Recommendations on the naming of identifiers

Identifiers are defined at the declaration of variables (Variable names), user-defined data types and at the creation of POUs (functions, function blocks, programs) and visualizations. You might follow the following recommendations concerning the naming of identifiers in order to make it as unique as possible.

(1) Variable names

The naming of variables in applications and libraries as far as possible should follow the Hungarian notation:

For each variable a meaningful, short description should be found, the **base name**. The first letter of each word of a base name should be a capital letter, the others should be small ones (Example: FileSize). If needed additionally an translation file for other languages can be created.

Before the base name, corresponding to the data type of the variable, **prefix(es)** are added in small letters.

Data type	***lower limit***	***upper limit***	***Information content***	***Prefix***	***Comment***
BOOL	FALSE	TRUE	1 Bit	x*	
				b	reserved
BYTE			8 Bit	by	Bit string, not for arithm. operations
WORD			16 Bit	w	Bit string, not for arithm. operations
DWORD			32 Bit	dw	Bit string, not for arithm. operations
LWORD			64 Bit	lw	not for arithm. operations
SINT	-128	127	8 Bit	si	
USINT	0	255	8 Bit	usi	
INT	-32.768	32.767	16 Bit	i	
UINT	0	65.535	16 Bit	ui	
DINT	-2.147.483.648	2.147.483.647	32 Bit	di	
UDINT	0	4.294.967.295	32 Bit	udi	
LINT	-2^{63}	$2^{63} - 1$	64 Bit	li	
ULINT	0	$2^{64} - 1$	64 Bit	uli	
REAL			32 Bit	r	
LREAL			64 Bit	lr	
STRING				s	
TIME				tim	
TIME_OF_DAY				tod	
DATETIME				dt	
DATE				date	
ENUM			16 Bit	e	
POINTER				p	
ARRAY				a	

* pointedly for BOOLean variables x is chosen as prefix, in order to differentiate from BYTE and also in order to accommodate the perception of an IEC-programmer (see addressing %IX0.0).

Examples:

```
bySubIndex: BYTE;
sFileName: STRING;
```