



## **Gestione ricette terminale Weintek**

**ELSIST S.r.l.**  
**Sistemi in elettronica**

*Via G. Brodolini, 15 (Z.I.)*  
**15033 CASALE M.TO**  
**ITALY**

**Internet: <http://www.elsist.it>**  
**Email: [elsist@elsist.it](mailto:elsist@elsist.it)**

**TEL. (39)-0142-451987**  
**FAX (39)-0142-451988**

### **INDICE**

1Programma SlimLine.....	2
2Gestione ricette CON INDICE.....	3
2.1Macro.....	4
3Gestione ricette con oggetto ricetta.....	5

# 1 PROGRAMMA SLIMLINE

Il programma sullo SlimLine è in linguaggio ladder ed utilizza il blocco funzione ONOFFCYCLE per gestire il lampeggio delle due uscite logiche sul modulo CPU. E' stata definita una struttura dati per i dati di ricetta.

	Name	Pos.	Type	Array	Init value	Description
1	Name	0	STRING	[11]		Recipe name
2	Cy1Off	1	UINT	No		Cycle 1 off time (mS)
3	Cy1On	2	UINT	No		Cycle 1 on time (mS)
4	Cy2Off	3	UINT	No		Cycle 2 off time (mS)
5	Cy2On	4	UINT	No		Cycle 2 on time (mS)

Una variabile Rcp di tipo RECIPEDATA è stata allocata nella memoria DB100.2048 (Zona tampone), i dati di ricetta saranno trasferiti in questa variabile.

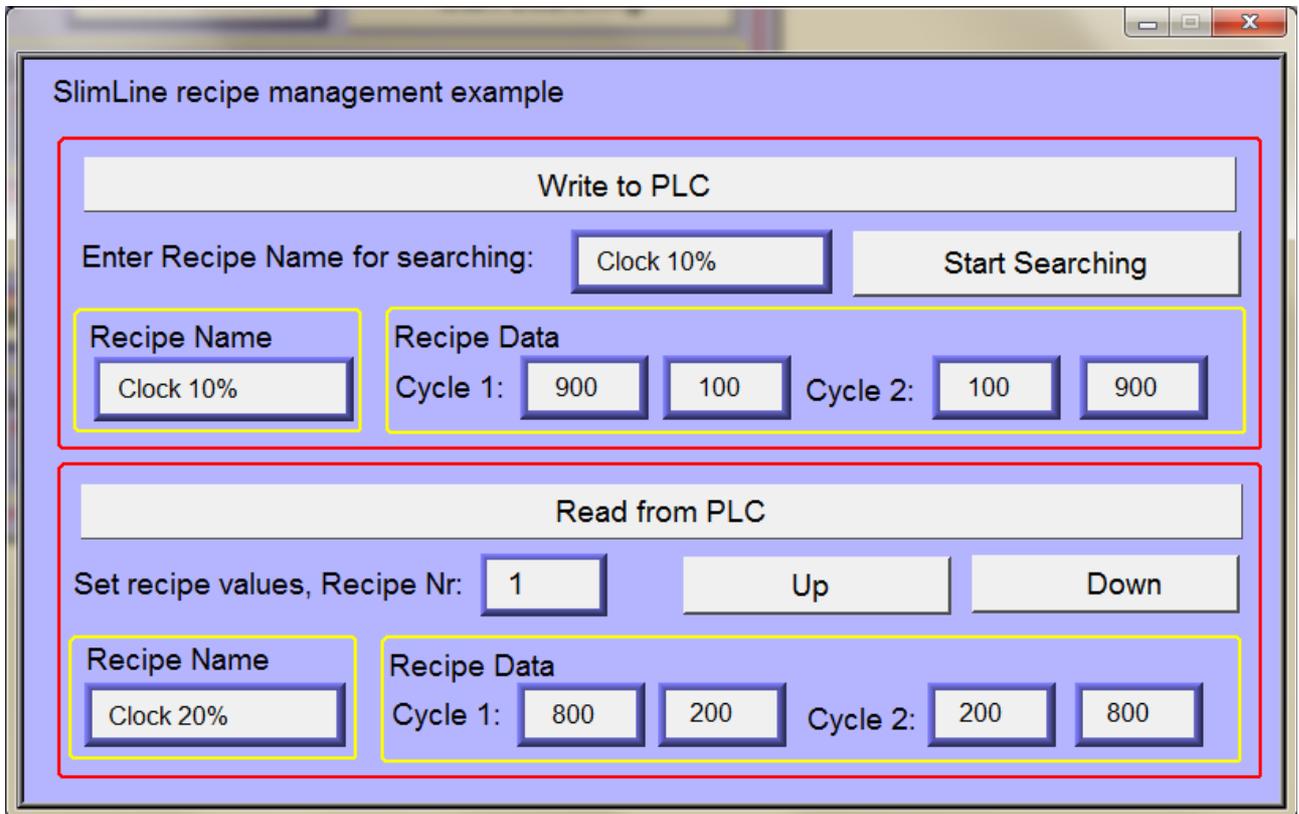
	Name	Type	Address	Group	Array	Init value	Attribute	Description
1	Do00CPU	BOOL	%QX255.0		No			Out 00 CPU module
2	Do01CPU	BOOL	%QX255.1		No			Out 01 CPU module
3	Rcp	RECIPEDATA	%MX100.2048		No			Recipe

Nel programma ladder si fa riferimento ai valori definiti nella ricetta.

The ladder logic shows two function blocks, Cycle1 (address 0001) and Cycle2 (address 0002), both of type ONOFFCYCLE. Each block has an 'Enable' input connected to a CPU output (Do00CPU for Cycle1, Do01CPU for Cycle2). The 'Delay' output of each block is also connected to a CPU output. The 'MinOffTime', 'MaxOffTime', 'MinOnTime', and 'MaxOnTime' inputs are connected to recipe variables: Rcp.Cy1Off, Rcp.Cy1On, Rcp.Cy2Off, and Rcp.Cy2On respectively.

## 2 GESTIONE RICETTE CON INDICE

Ho realizzato un semplice esempio di gestione ricette su di un pannello Weintek connesso ad un nostro modulo CPU della linea SlimLine. Ecco come si presenta il pannello con il progetto in esecuzione:



La parte inferiore è dedicata alla definizione delle ricette (Sono gestite 100 ricette), mentre la parte superiore è dedicata alla ricerca della ricetta nel ricettario ed al trasferimento dei valori in ricetta da e verso il PLC. E' previsto anche un pulsante per il salvataggio dei dati di ricetta letti dal PLC nel ricettario.

L'esempio prevede di gestire due uscite lampeggianti sul modulo CPU dello SlimLine con tempi di On e Off definibili tramite la ricetta. Le due uscite sono indicate come **Cycle 1** e **Cycle 2**, ed i due valori riportano il tempo di off e di on della relativa uscita espresso in mS.

### **Parte inferiore display**

Ogni ricetta è etichettata con un nome di 12 caratteri, con i tasti **Up** e **Down** è possibile spostarsi all'interno delle 100 ricette possibili visualizzandone e/o permettendo l'impostazione dei valori.

### **Parte superiore display**

E' possibile definire il nome della ricetta da ricercare, premendo il tasto **Start Searching** viene cercata la ricetta e se trovata i valori sono riportati nelle relative caselle. Il tasto **Write to PLC** trasferisce il valore di ricetta trovata sullo SlimLine. Il tasto **Read from PLC** legge i valori presenti nello SlimLine e li memorizza nella ricetta sostituendo quelli presenti.

## 2.1 MACRO

Per la ricerca della ricetta nel ricettario viene utilizzata una macro che è eseguita sulla pressione del tasto di ricerca.

### **Sorgente macro**

```
macro_command main()
char NFound[12]="Not found  "
short a[6],i,c[10]

// Eseguo acquisizione nome ricetta.
GetData(a[0],"Local HMI", LW, 100, 6)

// Eseguo loop di ricerca della ricetta nel ricettario.
// Ogni ricetta è lunga 10 word (6 per il nome e 4 per i valori).

for i = 0 to 1000 step 10

    // Trasferisco intera ricetta in variabile locale.
    GetData(c[0], "Local HMI", RW, i, 10)

    // Controllo nome ricetta se uguale a nome cercato. Se trovato
    // scrivo i valori nelle variabili di visualizzazione (Finestra superiore).
    if c[0]==a[0] and c[1]==a[1] and c[2]==a[2] and c[3]==a[3] and c[4]==a[4] and c[5]==a[5] then
        SetData(c[0], "Local HMI", LW, 0, 10)
        return
    end if
next i

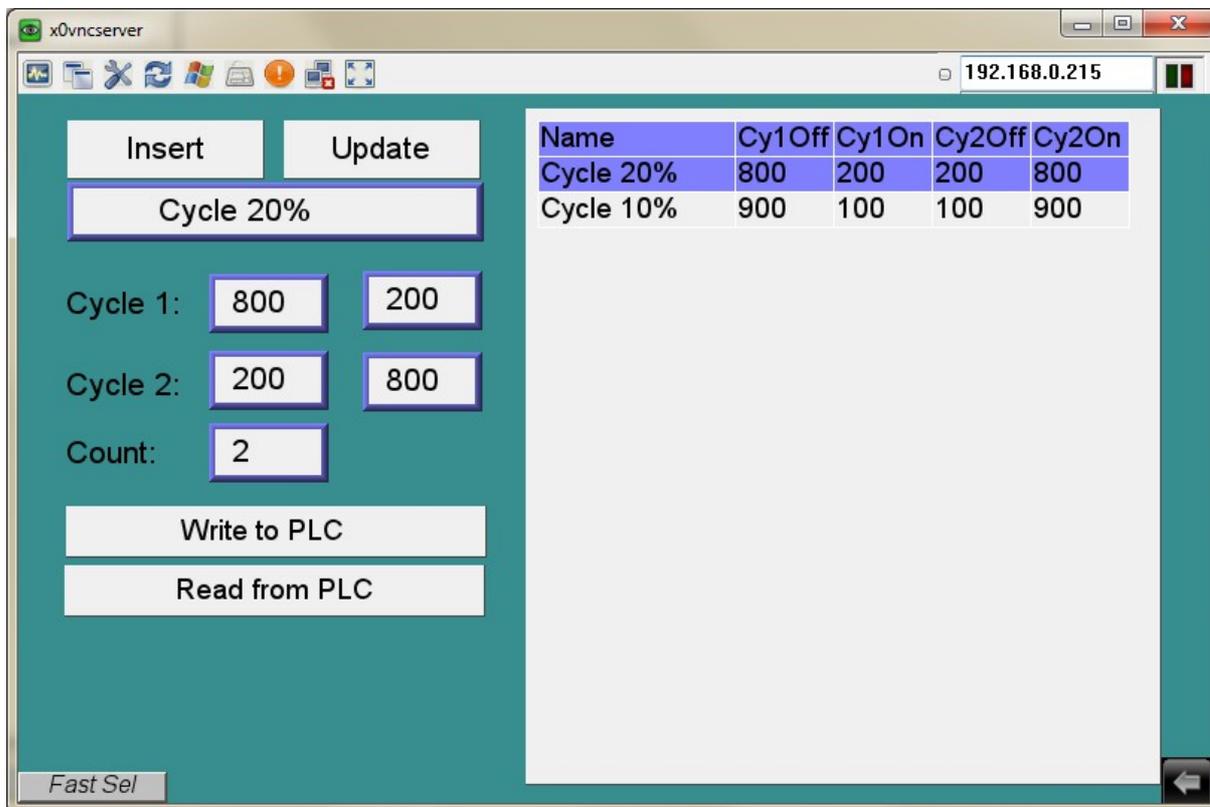
// Arrivo se non trovato ricetta abblenco risultato.

for i = 0 to 10
    c[i]=0
next i

SetData(c[0], "Local HMI", LW, 0, 10)
SetData(NFound[0], "Local HMI", LW, 0, 12)
end macro_command
```

### 3 GESTIONE RICETTE CON OGGETTO RICETTA

Nel programma EasyBuilder Pro è stato aggiunto l'oggetto RICETTA che facilita la gestione delle ricette ecco come si presenta il terminale.



Come si vede la ricetta viene visualizzata con l'oggetto RICETTA, il tasto **Insert** permette di inserire un nuovo record alla ricetta. Il tasto **Update** permette di aggiornare il record indirizzato della ricetta. Selezionando con un tocco il record di ricetta il record viene selezionato ed i dati trasferiti negli oggetti di visualizzazione.

Il comando di **Write to PLC** trasferisce i dati di ricetta nella struttura dello SlimLine. Il tasto Read from PLC trasferisce i dati dalla struttura ricetta del PLC nel terminale.