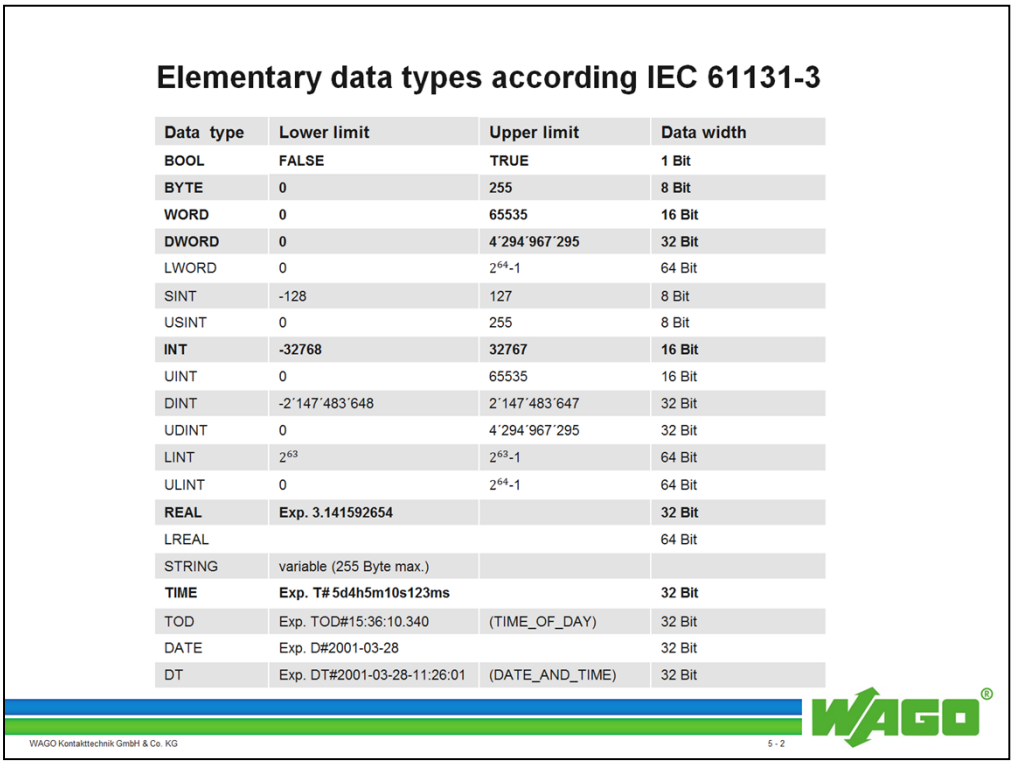
5

Data types & variables

- Data types
- Variables



Data type	Lower limit	Upper limit	Data width
BOOL	**FALSE**	**TRUE**	**1 Bit**
BYTE	**0**	**255**	**8 Bit**
WORD	**0**	**65535**	**16 Bit**
DWORD	**0**	**4´294´967´295**	**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	-2´147´483´648	2´147´483´647	32 Bit
UDINT	0	4´294´967´295	32 Bit
LINT	2^{63}	2^{63}-1	64 Bit
ULINT	0	2^{64}-1	64 Bit
REAL	**Exp. 3.141592654**		**32 Bit**
LREAL			64 Bit
STRING	variable (255 Byte max.)		
TIME	**Exp. T#5d4h5m10s123ms**		**32 Bit**
TOD	Exp. TOD#15:36:10.340	(TIME_OF_DAY)	32 Bit
DATE	Exp. D#2001-03-28		32 Bit
DT	Exp. DT#2001-03-28-11:26:01	(DATE_AND_TIME)	32 Bit

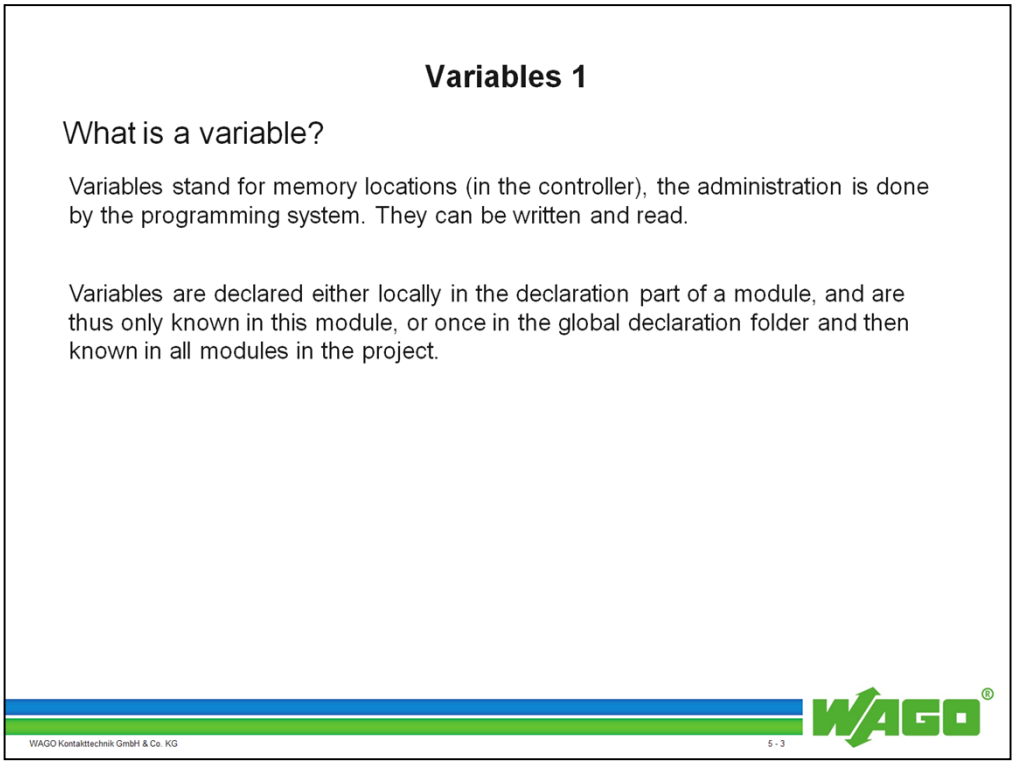
There has to be a standardization of data types (data formats) for the uniform presentation of process data beyond system boundaries of different manufacturers.

At this point, IEC 61131-3 introduces specific requirements for PLC systems.

The most commonly used elementary data types

- BOOL
- BYTE
- WORD
- INT
- DWORD
- REAL
- TIME

are marked in the table (above) in **bold**.



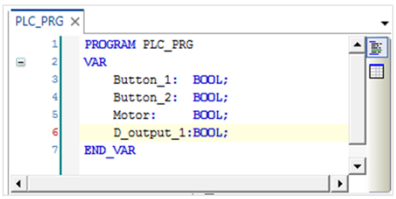
A local variable is only known in the module in which it is declared. In the standard case, most variables are local. In case values need to be exchanged with other modules/programs, they can be declared as global variables.

It is permissible to call a local variable and a global variable by the same name. In this case, the variable is used that is “closer” to the module, thus a local variable.

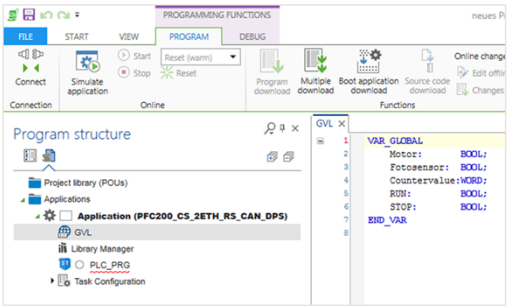
When considering identically named variables, the following sequence arises:

Local variable → Global variable → Variable from a library

Variables 1



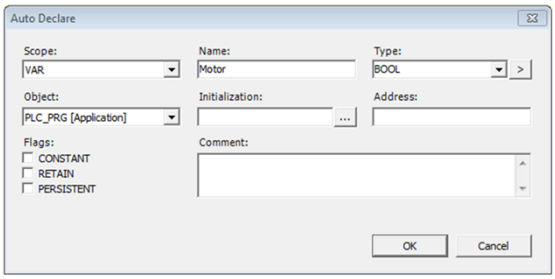
Local declaration



Global declaration

Variables 2

Declaration of variable class



Class	Description
VAR	Local variable in a POU (Standard)
VAR_INOUT	Input variable
VAR_OUTPUT	Output variable
VAR_IN_OUT	Input/output variable
VAR_GLOBAL	Global variable, known in all POUs

All variables that are only used in this module are declared in the declaration part of a module.

These can be

- Local variables (VAR = used for temporary data, that is only valid for processing the respective module)
- Input variables (VAR_INPUT = used for data that is transmitted into the module)
- Output variables (VAR_OUTPUT = used for output data that is generated in the module and output to other program parts)
- Input/output variables (VAR_IN_OUT = used for data that is transmitted into the module, processed there, and then output to additional program parts.

The declaration syntax follows the standard of IEC 61331-3

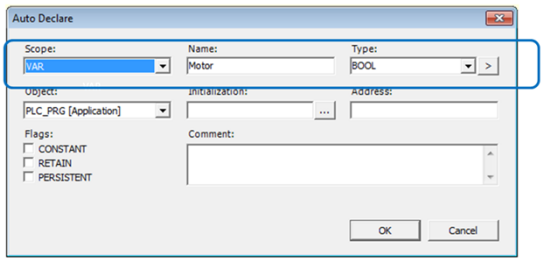
The declaration VAR_GLOBAL moves the variable out of the local declaration part into the list of Global_variables (to be found in the “Resources” tab 4).

All variables, constants, or residual variables can be declared global variables that should be known in the entire project.

Variables 3

Declaration of a variable

Required entries



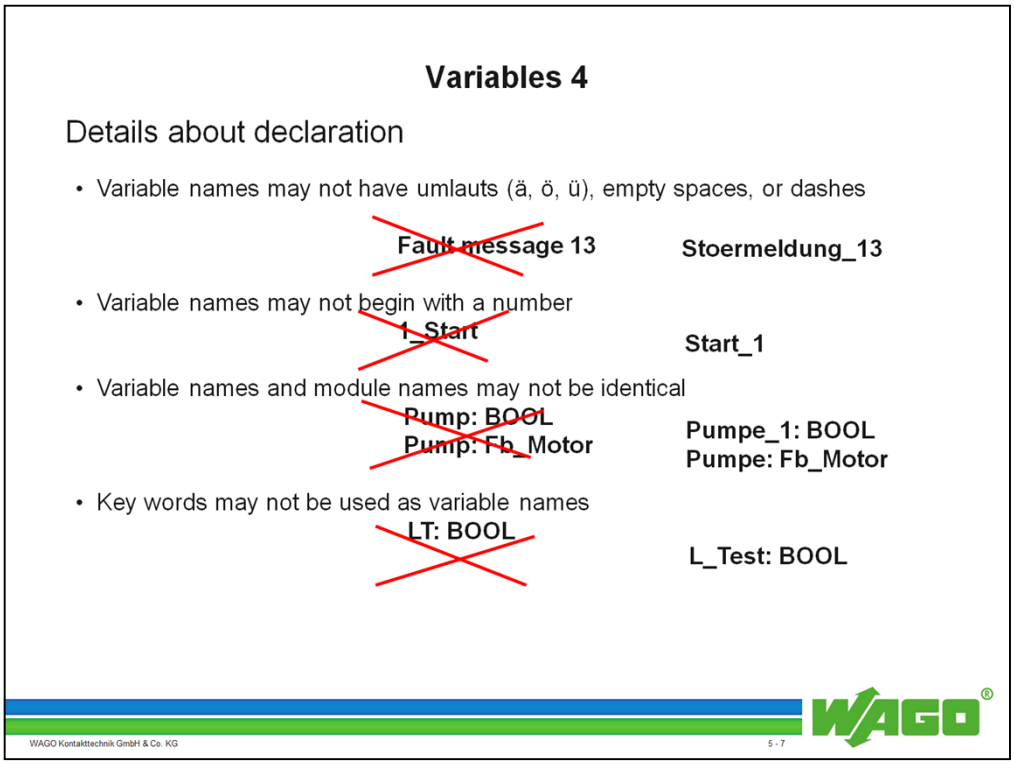
- Name
- Type
- Initial value
- Address
- Comment
- CONSTANT — Constants; the value is not changed during a program execution (only read)
- RETAIN — These variables retain their value after an uncontrolled termination of the control or a reset
- PERSISTENT — In contrast to Retain variables, these variables retain their value even after a cold start, i.e. when the control is terminated normally and has been restarted, or after a download. Persistent variables are not automatically Retain variables!

WAGO Kontakttechnik GmbH & Co. KG 5 - 6

Retain/ Persistant variables

Retain variables can keep their values even after warm reset or power off..
The behaviour is shown in the table below.

	Reset warm (supply failure)	Reset cold	Reset original
Standard	0	0	0
VAR RETAIN PERSISTENT	X	X	0



For the designations, that is the naming of variables, it should be taken into consideration that variables cannot have empty spaces or umlauts, they cannot be declared twice, or be identical to key words.

No distinction is made between upper and lower case spelling of variable names, meaning that VAR1, Var1 and var1 are all the same variable.

Underscores are significant in designation, e.g. "A_BCD" and "AB_CD" are interpreted as different designations.

Consecutive underscores are prohibited at the beginning of a designation or within a designation. The designation length as well as the significant range is unlimited (max. 255 characters) for all practical purposes.