

SIMULAZIONE DI UN AUTOMA A STATI FINITI CON VISUAL BASIC

1. Introduzione

L'automata preso in considerazione è quello che gestisce un distributore di bibite, già descritto come esempio nel documento "Automati a stati finiti", dal quale sono state estratte e riportate più avanti le matrici di Transizione di stato e di Trasformazione di uscita. Questo lavoro rappresenta un passo intermedio verso la realizzazione di un progetto esecutivo perché, trattandosi di una simulazione, in esso manca completamente l'interfacciamento con il mondo esterno (gettoniera e dispositivi di sgancio); il suo scopo quindi è solo quello di mettere a punto la logica di funzionamento dell'automata. Inoltre, poiché le tabelle di transizione di stato e di trasformazione di uscita vengono gestite in Visual Basic con delle matrici, i cui indici sono rappresentati dai valori, rispettivamente, dello "stato attuale" e dell'"ingresso", risulta necessario codificare tali valori con numeri interi appartenenti all'insieme naturale. Allo stesso modo, anche se per motivi diversi, è utile codificare le uscite.

2. Tabelle e matrici

		INGRESSO		
		0c	10c	20c
STATO ATTUALE	0	0	10b	20b
	10b	10	10b	10b
	10	10	20b	30b
	20b	20	20b	20b
	20	20	30b	40
	30b	30	30b	30b
	30	30	40	50
	40	0	40	40
	50	0	50	50

STATO FUTURO	USCITA
0	0/0c
10b	0/0c
10	0/0c
20b	0/0c
20	0/0c
30b	0/0c
30	0/0c
40	1/0c
50	1/10c

La codifica utilizzata sarà:

Stato	0	10b	10	20b	20	30b	30	40	50
Codice	0	1	2	3	4	5	6	7	8

Ingresso	0c	10c	20c
Codice	0	1	2

Uscita	0/0c	1/0c	1/10c
Codice	0	1	2

Da cui derivano le seguenti matrici:

TRANSIZIONE DI STATO					
		INGRESSO			
		(0c)	(10c)	(20c)	
		0	1	2	
STATO ATTUALE	(0)	0	0	1	3
	(10b)	1	2	1	1
	(10)	2	2	3	5
	(20b)	3	4	3	3
	(20)	4	4	5	7
	(30b)	5	6	5	5
	(30)	6	6	7	8
	(40)	7	0	7	7
	(50)	8	0	8	8

TRASFORMAZIONE DI USCITA			
STATO FUTURO		USCITA	
(0)	0	(0/0)	0
(10b)	1	(0/0)	0
(10)	2	(0/0)	0
(20b)	3	(0/0)	0
(20)	4	(0/0)	0
(30b)	5	(0/0)	0
(30)	6	(0/0)	0
(40)	7	1/0c	1
(50)	8	1/10c	2

Nota: In parentesi sono indicati i valori originali di "Stato", "Ingresso" e "Uscita".

3. Programma in Visual Basic

La form (Fig. 1) prevede due Shape a forma di LED rossi che evidenziano i comandi per lo sgancio, rispettivamente, della bibita e dell'eventuale resto. L'introduzione delle monete è simulato con due bottoni, uno per la moneta da 10c e l'altro per la moneta da 20c. Due textbox indicano, rispettivamente, le monete introdotte e quelle ancora da introdurre per raggiungere il costo della bibita. Il bottone "Fine" chiude l'applicazione.

ITIS E. BARSANTI – POMIGLIANO D'ARCO
CORSO DI SISTEMI ELETTRONICI AUTOMATICI – Prof. Paolo Bisconti

DISTRIBUTORE DI BIBITE

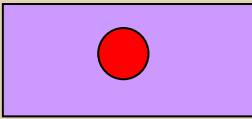
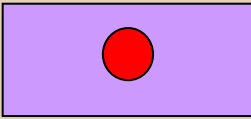
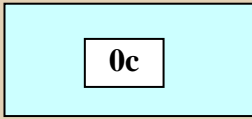
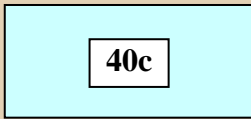



Bibita (Costo 40c)	Resto (10c)	Monete introdotte	Monete da introdurre
			
			
			

Fig. 1 - Form

Il codice è articolato intorno a due istruzioni fondamentali che ad ogni evoluzione del sistema prelevano dalle matrici lo stato futuro [Stato = M(Stato,Ingresso)] e, con questo, l'uscita [Uscita = V(Stato)]. Lo stato futuro viene conservato in una variabile globale (Stato) per essere utilizzato nella successiva evoluzione come stato attuale, mentre l'uscita pilota i due LED rossi (nella realtà comanda i dispositivi di sgancio della bibita e dell'eventuale resto). Il processo si ripete ad ogni evoluzione; ovviamente ogni evoluzione è attivata dalla pressione di uno dei due bottoni che simulano l'introduzione delle monete.

I due LED vengono comandati mediante una struttura Select Case che, in base al valore dell'uscita (0, 1 o 2), cambia il colore delle shape.

Per simulare anche l'ingresso "0" (intervallo di tempo tra il passaggio nella gettoniera di due monete) vengono utilizzati il metodo "MouseDown" per attivare il valore dell'ingresso (1 o 2) ed il metodo "MouseUp" per simulare il ritorno allo stato di riposo dell'ingresso.

Poiché le operazioni da svolgere sono sempre le stesse, indipendentemente dall'ingresso attivato (0, 1 o 2), si è preferito utilizzare una Sub chiamata "Automa", alla quale occorre passare solo il valore dell'ingresso.

Per avere i valori numerici da utilizzare per generare le informazioni "Monete introdotte" e "Monete da introdurre" è stato utilizzato un vettore che associa ai valori codificati per lo stato i corrispondenti valori numerici in centesimi di Euro.

CODICE

```
Dim M(8, 2) As Byte 'Matrice di Transizione di Stato  
Dim V(8) As Byte 'Vettore di Trasformazione di Uscita  
Dim Conv(8) As Byte 'Vettore di conversione codici stato --> valori numerici stato  
Dim Stato As Byte 'Variabile di stato
```

```
Private Sub Form_Load()
```

```
'Viene caricata la Matrice di Transizione di Stato
```

```
M(0, 0) = 0
```

```
M(0, 1) = 1
```

```
M(0, 2) = 3
```

```
M(1, 0) = 2
```

```
M(1, 1) = 1
```

```
M(1, 2) = 1
```

```
M(2, 0) = 2
```

```
M(2, 1) = 3
```

```
M(2, 2) = 5
```

```
M(3, 0) = 4
```

```
M(3, 1) = 3
```

```
M(3, 2) = 3
```

```
M(4, 0) = 4
```

```
M(4, 1) = 5
```

```
M(4, 2) = 7
```

```
M(5, 0) = 6
```

```
M(5, 1) = 5
```

```
M(5, 2) = 5
```

```
M(6, 0) = 6
```

```
M(6, 1) = 7
```

```
M(6, 2) = 8
```

```
M(7, 0) = 0
```

```
M(7, 1) = 7
```

```
M(7, 2) = 7
```

```
M(8, 0) = 0
```

```
M(8, 1) = 8
```

```
M(8, 2) = 8
```

```
'Viene caricato il Vettore Trasformazione di Uscita
```

```
V(0) = 0
```

```
V(1) = 0
```

```
V(2) = 0
```

```
V(3) = 0
```

```
V(4) = 0
```

```
V(5) = 0
```

```
V(6) = 0
```

```
V(7) = 1
```

```
V(8) = 2
```

'Viene caricato il Vettore Conversione "codici stato" --> "valori numerici stato"

```
Conv(0) = 0  
Conv(1) = 10  
Conv(2) = 10  
Conv(3) = 20  
Conv(4) = 20  
Conv(5) = 30  
Conv(6) = 30  
Conv(7) = 40  
Conv(8) = 50
```

```
Stato = 0      'Viene inizializzato il valore dello stato  
Form1.Show    'Viene anticipata la visualizzazione del Form  
CmdFine.SetFocus 'Viene spostato il Focus sul bottone di Fine  
End Sub
```

'Attiva l'ingresso 1 (10c)

```
Private Sub Cmd10_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    Call Automa(1)  
End Sub
```

'Attiva il ritorno a 0 per la moneta da 10c

```
Private Sub Cmd10_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    Call Automa(0)  
End Sub
```

'Attiva l'ingresso 2 (20c)

```
Private Sub Cmd20_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    Call Automa(2)  
End Sub
```

'Attiva il ritorno a 0 per la moneta da 20c

```
Private Sub Cmd20_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    Call Automa(0)  
End Sub
```

'Chiude l'applicazione

```
Private Sub CmdFine_Click()  
    End  
End Sub
```

```
Private Sub Automa(ByVal Ingresso As Byte)
```

```
'Determina i valori dello stato futuro e della corrispondente uscita
```

```
Stato = M(Stato, Ingresso)
```

```
Uscita = V(Stato)
```

```
'Colora i LED a seconda del valore dell'uscita
```

```
Select Case Uscita
```

```
Case 0
```

```
ShpBibita.FillColor = &HC0C0C0
```

```
ShpResto.FillColor = &HC0C0C0
```

```
Case 1
```

```
ShpBibita.FillColor = vbRed
```

```
ShpResto.FillColor = &HC0C0C0
```

```
Case 2
```

```
ShpBibita.FillColor = vbRed
```

```
ShpResto.FillColor = vbRed
```

```
End Select
```

```
'Calcola e pubblica il valore delle monete introdotte e quello delle monete da introdurre
```

```
Monete = Conv(Stato)
```

```
Ancora = 40 - Monete
```

```
If Ancora < 0 Then Ancora = 0
```

```
TxtMonete.Text = Str(Monete) + "c"
```

```
TxtAncora.Text = Str(Ancora) + "c"
```

```
End Sub
```

PERFEZIONAMENTO

Aggiungendo una .dll che consenta a Visual Basic di accedere alla porta parallela, è possibile migliorare la simulazione utilizzando un minilab; con esso, e con l'aggiunta di qualche istruzione di I/O, si possono generare i segnali di ingresso mediante degli switches e si possono visualizzare i comandi in uscita mediante dei LED.