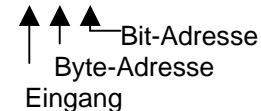


Adressierung

Damit man versteht, wie ein Automatisierungssystem (AS) das Steuerprogramm bearbeitet, ist es notwendig, die Begriffe **Bit**, **Byte**, **Wort** und **Doppelwort** zu kennen.

- Ein **Bit** ist die Einheit für eine Binärstelle oder ein Binärzeichen. Es ist die kleinste informationstechnische Einheit und kann nur den Signalzustand "0" oder "1" annehmen. Mehrere Bits können zu größeren Einheiten zusammengefasst werden.
Damit die Signalgeber und Stellgeräte angesprochen werden können, müssen sie eine Adresse im AS besitzen. Jedes Bit erhält im AS eine Nummer, die so genannte **Bit-Adresse**. 0 - 7 stehen dafür zur Verfügung.
- Ein **Byte** stellt eine Einheit von **8 Bit** dar. Es wird z.B. für die Zusammenfassung von Signalzuständen von 8 Ein- oder Ausgängen benutzt, um damit z.B. Zahlen mit 2 Ziffern hexadezimal darzustellen.
Die einzelnen Bytes erhalten ebenfalls eine Nummer, die **Byte-Adresse**. Die Nummer hängt bei einigen CPUs vom Steckplatz der Baugruppe ab, bei anderen kann sie frei gewählt werden.
Die Byte-Adressen werden zusätzlich durch Bezeichnungen wie Eingang (**E**) und Ausgang (**A**) oder Merker (**M**) noch näher festgelegt. Spricht man z.B. von dem Eingangs-Byte EB 0, so meint man alle 8 Signale (Klemmen 0 - 7), die an dem Byte 0 angeschlossen sind.
Die Bit-Adresse wird durch einen Punkt von der Byte-Adresse getrennt (z. B. E 3.2).



Die Darstellungsmöglichkeiten aller Bits sind:

- a.) als Bitmuster : 0000 0000 bis 1111 1111
- b.) hexadezimal : 00 bis FF

Die Bitmuster können als Zahlen gedeutet werden, können aber auch nur Schaltbefehle für die zugehörigen Ausgänge darstellen. Somit können 8 Ausgänge mit einem einzigen Befehl – entsprechend der vorgewählten Zahl – geschaltet werden.

Zur Erläuterung: Zuordnung von Dualzahlen zu Hexadezimalzahlen

Dual	Hex	Dezimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

Dual	Hex	Dezimal
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Die Bitmuster von jeweils 4 Stellen dual werden hexadezimal 1-stellig dargestellt.

1001 1100₂
9 C_{hex}

Darstellung und Wert einer 8-stelligen Dualzahl

1) Operand	E0.7	E0.6	E0.5	E0.4	E0.3	E0.2	E0.1	E0.0
------------	------	------	------	------	------	------	------	------

2) Potenz	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
-----------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

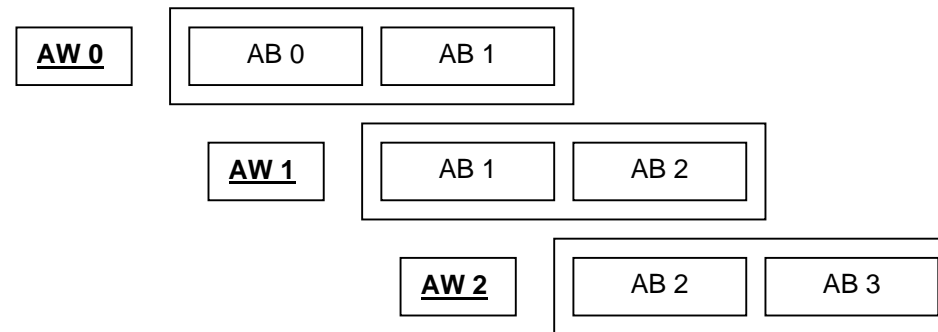
3) Stellenwert	128	64	32	16	8	4	2	1
----------------	-----	----	----	----	---	---	---	---

4) Bitmuster dual	1	0	0	1	0	0	1	1
-------------------	---	---	---	---	---	---	---	---

5) Wert dezimal	128 + 0 + 0 + 16 + 0 + 0 + 2 + 1								= 147 _{dez}
-----------------	----------------------------------	--	--	--	--	--	--	--	----------------------

Ein **Wort** besteht aus **2 Byte** bzw. **16 Bit**. Mit einem Wort können z.B. 16 Eingänge oder Ausgänge zusammengefasst werden. Dadurch lassen sich durch eine gemeinsame Auswertung der 16 „Klemmen“ $2^{16} = 65.536$ verschiedene Werte bestimmen oder zuweisen.

- Zwei Eingangs-Bytes sind z.B. ein Eingangswort (EW). Dabei liegen die Bytes direkt nebeneinander. Das Eingangswort 0 (EW0) enthält die Eingangs-Bytes 0 (EB0) und 1 (EB1). Bei einem Wort entspricht die Wortadresse immer der des Bytes mit der kleineren Nummer.



Achten Sie darauf, dass z.B. AB 1 sowohl zu AW0 als auch zu AW1 gehört. Bei der Programmierung müssen Sie darauf achten, dass Sie nur jedes zweite Wort verwenden, damit es keine Überschneidungen gibt.

- Ein **Doppelwort (ED bzw. AD)** besteht aus **2 Wörtern** oder **4 Byte** oder **32 Bit**. Ein Doppelwort ist die größte Einheit, die von einem Automatisierungssystem in einem Schritt verarbeitet werden kann. Das ED0 besteht aus den Eingangs-Bytes EB0, EB1, EB2 und EB3. Wenn z.B. das ED4 verwendet wird, ist erst das Byte 8 frei für zusätzliche Abfragen.

Bei der **absoluten Adressierung** wird direkt die Adresse (z.B. des Eingangs E 1.0) angegeben. In diesem Fall ist keine Symboltabelle erforderlich, das Programm ist aber schlechter lesbar.

Die **symbolische Adressierung** ermöglicht es, anstelle von direkten Adressen mit Symbolen (z.B. "MOTOR_EIN") zu arbeiten. Diese Symbole werden in der Symboltabelle den tatsächlichen Adressen zugeordnet. Man nannte sie daher auch "Zuordnungsliste".

Steueranweisung

Die Steueranweisung ist die kleinste Einheit eines Steuerprogramms. Sie besteht in der Anweisungsliste und auch im Programmspeicher aus dem **Operationsteil** und dem **Operandenteil**.

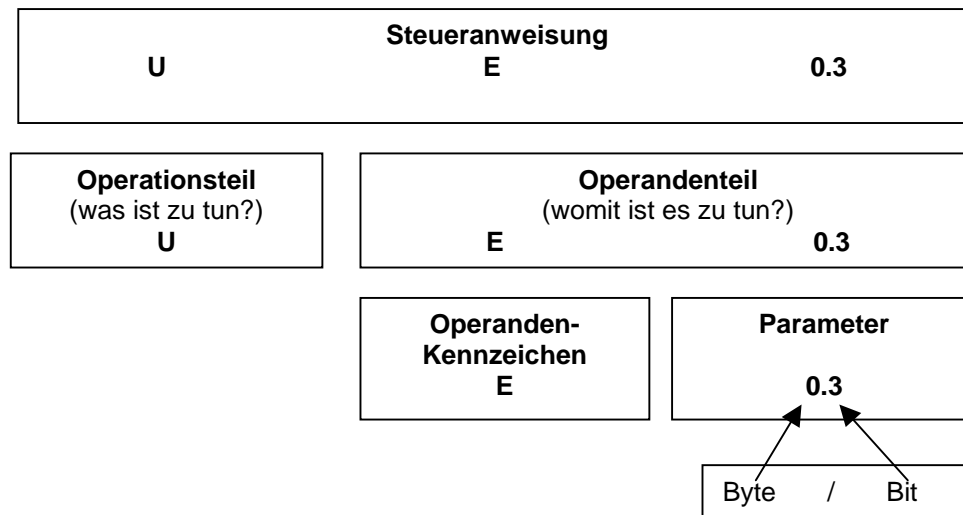
Beispiel für digitale Abfragen: U E 0.3

Der Operationsteil (U für die logische Verknüpfung UND) bestimmt, **was** zu tun ist.

Der Operandenteil (E 0.3) gibt an, **an welcher Adresse** es zu tun ist.

Die "Klemmleiste" E oder A wird als **Operandenkennzeichen** bezeichnet.

Der **Parameter** ist die Adresse des Operanden (0.3)



Ganz anders sind die Operationen für Byte- bzw. Wordanwendungen:

L EW 2 Ladebefehl in AWL für das Eingangswort 2 (Byte 2 + Byte 3) der DI-Baugruppe.

T AW 4 Transferbefehl in AWL für das Ausgangswort 4 der DA-Baugruppe.

L PEW 288 Ladebefehl in AWL für das Eingangswort 288 der Analogbaugruppe (Byte 288 + Byte 289)

T PAW 288 Transferbefehl in AWL für das Ausgangswort 288 der Analogbaugruppe (Byte 288 + Byte 289)