

Übung Wortverknüpfung „Heizen eines Ofens“

Lerneinheit

SPS-Programmbeispiel Heizen eine Ofens

Inhaltsübersicht

Bedienen von Programmier- und Steuergeräten
Erstellen von Symboltabelle, Funktionsplan
oder Anweisungsliste
Steuerungsprogramm eingeben, in Betrieb
nehmen und dokumentieren

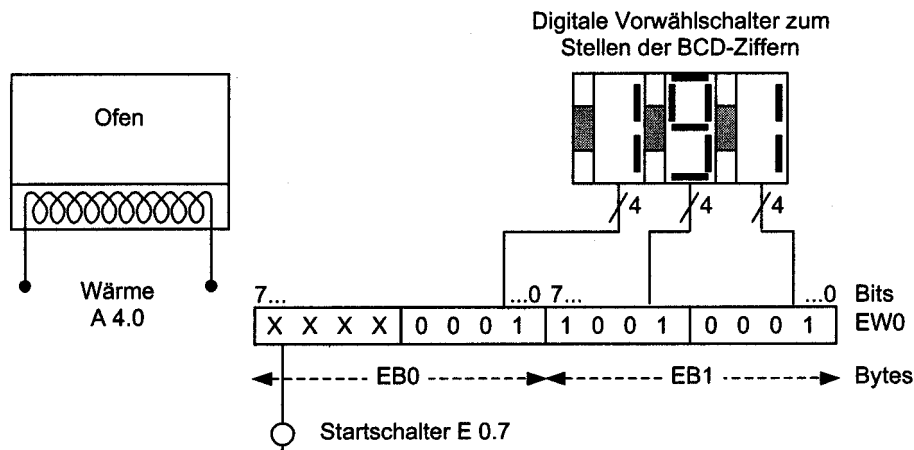
Übung Wortverknüpfung „Heizen eines Ofens“

Aufgabe

Heizen eines Ofens

Der Bediener startet das Heizen des Ofens, indem er den Startschalter drückt. Mit den digitalen Vorwählschaltern kann er die Dauer der Heizzeit festlegen. Der Wert, den er setzt, gibt die Sekunden im binär-codierten Dezimalformat (BCD) an.

Der Timer verlangt ein BCD code wenn ein größerer Wert wie 9 eingegeben wird geht die CPU in den Stop-Zustand.



Systemkomponente

Absolute Adresse

Startschalter	E 0.7
Digitale Vorwählschalter für Einer	E 1.0 bis E 1.3
Digitale Vorwählschalter für Zehner	E 1.4 bis E 1.7
Digitale Vorwählschalter für Hunderter	E 0.0 bis E 0.3
Beginn Heizvorgang	A 4.0

Ausmaskierung:

EW 0															
EB 0								EB 1							
0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0	1.7	1.6	1.5	1.4	1.3	1.2	1.1	1.0
0	1	1	1	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
UW															
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
EW 0															
z.B.: 44															
Hex 0FFF ausmaskieren															
nur wenn beide "1" dann steht im Ergebniss "1" d.h. 0.4 bis 0.7 kann nie eine "1" enthalten															

Timer fest auf Sekunden einstellen:

0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
AKKU1 z.B.: 44															
AKKU2 fest 2s															
OW															
0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
li. 4 Bit Zeitbasis 0, 1, 2 oder 3 wird fest auf 2 (Sek) gesetzt + Wert 44 = 44s															

Übung Wortverknüpfung „Heizen eines Ofens“

OB1

OB1 : "Main Program Sweep (Cycle)"

Kommentar:

☐ Netzwerk 1: Hilfsmerker immer low/high

CLR setzt das VKE auf den Signalzustand "0"
SET setzt das VKE auf den Signalzustand "1"

```
SET
=   "immer low"           MO.0           -- immer low
CLR
=   "immer high"          MO.1           -- immer high
```

☐ Netzwerk 2: Richtimpulsmerker

Nach dem STOP ==> RUN Übergang der CPU (OB100) wird der Richtimpulsmerker "Richtimpuls Anlauf" (M0.2) für 1 Sekunde gesetzt.

```
U    M    0.7
L    S5T#5S
SE   T    1
U    T    1
R    M    0.7
```

☐ Netzwerk 3: Zykluszeitmessung

Die mittlere Zykluszeit wird über den DB1.DBW4 in das MP übergeben

```
L    #OB1_MAX_CYCLE        #OB1_MAX_CYCLE    -- Maximum cycle time of OB1 (milliseconds)
T    MW    110
L    #OB1_PREV_CYCLE        #OB1_PREV_CYCLE    -- Cycle time of previous OB1 scan (milliseconds)
T    MW    112
L    #OB1_MIN_CYCLE        #OB1_MIN_CYCLE    -- Minimum cycle time of OB1 (milliseconds)
T    MW    114
```

☐ Netzwerk 4: Titel:

```
CALL "Anlage EIN/AUS"      FC1              -- Anlage EIN_AUS
```

OB100

OB100 : "Complete Restart"

Kommentar:

☐ Netzwerk 1: "Einschaltimpulsmerker"

```
SET
S    M    0.7
```

Übung Wortverknüpfung „Heizen eines Ofens“

FC1

FC1 : Allgemeine Funktionen

Kommentar:

Netzwerk 1: Titel:

```

U   T   1           //Richtimpuls nach Steuerung eine T10 AUS
R   T   10
                                     |

U   T   10          //Wenn die Zeit läuft wir das Programm in Zeile 3 beendet
=   A   4.0
BEB

L   EW   0           //lade EW0 nach AKKU 1 (Timer Wert in BCD also nur 0-9 pro Byte möglich)
UW   W#16#FFF        //ausmaskieren der Eingänge E0.4 - E0.7

//E0.6 und E0.7 im AKKU1 ist die Zeitkonstante für den Timer

OW   W#16#2000        //Zeitbasis im linken Byte (1,2,oder 3 sind möglich; 2 für Sek.) auch in AKKU 1 schreiben

//somit steht der Timer Wert komplett in AKKU 1 zur verfügung

U   E   0.7
SV   T   10

```

Ausführliche Beschreibung:

FC2 : Allgemeine Funktionen

Kommentar:

Netzwerk 1: Ausführliche Version

Kommentar:

```

U   E   3.0           //Timer rücksetzen
R   T   10

U   T   10           //Wenn die Zeit läuft wir das Programm in Zeile 3 beendet
=   A   4.0
BEB

L   EW   0           //lade EW0 (Timer Wert in BCD also nur 0-9 pro Byte möglich)
T   MW   50          //transferiere Wert aus EW0 nach AKKU1

L   MW   50          //lade MW50 in AKKU1
UW   W#16#FFF        //ausmaskieren der Eingänge E0.4 - E0.7 über die UW Verknüpfung
NOP   0              //Wert aus AKKU1 (EW0) nach AKKU2 schieben und Wert "0FFF" in AKKU1
T   MW   52          //transferiere das Ergebniss aus der UND-Verknüpfung in das MW52

L   MW   52          //lade MW52 in AKKU1
OW   W#16#2000        //Zeitbasis im linken Byte (1,2,oder 3 sind möglich; 2 für Sek.) lade Wert "2000" in AKKU 1
NOP   0              //Wert aus AKKU1 nach AKKU2 schieben und Wert "2000" in AKKU1
T   MW   54          //transferiere das Ergebniss aus der ODER-Verknüpfung in das MW54

NOP   0              //somit steht der Timer Wert aufgearbeitet in AKKU 1 zur verfügung

U   E   0.7
SV   T   10          //der T10 bedient sich aus dem AKKU1

```