************************************/
```