

**STEP 7**

**S7-300**

# **MICROMASTER – Steuerwort**

Programmieranleitung

© U. Ohm [www.u-ohm.de](http://www.u-ohm.de)

**1. Einstellung über die Variablen-tabelle VAT1 und Datenbaustein DB1****1.1 Separate Abfrage von Steuerwort und Sollwert**

**Hinweis:** Steuerwort und Sollwert werden in 2 aufeinander folgenden Wörtern abgelegt.

*Die Bits werden von Hand in der VAT geändert und im DB gespeichert...*

Dabei ist es beliebig, welcher DB verwendet wird, z.B. hier: DB 1.  
Auch das Datenwort kann beliebig gewählt werden.

Beispiel: Das Steuerwort soll im DB1.Datenwort 0 stehen (DB1.DBW0)

Das Wort setzt sich aus 2 Bytes zusammen. Hier: Byte 0 und Byte 1.  
Die Wortnummer ist identisch mit der kleineren Bytenummer.  
Das Datenwort für den Sollwert folgt im nächsten freien Wort.

Der Sollwert wird im DB1.Datenwort 2 abgelegt. (DB1.DBW2)

Das Datenwort 2 setzt sich aus den Bytes 2 und 3 zusammen.

*... und von dort mit folgendem Programm abgefragt.*

```
L   DB1.DBW0    //Lade den Wert in den Akku ...
T   PAW 256     //... und transferiere ihn an das Ausgangswort der SPS
      (damit ist die Steuerwortabfrage des MICROMASTERS über den Bus verknüpft).
```

```
L   DB1.DBW2    //Lade den Wert in den Akku ...
T   PAW 258     //... und transferiere ihn an das Ausgangswort der SPS
      (damit ist die Sollwertabfrage des MICROMASTERS über den Bus verknüpft).
```

## 1.2 Freigabe über einen separaten Schalter und Übernahme von Steuerwort und Sollwert aus DB

Die Freigabe soll jetzt nicht mehr über die VAT erfolgen, sondern durch eine Schalterabfrage.

Ein Datenbaustein (DB) ist der Ort, an dem Informationen dauerhaft gespeichert werden können, d.h. auch nach dem Ausschalten der SPS.  
Durch dieses Programm soll beispielhaft über ein Bit der Micromaster ein- bzw. ausgeschaltet werden. Dieses Steuerbit soll über den DB abgefragt werden. Auf den ersten Blick mögen die folgenden 2 Programmzeilen überflüssig erscheinen.

```
L      2#0          //Der Wert „0“ ...
=      DB1.DBX1.0  //... wird dem Steuerbit 1.0 bei jedem Programmzyklus zugewiesen.
```

Sie sind aber bei dieser Programmiervarianten erforderlich, damit „Betriebsbereit“ auch wieder ausgeschaltet wird, wenn der zugehörige Schalter auf „0“ gesetzt wird.

```
U      E 0.0       //Wenn das Eingangs-Bit auf „1“ gesetzt ist...
SPBN end          //(falls nicht, springe zur Sprungmarke <end>), sonst...
=      DB1.DBX1.0  // ... wird der Wert des abgefragten Bits einem bestimmten Bit im DB
// zugewiesen. (Jedes Bit des Steuerwortes wäre möglich).
```

Warum „DBX1.0“?

„X“ steht für die Bit-Verarbeitung; „W“ für eine Wort-Operation.  
„1.0“ ist das niederwertigste Bit des Steuerwortes (siehe auch nächste Seite).

Laut Steuertabelle wird mit Bit 0 der Zustand „Betriebsbereit“ geschaltet.

Die Bits des Wortes 0 (das ist hier das Steuerwort) sind in folgender Weise angeordnet:  
Bit 0.7 ... Bit 0.0 – Bit 1.7 ... Bit 1.0  
Dabei ist Byte 0 das höchstwertige [links] und Byte 1 das niederwertige [rechts]  
Innerhalb eines Bytes ist Bit 7 das höchstwertigste und Bit 0 das niederwertigste.

```
end:   NOP 0       //Sprungziel ohne weitere Operation.
```

**weiter wie gehabt:**

```
L      DB1.DBW0    //Lade den Wert in den Akku ...
T      PAW 256     //... und transferiere ihn an das Ausgangswort der SPS
        (damit ist die Steuerwortabfrage des MICROMASTERS über den Bus verknüpft).
L      DB1.DBW2    //Lade den Wert in den Akku ...
T      PAW 258     //... und transferiere ihn an das Ausgangswort der SPS
        (damit ist die Sollwertabfrage des MICROMASTERS über den Bus verknüpft).
```

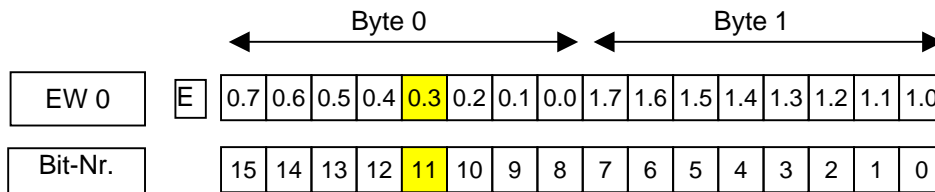
### 1.3 Freigabe über separate Schalter und Übernahme von Steuerwort und Sollwert aus DB

Nicht nur die Freigabe, sondern auch die Drehrichtung soll über Schalter gesteuert werden.

```

L      2#0           //Der Wert „0“ ...
=      DB1.DBX1.0   //... wird dem Steuerbit 1.0 bei jedem Programmzyklus zugewiesen.
CLR    DB1.DBX0.3   // Ein Bit wird ohne weitere Verknüpfung auf „0“ vorbereitet
=      DB1.DBX0.3   //Zuweisung der „0“ (Grundstellung bei jedem Programmzyklus)
                        //(siehe gleich die Erläuterung zu Bit 11)
U      E 1.0        //Wenn das Eingangs-Bit auf „1“ gesetzt ist...
SPBN end // (falls nicht, springe zur Sprungmarke <end>), sonst...
=      DB1.DBX1.0   // ... wird der Wert des abgefragten Bits im DB zugewiesen
U      E 0.1        //Wenn der Schalter für Linkslauf auf „1“ gesetzt wird, ...
SPBN end
=      DB1.DBX0.3   //... wird Bit 11 der Steuertabelle (das 12. Bit des Steuerwortes !) ...
                        //... ebenfalls auf „1“ gesetzt (Linkslauf).

```



```

end:  L      DB1.DBW0   //Lade den Wert in den Akku ...
      T      PAW 256   //... und transferiere ihn an das Ausgangswort der SPS
      L      DB1.DBW2   //Lade den Wert in den Akku ...
      T      PAW 258   //... und transferiere ihn an das Ausgangswort der SPS

```

**1.4 Übergabe des Steuerwortes als Eingangswort an den DB  
(Abfrage der 16 Eingangsschalter)**

Mit der Wort-Abfrage kann jedes einzelne Bit des Steuerwortes separat am Schaltableau unter Betriebsbedingungen verändert werden.

```
L    EW 0          //Lade das Eingangswort (Bit 0.7 ... Bit 0.0 – Bit 1.7 ... Bit 1.0)
                          //(Auf die Reihenfolge, bzw. Anordnung achten!!)
T    DB1.DBW0     // Transferiere das Steuerwort in den Datenbaustein ab DBB 0
```

**weiter wie gehabt:**

```
L    DB1.DBW0     //Lade den Wert in den Akku ...
T    PAW 256      //... und transferiere ihn an das Ausgangswort der SPS
L    DB1.DBW2     //Lade den Wert in den Akku ...
T    PAW 258      //... und transferiere ihn an das Ausgangswort der SPS
```

**1.5 Übergabe des Steuerwortes als Eingangswort direkt an das PAW  
(Abfrage der 16 Eingangsschalter)**

Es ist nicht unbedingt erforderlich, die Steuerbits aus dem DB abzurufen, sondern sie können auch direkt an das PAW geleitet werden:

L	EW 0	//Lade das Eingangswort (Bit 0.7 ... Bit 0.0 – Bit 1.7 ... Bit 1.0)
T	PAW 256	//... und transferiere ihn an das Ausgangswort der SPS
L	DB1.DBW2	//Lade den Wert in den Akku ...
T	PAW 258	//... und transferiere ihn an das Ausgangswort der SPS

**1.6 Übertragung von Steuerwort und Sollwert mit einer Operation (Doppelwort)**

Wie in 1.4 gehabt:

L      EW 0  
T      DB1.DBW0

Die Abfrage beider Datenwörter des DB kann auch als Doppelwortabfrage programmiert werden. An Stelle des Wortes (W) wird dann nur ein „D“ (für Doppelwort) geschrieben.

L      DB1.DBD0      //Lade das Doppelwort (Byte 0, 1, 2, 3)

(dabei ist Byte 0 das höchstwertige [links] und Byte 3 das niederwertigste {rechts})  
(0.7 ... 0.0 – 1.7 ... 1.0 – 2.7 ... 2.0 – 3.7 ... 3.0)

T      PAD 256      //Transferiere das Doppelwort (Byte 256, 257, 258, 259)

Da Steuerwort und Hauptsollwert beim Micromaster direkt hintereinander liegen, ist es möglich, diese beiden Wörter mit der Doppelwort-Operation zu übertragen.

## 2 Änderung der Steuerung aus dem SPS-Programm heraus

Man könnte verschiedene „Werte“ des Steuerwortes oder / und des Sollwertes im SPS-Programm vorbereiten und durch einzelne Steuerschalter die neuen Steuerwörter laden.

Zuerst kommen wieder die **Bedingungen für jeden Zyklusstart**.

```
L 2#100_0111_1110 //Lade als Dualzahl kodiert das Betriebsmuster (ausgeschaltet)
// „_“ dient nur zur Verbesserung der Lesbarkeit
T DB1.DBW0
L W#16#0000 //Wert „0“ hex.
T DB1.DBW2 //Hauptsollwert der Drehfrequenz wird auf „0“ gesetzt
```

Die max. Drehfrequenz wird direkt im Micromaster parametrierd. Über den Hauptsollwert wird der prozentuale Anteil eingestellt. Dabei bedeutet hex. 4000 = 100 %. Proportional dazu lassen sich somit leicht alle Drehfrequenzen von  $n = 0$  (hex. 0000) bis  $n_{\max}$  einstellen.

Jetzt folgen die Varianten:



a) für das Steuerwort

```

U   E 0.0           //Wenn der Eingang „1“ führt, ...
SPBN w1           //... wenn nicht, springe zur Marke <w1>, sonst...
L   2#0000_0100_1111_1111 (Start, Rechtslauf, ...)
T   DB1.DBW0      //der neue Wert wird in den DB geschrieben (Steuerwort !)
SPA dr_1         //springe absolut zur Drehfrequenzauswahl

```

```

w1: U   E 0.1      // hier könnten weitere Steuerwortalternativen vorbereitet werden.
      SPBN dr_1
      L   2# .....
      T   DB1.DBW0
           //hier ist kein Sprung erforderlich, da zwingend mit der Drehfrequenz fortgesetzt
           //werden soll.

```

b) für die Drehfrequenz:

```

dr_1: U   E 0.2           //Wenn der Eingang „1“ führt, ...
      SPBN dr_2         //... wenn nicht, springe zur Marke <dr_2>, sonst...
      L   W#16#1000     // Vorwahl Drehfrequenz 1 (hex.) = 25 %
      T   DB1.DBW2     //der neue Wert wird in den DB geschrieben (Sollwert !)
      SPA end           //absoluter Sprung, wenn diese Drehfrequenz gewählt wurde

```

```

dr_2: U   E 0.3           //Wenn der Eingang „1“ führt, ...
      SPBN end         //... wenn nicht, springe zur Marke <end>, sonst...
      L   W#16#4000     // Vorwahl Drehfrequenz 2 (hex.) = 100 %
      T   DB1.DBW2     //der neue Wert wird in den DB geschrieben

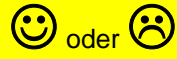
```

wie gehabt: Übertragung der vorgewählten Parameter an PAW > Bus > Micromaster

```

end:   L   DB1.DBD0    //Lade beide Wörter in den Akku ...
      T   PAD 256      //... und transferiere sie an das Ausgangsdoppelwort

```



bitte an [u.ohm@freenet.de](mailto:u.ohm@freenet.de)