

# Distributore automatico

Progetto di Tecnologie Informatiche per l'Automazione Industriale - A.A. 2019/20  
**prof. Mele Adriano**

Capasso Carmine

N46002108

Scala Simona

N46003295

<b>Panoramica</b>	<b>2</b>
Il problema di controllo	2
Le specifiche richieste	2
<b>Dispositivo di controllo PLC</b>	<b>3</b>
Implementazione	3
MAIN	4
Tipo PRODOTTO	4
INIZIALIZZAZIONE	5
GETTONIERA	8
SCELTA_PRODOTTO	9
EROGA_PRODOTTO	11
LETTURA_CARTA	13
SENSORI_RIFORNIMENTO	13

## Panoramica

### Il problema di controllo

Si vuole progettare un distributore automatico per la distribuzione di cibi e bevande freschi.

La macchina deve permettere all'utente di effettuare il pagamento in monete, banconote o carta di credito. All'erogazione del prodotto il distributore deve dare resto, se presente. La scelta di un prodotto viene effettuata tramite un tastierino numerico, digitando il numero del prodotto desiderato.

Si richiede, inoltre, che la macchina sia in grado di segnalare alla centrale quando è necessario un rifornimento e di segnalare eventuali malfunzionamenti.

### Le specifiche richieste

Per ovviare al problema di controllo posto in esame si richiede, innanzitutto, che il distributore sia dotato dei seguenti sistemi di pagamento:

- Lettore di monete (0,50€, 1€, 2€)
- Lettore di banconote (5€, 10€)
- POS self service con tecnologia contactless

I prodotti saranno disposti secondo lo schema di una matrice 4x4.

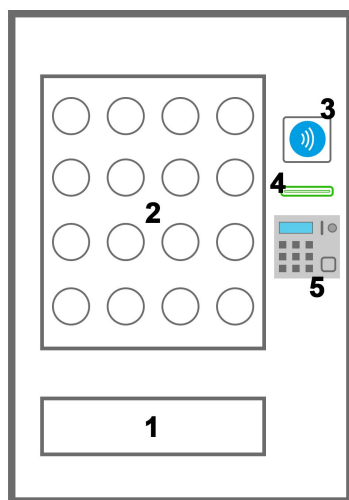
Ogni "canalina" di prodotti sarà dotata di una spirale messa in movimento da un motore all'atto dell'erogazione di un prodotto. Inoltre, ogni canalina sarà dotata di un sensore in grado di rilevare la presenza o meno di prodotti al suo interno. Nel momento in cui una riga ed una colonna di prodotti risulteranno vuote, il sistema avvertirà la centrale per un rifornimento.

All'atto dell'erogazione del prodotto, il vano per il ritiro del prodotto si sbloccherà per un tempo pari a 30 secondi per poi bloccarsi nuovamente.

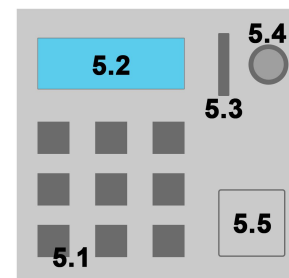
Se l'utente dovesse rinunciare all'acquisto dopo aver inserito il denaro (monete o banconote) e prima di aver digitato la propria scelta sul tastierino numerico, può riavere indietro il credito inserito mediante un pulsante apposito.

Nel caso di acquisto con carta di credito, la procedura avverrà come segue: lettura della carta mediante contactless, scelta del prodotto, erogazione del prodotto, addebito su carta di credito. Dunque l'utente riceverà l'addebito sulla propria carta contestualmente all'erogazione del prodotto.

Si presenta, di seguito, il modello del distributore automatico da progettare.



1. Vano per ritiro prodotto
2. Matrice 4x4 di prodotti
3. POS self service con tecnologia contactless
4. Lettore di banconote
5. Dettaglio



- 5.1. Tastierino numerico
- 5.2. Display
- 5.3. Lettore di monete
- 5.4. Pulsante reset
- 5.5. Vano ritiro resto

## Dispositivo di controllo PLC

È stato affidato il controllo logico ad dispositivo di controllo PLC, implementato mediante l'utilizzo del software open-source [OpenPLC](#).

- 0A 0B 0C 0D

1A 1B 1C 1D

2A 2B 2C 2D

3A 3B 3C 3D

La gestione dei prodotti all'interno del distributore è stata pensata secondo uno schema ordinato da numeri (da 0 a 4) e lettere (da A a D), raffigurato accanto.
- In questo modo è stato possibile identificare univocamente ogni prodotto, rendendo al contempo più semplice la gestione della logica del controllore.

Si dà per scontato che i numeri di ogni prodotto siano ben visibili all'utente da una targhetta, posta in corrispondenza di ogni canalina contenente i prodotti stessi.

## Implementazione

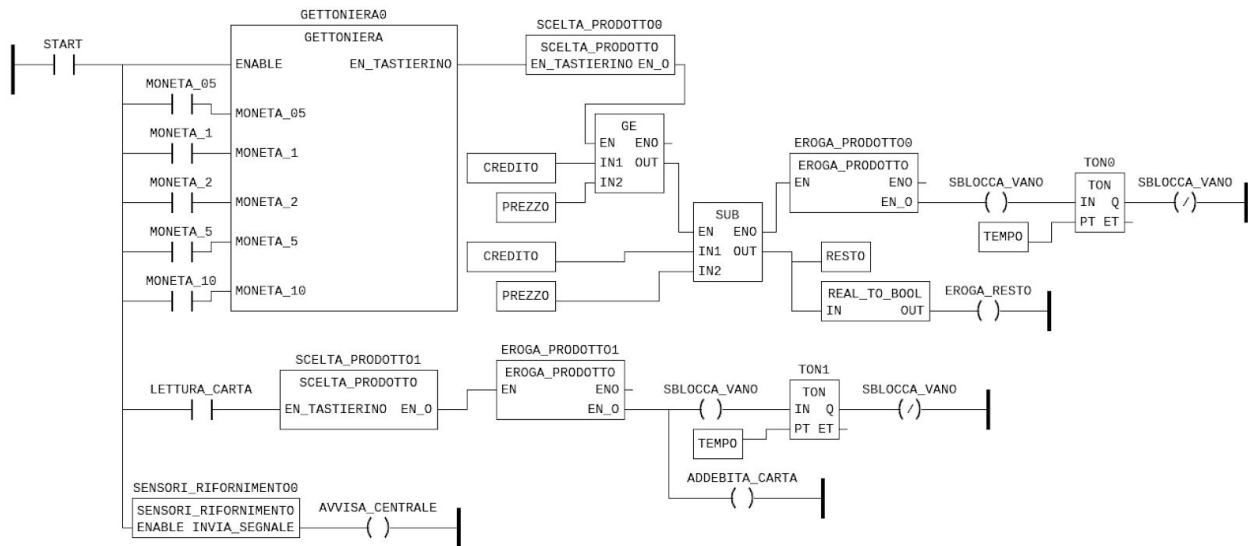
L'implementazione è stata realizzata adoperando **Ladder Diagram (LD)** come linguaggio grafico principale e **Structured Text (ST)** come linguaggio testuale d'ausilio in alcuni blocchi funzionali.

All'interno del progetto sono presenti due programmi: *MAIN* e *INIZIALIZZAZIONE*.

Il primo è responsabile del funzionamento stesso del distributore, ne contiene infatti tutta la logica. Il secondo si occupa di inizializzare tutti i prodotti del distributore, dichiarati mediante l'uso di un tipo '*PRODOTTO*' personalizzato composto da tre campi: *MOTORE*, che identifica il motore che aziona la spirale della canalina corrispondente; *PREZZO*, ossia il prezzo del prodotto; *SENSORE*, che identifica, infine, il sensore posto all'inizio di ogni

canalina e responsabile di rilevare la presenza o meno di prodotti all'interno di quella canalina (e.g. sensore fotoelettrico). In questo caso si suppone che il distributore sia tutto pieno, dunque tutti i sensori sono stati inizializzati a 1 (oggetto rilevato).

## MAIN



## Tipo *PRODOTTO*

**PRODOTTO**

Derivation Type:

Type infos:

Elements :

+ - ↑ ↓

#	Name	Type	Initial Value
1	MOTORE	BOOL	
2	PREZZO	REAL	
3	SENSORE	BOOL	

## INIZIALIZZAZIONE

```

1 IF SINGLE_EX THEN
2     PRODOTTO_0A.MOTORE:=0;
3     PRODOTTO_0A.PREZZO:=0.70;
4     PRODOTTO_0A.SENSORE:=1;
5
6     PRODOTTO_0B.MOTORE:=0;
7     PRODOTTO_0B.PREZZO:=0.80;
8     PRODOTTO_0B.SENSORE:=1;
9
10    PRODOTTO_0C.MOTORE:=0;
11    PRODOTTO_0C.PREZZO:=0.80;
12    PRODOTTO_0C.SENSORE:=1;
13
14    PRODOTTO_0D.MOTORE:=0;
15    PRODOTTO_0D.PREZZO:=0.90;
16    PRODOTTO_0D.SENSORE:=1;
17
18    PRODOTTO_1A.MOTORE:=0;
19    PRODOTTO_1A.PREZZO:=1.0;
20    PRODOTTO_1A.SENSORE:=1;
21
22    PRODOTTO_1B.MOTORE:=0;
23    PRODOTTO_1B.PREZZO:=1.10;
24    PRODOTTO_1B.SENSORE:=1;
25
26    PRODOTTO_1C.MOTORE:=0;
27    PRODOTTO_1C.PREZZO:=1.10;
28    PRODOTTO_1C.SENSORE:=1;
29
30    PRODOTTO_1D.MOTORE:=0;
31    PRODOTTO_1D.PREZZO:=1.20;
32    PRODOTTO_1D.SENSORE:=1;
33
34    PRODOTTO_2A.MOTORE:=0;
35    PRODOTTO_2A.PREZZO:=1.50;
36    PRODOTTO_2A.SENSORE:=1;
37
38    PRODOTTO_2B.MOTORE:=0;
39    PRODOTTO_2B.PREZZO:=1.50;
40    PRODOTTO_2B.SENSORE:=1;
41
42    PRODOTTO_2C.MOTORE:=0;
43    PRODOTTO_2C.PREZZO:=1.80;
44    PRODOTTO_2C.SENSORE:=1;
45
46    PRODOTTO_2D.MOTORE:=0;
47    PRODOTTO_2D.PREZZO:=1.80;
48    PRODOTTO_2D.SENSORE:=1;
49
50    PRODOTTO_3A.MOTORE:=0;
51    PRODOTTO_3A.PREZZO:=0.80;
52    PRODOTTO_3A.SENSORE:=1;
53
54    PRODOTTO_3B.MOTORE:=0;
55    PRODOTTO_3B.PREZZO:=1.50;
56    PRODOTTO_3B.SENSORE:=1;
57
58    PRODOTTO_3C.MOTORE:=0;
59    PRODOTTO_3C.PREZZO:=0.50;
60    PRODOTTO_3C.SENSORE:=1;
61
62    PRODOTTO_3D.MOTORE:=0;
63    PRODOTTO_3D.PREZZO:=0.50;
64    PRODOTTO_3D.SENSORE:=1;
65
66    SINGLE_EX:=0;
67 END_IF;

```

La variabile *SINGLE\_EX* (Single Execution) è una variabile globale che viene utilizzata a livello di risorsa per controllare l'esecuzione singola del task *INIZIALIZZAZIONE*.

A livello di risorsa, infatti, il *MAIN* è stato istanziato di default come un task ciclico di periodo 20ms, mentre l'*INIZIALIZZAZIONE* è stata prevista come task singolo, come già detto.

37	SINGLE_EX	Global	BOOL		1	
----	-----------	--------	------	--	---	--

Tasks:				
Name	Triggering	Single	Interval	Priority
task0	Cyclic		T#20ms	1
task1	Interrupt	SINGLE_EX		0

Instances:		
Name	Type	Task
instance0	MAIN	task0
instance1	INIZIALIZZAZIONE	task1

Come priorità è stata assegnata la più alta (0) al *task1* istanziato come *INIZIALIZZAZIONE* e riconosciuto come primo task da eseguire poiché necessario ad inizializzare le variabili utili al funzionamento del distributore stesso.

Sono state dichiarate, inoltre, le seguenti variabili globali:

Config0.Res0							
Class Filter: All							
#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	CREDITO	Global	REAL				Il credito inserito dall'utente
2	NUMERO	Global	STRING				Il numero del prodotto selezionato
3	PREZZO	Global	REAL				Il prezzo del prodotto selezionato
4	RESTO	Global	REAL				Il resto da erogare
5	PRODOTTO_0A	Global	PRODOTTO				
6	PRODOTTO_0B	Global	PRODOTTO				
7	PRODOTTO_0C	Global	PRODOTTO				
8	PRODOTTO_0D	Global	PRODOTTO				
9	PRODOTTO_1A	Global	PRODOTTO				
10	PRODOTTO_1B	Global	PRODOTTO				
11	PRODOTTO_1C	Global	PRODOTTO				
12	PRODOTTO_1D	Global	PRODOTTO				
13	PRODOTTO_2A	Global	PRODOTTO				
14	PRODOTTO_2B	Global	PRODOTTO				
15	PRODOTTO_2C	Global	PRODOTTO				
16	PRODOTTO_2D	Global	PRODOTTO				
17	PRODOTTO_3A	Global	PRODOTTO				
18	PRODOTTO_3B	Global	PRODOTTO				
19	PRODOTTO_3C	Global	PRODOTTO				
20	PRODOTTO_3D	Global	PRODOTTO				
21	S_0A	Global	BOOL				
22	S_0B	Global	BOOL				
23	S_0C	Global	BOOL				
24	S_0D	Global	BOOL				
25	S_1A	Global	BOOL				
26	S_1B	Global	BOOL				
27	S_1C	Global	BOOL				
28	S_1D	Global	BOOL				
29	S_2A	Global	BOOL				
30	S_2B	Global	BOOL				
31	S_2C	Global	BOOL				
32	S_2D	Global	BOOL				
33	S_3A	Global	BOOL				
34	S_3B	Global	BOOL				
35	S_3C	Global	BOOL				
36	S_3D	Global	BOOL				
37	SINGLE_EX	Global	BOOL		1		Variabile utilizzata per l'esecuzione

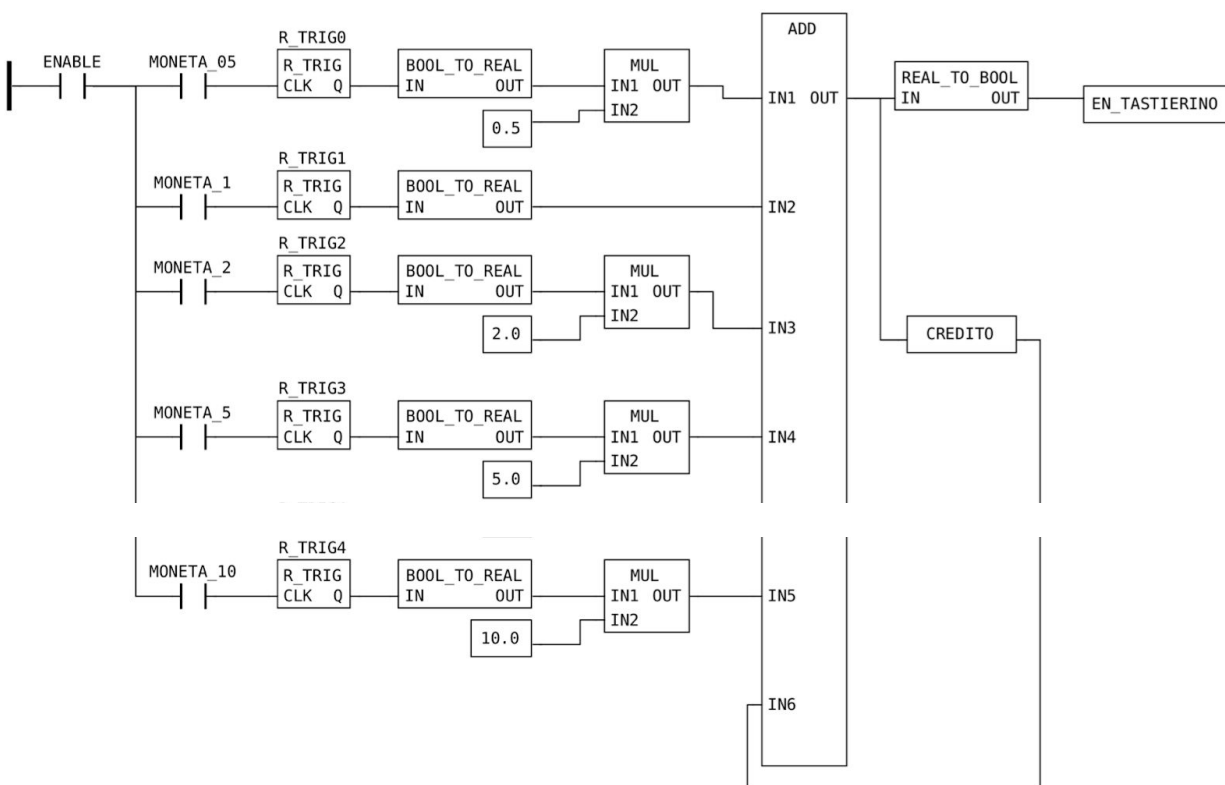
Di cui *CREDITO* tiene conto del credito inserito dall'utente, *NUMERO* e *PREZZO* tengono conto del numero e del prezzo del prodotto scelto dall'utente e infine *RESTO* tiene conto del resto da erogare.

Tutti gli altri sono i *PRODOTTI* presenti all'interno del distributore. Le variabili *S\_XX* sono state utilizzate ai fini del corretto funzionamento del blocco funzionale *SENSORI\_RIFERIMENTO* (a causa di un bug in OpenPLC non è possibile utilizzare le variabili *PRODOTTO\_XX.SENSORE* esplicitamente all'interno dei costrutti di controllo).



## GETTONIERA

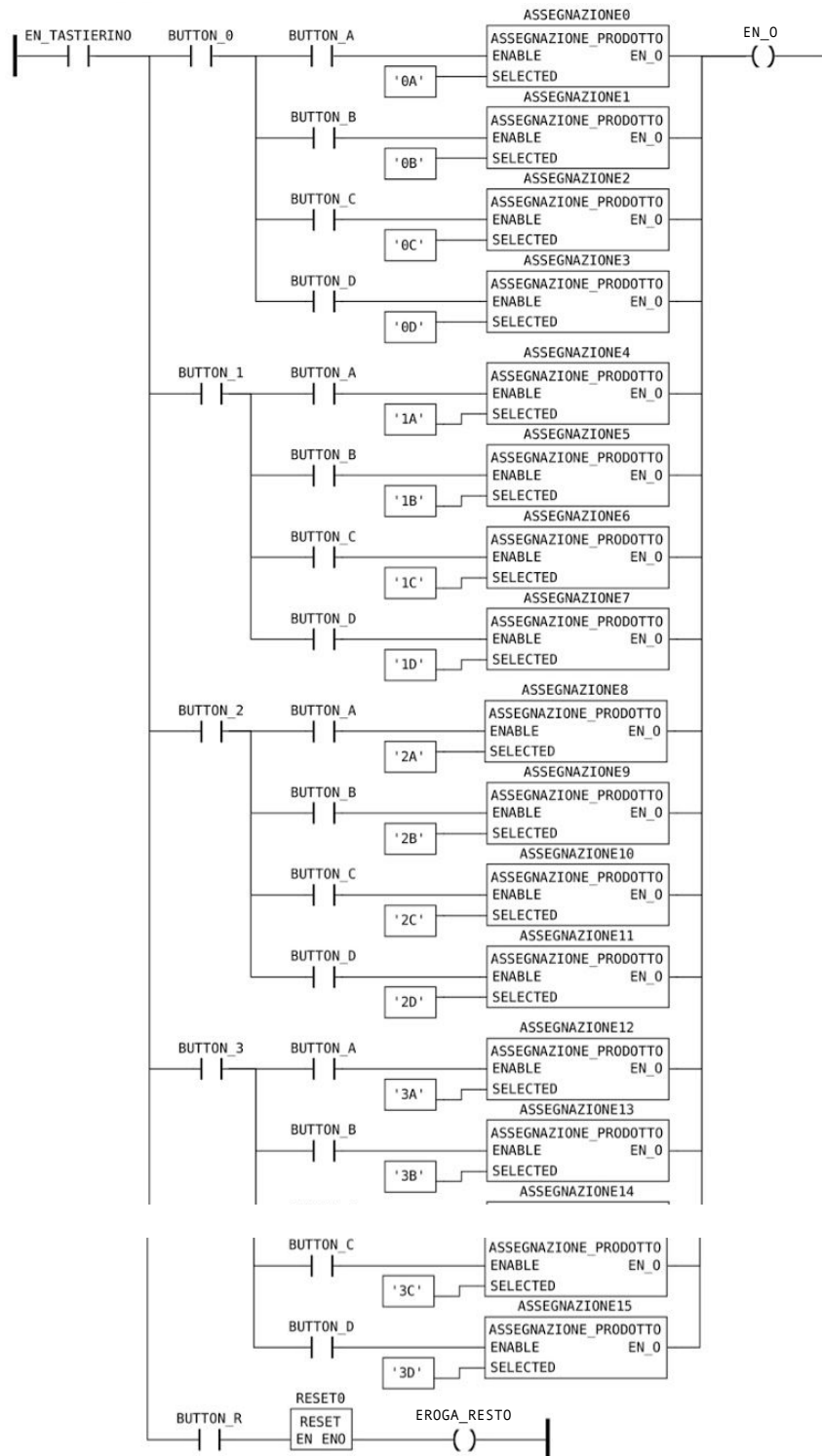
Il primo blocco funzionale che si incontra all'interno del *MAIN* è la *GETTONIERA* che permette l'inserimento di monete (0,50 euro, 1 euro, 2 euro) e banconote (5 euro, 10 euro) all'interno del distributore.



Il blocco *R\_TRIG* abilita la conversione e la successiva moltiplicazione al valore effettivo della moneta solo sul fronte di salita della moneta stessa, evitando conversioni e moltiplicazioni a "loop infinito".

Non appena viene inserita una moneta o una banconota, il blocco funzionale abilita il tastierino per la scelta del prodotto.

## SCELTA\_PRODOTTO



La variabile *BUTTON\_R* permette all'utente di riavere indietro il credito inserito nel caso in cui non si voglia più acquistare alcun prodotto.

**RESET**

Description:  Class Filter:

#	Name	Class	Type	Initial Value	Option	Documentation
1	CREDITO	External	REAL			
2	RESTO	External	REAL			
3	ENABLE	Input	BOOL			
4	EN_O	Output	BOOL			

```

1 IF ENABLE THEN
2   RESTO:=CREDITO;
3   CREDITO:=0.0;
4   EN_O:=1;
5 END_IF;
  
```

Una volta premuto il suddetto pulsante, il credito inserito fino a quel momento viene assegnato a *RESTO* e successivamente erogato ed il credito si azzerà.

Tornando alla scelta del prodotto, una volta effettuata la digitazione di numero e lettera sul tastierino, questi ultimi si assegnano alla variabile globale *NUMERO* ed al corrispettivo *PREZZO*.

Il blocco *ASSEGNAZIONE* è sempre lo stesso, chiamato più volte con un parametro di ingresso diverso a seconda del prodotto scelto.

**ASSEGNAZIONE\_PRODOTTI**

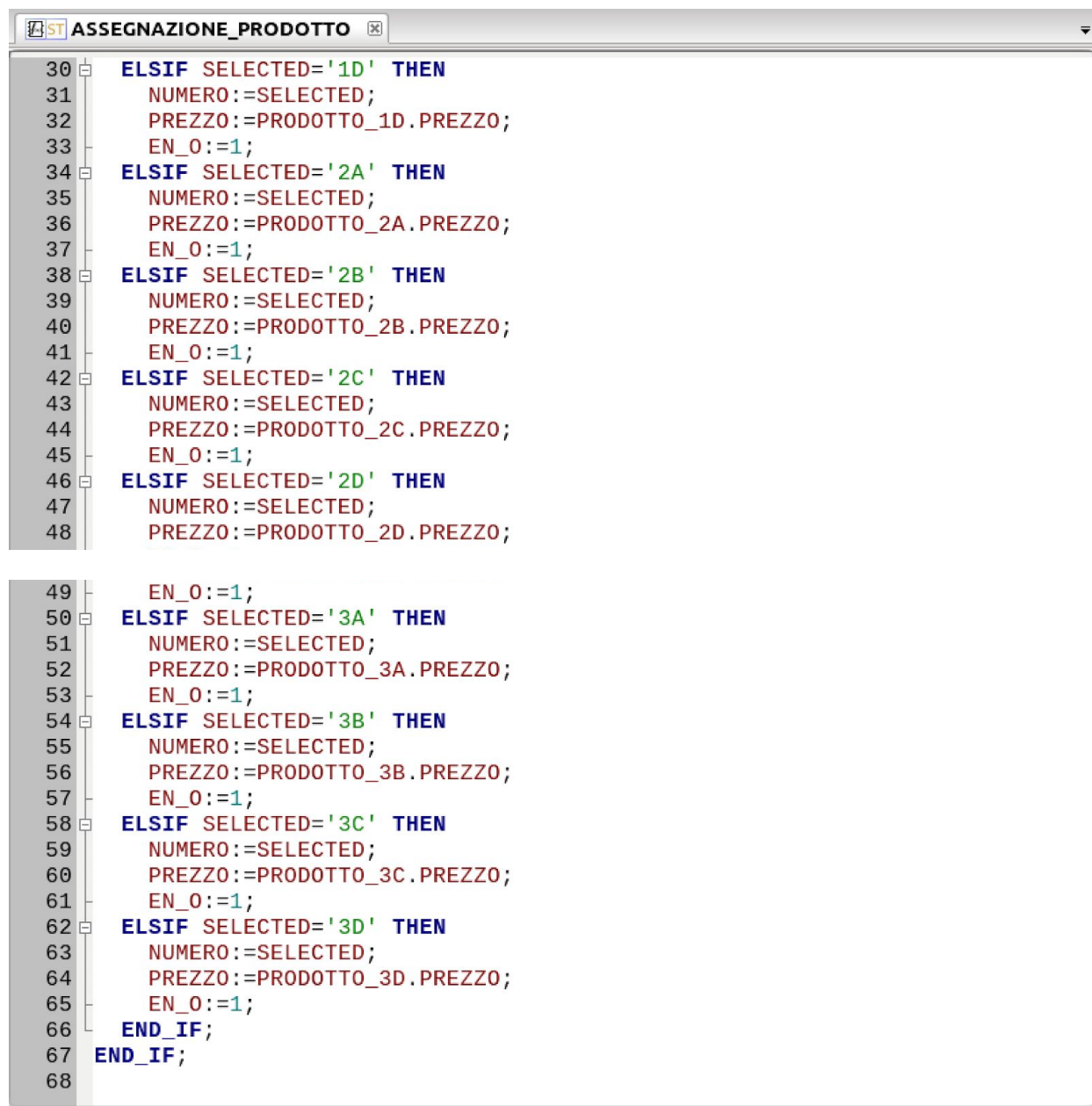
```

1 IF ENABLE THEN
2   IF SELECTED='0A' THEN
3     NUMERO:=SELECTED;
4     PREZZO:=PRODOTTO_0A.PREZZO;
5     EN_O:=1;
6   ELSIF SELECTED='0B' THEN
7     NUMERO:=SELECTED;
8     PREZZO:=PRODOTTO_0B.PREZZO;
9     EN_O:=1;
10  ELSIF SELECTED='0C' THEN
11    NUMERO:=SELECTED;
12    PREZZO:=PRODOTTO_0C.PREZZO;
13    EN_O:=1;
14  ELSIF SELECTED='0D' THEN
15    NUMERO:=SELECTED;
16    PREZZO:=PRODOTTO_0D.PREZZO;
17    EN_O:=1;
18  ELSIF SELECTED='1A' THEN
19    NUMERO:=SELECTED;
20    PREZZO:=PRODOTTO_1A.PREZZO;
21    EN_O:=1;
  
```

```

22  ELSIF SELECTED='1B' THEN
23      NUMERO:=SELECTED;
24      PREZZO:=PRODOTTO_1B.PREZZO;
25      EN_0:=1;
26  ELSIF SELECTED='1C' THEN
27      NUMERO:=SELECTED;
28      PREZZO:=PRODOTTO_1C.PREZZO;
29      EN_0:=1;

```



```

ASSEGNAZIONE_PRODOTTO
30  ELSIF SELECTED='1D' THEN
31      NUMERO:=SELECTED;
32      PREZZO:=PRODOTTO_1D.PREZZO;
33      EN_0:=1;
34  ELSIF SELECTED='2A' THEN
35      NUMERO:=SELECTED;
36      PREZZO:=PRODOTTO_2A.PREZZO;
37      EN_0:=1;
38  ELSIF SELECTED='2B' THEN
39      NUMERO:=SELECTED;
40      PREZZO:=PRODOTTO_2B.PREZZO;
41      EN_0:=1;
42  ELSIF SELECTED='2C' THEN
43      NUMERO:=SELECTED;
44      PREZZO:=PRODOTTO_2C.PREZZO;
45      EN_0:=1;
46  ELSIF SELECTED='2D' THEN
47      NUMERO:=SELECTED;
48      PREZZO:=PRODOTTO_2D.PREZZO;
49      EN_0:=1;
50  ELSIF SELECTED='3A' THEN
51      NUMERO:=SELECTED;
52      PREZZO:=PRODOTTO_3A.PREZZO;
53      EN_0:=1;
54  ELSIF SELECTED='3B' THEN
55      NUMERO:=SELECTED;
56      PREZZO:=PRODOTTO_3B.PREZZO;
57      EN_0:=1;
58  ELSIF SELECTED='3C' THEN
59      NUMERO:=SELECTED;
60      PREZZO:=PRODOTTO_3C.PREZZO;
61      EN_0:=1;
62  ELSIF SELECTED='3D' THEN
63      NUMERO:=SELECTED;
64      PREZZO:=PRODOTTO_3D.PREZZO;
65      EN_0:=1;
66  END_IF;
67 END_IF;
68

```

### EROGA\_PRODOTTO

Questo blocco funzionale viene attivato dopo aver effettuato il controllo sul credito inserito dall'utente e il prezzo del prodotto scelto e aver fatto la differenza tra i due.

```

ST EROGA_PRODOTTO
1 IF NUMERO='0A' THEN
2   PRODOTTO_0A.MOTORE:=1;
3   EN_0:=1;
4 ELSIF NUMERO='0B' THEN
5   PRODOTTO_0B.MOTORE:=1;
6   EN_0:=1;
7 ELSIF NUMERO='0C' THEN
8   PRODOTTO_0C.MOTORE:=1;
9   EN_0:=1;
10 ELSIF NUMERO='0D' THEN
11   PRODOTTO_0D.MOTORE:=1;
12   EN_0:=1;
13 ELSIF NUMERO='1A' THEN
14   PRODOTTO_1A.MOTORE:=1;
15   EN_0:=1;
16 ELSIF NUMERO='1B' THEN
17   PRODOTTO_1B.MOTORE:=1;
18   EN_0:=1;
19 ELSIF NUMERO='1C' THEN
20   PRODOTTO_1C.MOTORE:=1;
21   EN_0:=1;
22 ELSIF NUMERO='1D' THEN
23   PRODOTTO_1D.MOTORE:=1;
24   EN_0:=1;
25 ELSIF NUMERO='2A' THEN
26   PRODOTTO_2A.MOTORE:=1;
27   EN_0:=1;
28 ELSIF NUMERO='2B' THEN
29   PRODOTTO_2B.MOTORE:=1;
30   EN_0:=1;

```

```

ST EROGA_PRODOTTO
22 ELSIF NUMERO='1D' THEN
23   PRODOTTO_1D.MOTORE:=1;
24   EN_0:=1;
25 ELSIF NUMERO='2A' THEN
26   PRODOTTO_2A.MOTORE:=1;
27   EN_0:=1;
28 ELSIF NUMERO='2B' THEN
29   PRODOTTO_2B.MOTORE:=1;
30   EN_0:=1;
31 ELSIF NUMERO='2C' THEN
32   PRODOTTO_2C.MOTORE:=1;
33   EN_0:=1;
34 ELSIF NUMERO='2D' THEN
35   PRODOTTO_2D.MOTORE:=1;
36   EN_0:=1;
37 ELSIF NUMERO='3A' THEN
38   PRODOTTO_3A.MOTORE:=1;
39   EN_0:=1;
40 ELSIF NUMERO='3B' THEN
41   PRODOTTO_3B.MOTORE:=1;
42   EN_0:=1;
43 ELSIF NUMERO='3C' THEN
44   PRODOTTO_3C.MOTORE:=1;

```

```

45   EN_0:=1;
46   ELSIF NUMERO='3D' THEN
47     PRODOTTO_3D.MOTORE:=1;
48     EN_0:=1;
49   END_IF;
50

```

A seconda del prodotto scelto, viene azionato il motore corrispondente alla canalina di quel prodotto.

Una volta erogato il prodotto, l'utente ha 30 secondi per ritirare il prodotto dal vano di ritiro che si bloccherà nuovamente allo scadere del tempo.

### LETTURA\_CARTA

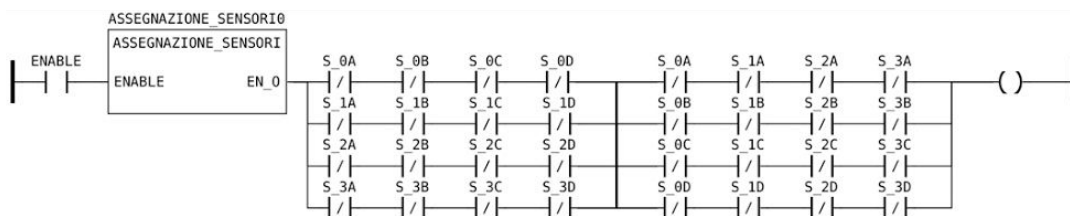
Come da specifica, l'utente ha anche la possibilità di effettuare il proprio acquisto mediante carta di credito.

In questo caso, dopo la corretta lettura della carta di credito, l'utente sarà abilitato all'uso del tastierino per la scelta del prodotto. All'atto dell'erogazione viene addebitato l'importo sulla carta di credito, in questo modo ci si accerta che l'addebito avvenga solo ed esclusivamente nel caso in cui il prodotto è stato effettivamente erogato.

### SENSORI\_RIFORNIMENTO

Contestualmente al segnale di *START* (accensione) del distributore, viene attivato questo function block responsabile di comunicare alla quando è necessario un rifornimento.

In particolare è stato deciso di rendere necessario un rifornimento qualora dovessero risultare vuote quattro canaline di una stessa colonna e quattro canaline di una stessa riga (un totale di 8/16 prodotti mancanti).



È stato necessario adoperare un function block *ASSEGNAZIONE\_SENSORI* affinché si superasse il problema dell'accesso ai campi di una variabile strutturata all'interno dei costrutti di controllo (bug di OpenPLC).

```

1   IF ENABLE THEN
2     S_0A:=PRODOTTO_0A.SENSORE;
3     S_0B:=PRODOTTO_0B.SENSORE;
4     S_0C:=PRODOTTO_0C.SENSORE;

```

```
5  S_0D:=PRODOTTO_0D.SENSORE;  
6  S_1A:=PRODOTTO_1A.SENSORE;  
7  S_1B:=PRODOTTO_1B.SENSORE;  
8  S_1C:=PRODOTTO_1C.SENSORE;  
9  S_1D:=PRODOTTO_1D.SENSORE;  
10 S_2A:=PRODOTTO_2A.SENSORE;  
11 S_2B:=PRODOTTO_2B.SENSORE;  
12 S_2C:=PRODOTTO_2C.SENSORE;  
13 S_2D:=PRODOTTO_2D.SENSORE;  
14 S_3A:=PRODOTTO_3A.SENSORE;  
15 S_3B:=PRODOTTO_3B.SENSORE;  
16 S_3C:=PRODOTTO_3C.SENSORE;  
17 S_3D:=PRODOTTO_3D.SENSORE;  
18  
19     EN_0:=1;  
20 END_IF;
```