

BECKHOFF



- Variable Types
- Declarations
- Variable Scope
- Initial Values
- Constants

What is a Variable?

- A Variable is a name given to a location in memory that stores a value
- A Variable has up to 5 properties
 1. Name
 2. Size (Defined by the Type)
 3. Value
 4. Memory Location
 5. PLC Address

- In accordance with IEC 61131-3 a variable name must adhere to the following rules
 1. The variable must begin with a Letter or an Underscore
 2. The variable can be followed by Letters, Underscores, or Numbers
- No distinction is made between Uppercase and Lowercase Letters

The following are not permitted

- Special characters can not be used (!@#\$%^&*)
- Blanks or Spaces are not allowed

- Keywords are identifiers specified by IEC61131-3. They are thus fixed components of the syntax and therefore may not be used for other purposes.

Examples

- Standard operators **AND, OR, NOT...**
- Standard types **BOOL, INT, REAL...**
- Custom Types **TYPE, STRUCT**
- Block types **FUNCTION, FUNCTION_BLOCK, PROGRAM**

- The comments are delimited by character strings with (* or *) at the beginning and at the end. Comments may be placed wherever spaces are also permitted. Exception: within string literals.

(*Digital Input*)

bStart AT %IX0.0 :BOOL;(*Machine Start*)

(*Analog Input*)

TempK1 AT %IW10 (*Byte 10-11*) :WORD;

bStart AT %IX0.0 :BOOL; // Machine Start

(* bStop at %IX0.0 : BOOL; (*Machine Stop*) *)

Comments

- If comments are placed on the line above or after the variable, then the comment will appear in the tooltip when hovering the mouse over the variable name.

VAR

(Machine Start *)*

xStart : BOOL;

xStop : BOOL; *(* Machine Stop *)*

END_VAR

1

xStart;

2

VAR MAIN.xStart : BOOL
Machine Start

3

4

3

xStop;

4

VAR MAIN.xStop : BOOL
Machine Stop

5

6

What type of data will the variable hold and how big is the data?

Elementary data types

BECKHOFF

Type	Lower	Upper	Size (Bits)	Prefix	
BOOL	FALSE	TRUE	1	x b	
BYTE			8	by	Bitstring
WORD			16	w	Bitstring
DWORD			32	dw	Bitstring

Elementary data types

BECKHOFF

Type	Lower	Upper	Size (Bits)	Prefix
SINT	-127	127	8	si
USINT	0	255	8	usi
INT	-32 768	32 767	16	i
UINT	0	65 535	16	ui
DINT	-2 147 483 648	2 147 483 648	32	di
UDINT	0	4 294 967 295	32	udi
LINT	-2^{63}	$2^{63} - 1$	64	li
ULINT	0	$2^{64} - 1$	64	uli

Elementary data types

BECKHOFF

Type	Lower	Upper	Size (Bits)	Prefix
TIME_OF_DAY	TOD#0:0:0	TOD#23:59:59	32	tod
DATE	D#1970-01-01 D#YYYY-MM-DD	D#2106-2-7	32	date
DATE_AND_TIME	DT#1970-01-01-00:00:00	DT#2106-2-7-6:28:15	32	dt
TIME	T#0s	T#49d17h2m47s 295ms	32	tim

Elementary data types

Type	Lower	Upper	Size	Prefix	
REAL	$-1.18 \cdot 10^{-38}$	$3.4 \cdot 10^{38}$	32 Bits 4 Byte	r	
LREAL	$-2.22 \cdot 10^{-308}$	$1.798 \cdot 10^{308}$	64 Bits 8 Byte	lr	

String data type

BECKHOFF

Type	Description	Example	Size	Prefix
STRING	String in ASCII code. Standard length 80 characters. Maximum length 255. Strings are null-terminated	'1234ABCDE'	80 +1	s
		'ABCDE\$R\$L'		
		'ABCDE\$0D\$0A'		

String length specifications

Example declaration	Assignment	Result SIZEOF	Result LEN
sVar : STRING;	sVar:='ABC';	81	3
sVar1 :STRING(1);	sVar := 'X';	2	1
sVar: STRING(255);	sVar:='ABC';	256	3

String data type

Constants

\$<2 Hex values>		ASCII Code
\$0D		CR
\$R	\$r	CR
\$L	\$l	Line Feed
\$N	\$n	New Line
\$T	\$t	Tab

String data type

BECKHOFF

Type	Description	Example	Description	Prefix
WSTRING	String in Unicode format	“Обучение”	Level 0,Block 0x0400-0x4FFF Cyrillic	ws
		“培训、讲座、研讨会”		
		“Training, seminar”	Level 0 Block 0x0000-0x007F Basic Latin	

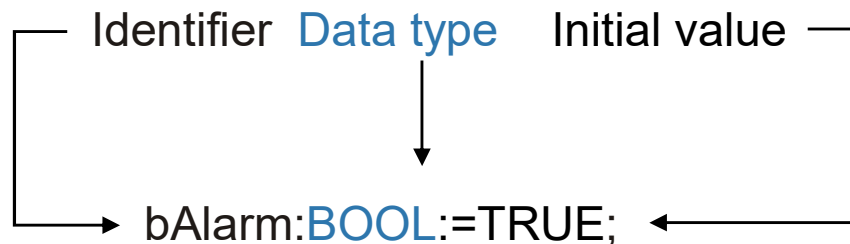
Examples of Literals

BECKHOFF

Variable Type		Examples			
BOOL		TRUE	2#1	16#1	1
		FALSE	2#0	16#0	0
WORD, DWORD		2#1010111111111110		16#AFFE	45054
INT		2#10000000000000001		16#8001	-32768
TIME		t#1h		t#60m	t#3600000ms
d	day				
h	hours				
m	min	t#0.5d		t#12h	t#43200000ms
s	sec				
ms	ms				
REAL		0.3333	3.333e-1		

Declaring a Variable

- All variables must be defined between **VAR** and **END_VAR**
- A variable possesses a name behind which a value (number, string, date, etc.) is stored. Variables are characterised by the fact that their contents can be changed at runtime.

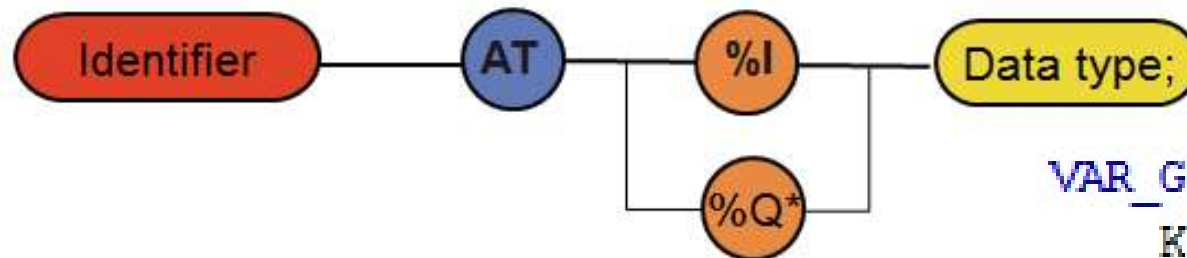


The physical-logical storage location of this variable is unknown to the user (unlocated)

Declaring a Variable

BECKHOFF

- It is possible when declaring a variable to link the name with an address that must be explicitly specified.
- For the allocation of inputs and outputs of the hardware the **incomplete** location is to be carried out with I* or Q*



VAR_GLOBAL

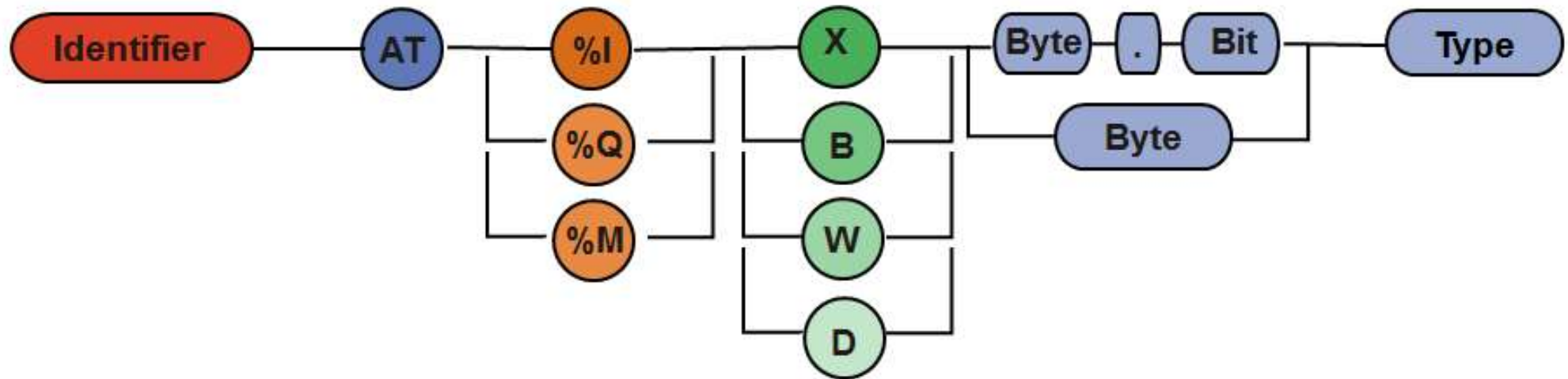
```
K12531Velocity  AT%Q* :INT;  
K12531Ctrl      AT%Q* :BYTE;  
K12531ExtCtrl   AT%Q* :WORD;
```

```
K12531Position  AT%I* :UINT;  
K12531Status    AT%I* :BYTE;  
K12531ExtStatus AT%I* :WORD;
```

END_VAR

Declaring a Variable

- Hard coded addresses



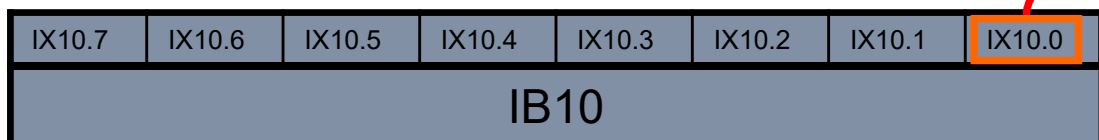
These variables possess a clear address
(located)

In TwinCAT 3 incompletely located variables can be used for inputs and outputs

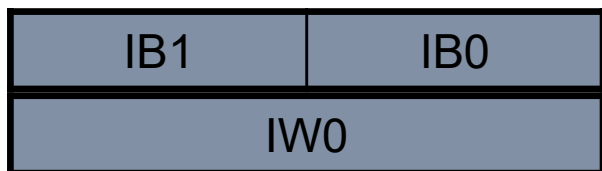
Applications for %M variables can be solved simply with Unions and direct masking

Address division

Examples:



Din0 AT%IX10.0 : BOOL;



Ain AT%IB0 : INT;

↕ equivalent

Ain AT%IW0 : INT;



BitVar AT%IX22.1 : BOOL;

Posi AT%IB20 : UDINT;

↕ equivalent

Posi AT%ID20 : UDINT;

Variable Scope

Local variables are restricted to the block or POU in which they were declared.

Keywords

VAR ..

END_VAR

VAR_INPUT ..

END_VAR

VAR_IN_OUT ..

END_VAR

VAR_OUTPUT ..

END_VAR

Global variables are known in each block or POU within a project.

Keywords

VAR_GLOBAL ..

END_VAR

VAR_CONFIG ..

END_VAR

Variable Scope

- Global Variables can be read and written from anywhere in the PLC program
- Local Variables can only be written from within the POU where they are defined
- The local variable of any POU can be read by first accessing the POU instance that the variable is defined in and then using the '.' to access the local variables defined within that POU
- Local variables cannot be written from another POU

Initial Values

- All Variables have the option of assigning an initial value
- This value will be written to memory when the PLC starts, after which the code of the PLC will control the value

VAR_GLOBAL

(xStart is of type BOOL and has an initial value of FALSE *)*

xStart : **BOOL** := **FALSE**;

(iProductNumber is of type INT and has an initial value of 123 *)*

iProductNumber : **INT** := **123**;

(lPressure is of type LREAL and has an initial value of 2.3 *)*

lPressure : **LREAL** := **2.3**;

END_VAR

Constants

- Variables defined as Constants cannot be written to by the PLC
- Constants are declared similar to initial values
- Use of the keyword 'Constant' at the beginning of the declaration section signals the compiler that the variable is a constant.

```
1  VAR_GLOBAL CONSTANT
2      iMachineNumber : INT := 234;
3
4  END_VAR
```

Beckhoff Automation GmbH & Co. KG

Headquarters
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963-0
Fax: +49 5246 963-198
E-Mail: info@beckhoff.com
Web: www.beckhoff.com

© Beckhoff Automation GmbH & Co. KG

All images are protected by copyright. The use and transfer to third parties is not permitted.

Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this presentation may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

The information provided in this presentation contains merely general descriptions or characteristics of performance which in case of actual application do not always apply as described or which may change as a result of further development of the products. An obligation to provide the respective characteristics shall only exist if expressly agreed in the terms of contract.

Beckhoff Automation GmbH & Co. KG

Unternehmenszentrale
Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0
Fax: +49 5246 963-198
E-Mail: info@beckhoff.de
Web: www.beckhoff.de

© Beckhoff Automation GmbH & Co. KG

Alle Bilder sind urheberrechtlich geschützt. Die Weitergabe und Nutzung durch Dritte ist nicht gestattet.

Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH. Die Verwendung anderer in dieser Präsentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichen führen.

Die Informationen in dieser Präsentation enthalten lediglich allgemeine Beschreibungen bzw. Leistungsmerkmale, welche im konkreten Anwendungsfall nicht immer in der beschriebenen Form zutreffen bzw. welche sich durch Weiterentwicklung der Produkte ändern können. Die gewünschten Leistungsmerkmale sind nur dann verbindlich, wenn sie bei Vertragsabschluss ausdrücklich vereinbart werden.