

IAS

Zusammenfassung

Dario Scheuber / [Quelldateien](#)

Inhaltsverzeichnis

Data types	1
Array	1
Struct	2
Timer TON	2

Data types

Data type	Lower bound	Upper bound	Memory space
BYTE	0	255	8 bit
WORD	0	65535	16 bit
DWORD	0	4294967295	32 bit
LWORD	0	$2^{64}-1$	64 bit
SINT	-128	127	8 bit
USINT	0	255	8 bit
INT	-32768	32767	16 bit
UINT	0	65535	16 bit
DINT	-2147483648	2147483647	32 bit
UDINT	0	4294967295	32 bit
LINT	-2^{63}	$2^{63}-1$	64 bit
ULINT	0	$2^{64}-1$	64 bit
BOOL	FALSE (0)	TRUE (1)	8 bit
REAL	$-3.402823e+38$	$3.402823e+38$	32 bit
LREAL	$-1.7976931348623158e+308$	$1.7976931348623158e+308$	64 bit

```
VAR
    var :INT:= 10;
END_VAR
```

Array

Declaration 1:

One-dimensional array of 10 integer elements

Lower index limit: 0 Upper index limit: 9

```
VAR
    aCounter : ARRAY[0..9] OF INT;
```

```
bCounter : ARRAY[0..9] OF INT := [0, 10, 20, 30, 40, 50, 60, 70, 80, 90];
END_VAR
```

Struct

Eine Struktur ist ein benutzerdefinierter Datentyp und fasst mehrere Variablen mit beliebigen Datentypen zu einer logischen Einheit zusammen. Die innerhalb einer Struktur deklarierten Variablen werden als Komponenten bezeichnet.

```
TYPE ST_CONTROL :
STRUCT
    bitOperationEnabled : BIT;
    bitSwitchOnActive   : BIT;
    bitEnableOperation  : BIT;
    bitError            : BIT;
    bitVoltageEnabled   : BIT;
    bitQuickStop        : BIT;
    bitSwitchOnLocked   : BIT;
    bitWarning          : BIT;
END_STRUCT
END_TYPE
```

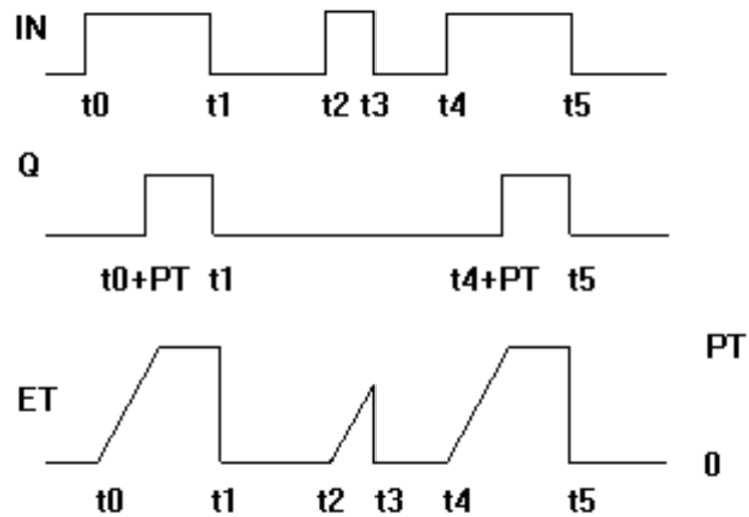
Timer TON

Wenn IN = FALSE ist, sind die Ausgaben FALSE bzw. 0. Sobald IN = TRUE ist, wird in ET die Zeit in Millisekunden hochgezählt, bis der Wert gleich dem in PT ist, dann bleibt er gleich. Q ist TRUE wenn IN = TRUE und ET = PT ist. Sonst ist Q = FALSE. Q hat somit eine steigende Flanke, wenn die in PT in Millisekunden angegebene Zeit abgelaufen ist.



```
VAR_INPUT
    IN : BOOL; (* starts timer with rising edge, resets timer with falling edge *)
    PT : TIME; (* time to pass, before Q is set *)
END_VAR
```

```
VAR_OUTPUT
    Q : BOOL; (* is TRUE, PT seconds after IN had a rising edge *)
    ET : TIME; (* elapsed time *)
END_VAR
```



```

PROGRAM MAIN
VAR
    timer1 :TON;
    bTimerDone:BOOL;
    bTimerIn:BOOL;
END_VAR

timer1(IN:=bTimerIn,PT := T#20S);
IF timer1.Q THEN
    //timer1.IN := FALSE; geht nicht
    timer1(IN:=FALSE);

    timer1.IN := FALSE;
    timer1();
END_IF

```