Core Virtual Machine Functions

Version: **2.6.0** from date **2009.07-13 21:09**

Extended types

Type	Implementation	Details	Comment
USINT	alias	BYTE	Type USINT (for compability with IEC standard is alias for BYTE)
UINT	alias	WORD	Type UINT (for compability with IEC standard is alias for WORD)
UDINT	alias	DWORD	Type UDINT (for compability with IEC standard is alias for DWORD)
ULINT	alias	LWORD	Type ULINT (for compability with IEC standard is alias for LWORD)

Removed elementary types

No	Type	Comment

Dependent files

Type	Order	Name

Implemented function

Code	Name:Type	Arguments	Description
01*1	ADD:SINT	* summand0:SINT	Adds two or more SINT operands
01*2	ADD:INT	* summand0:INT	Adds two or more INT operands
01*3	ADD:DINT	* summand0:DINT	Adds two or more DINT operands
01*5	ADD:BYTE	* summand0:BYTE	Adds two or more BYTE operands
01*6	ADD:WORD	* summand0:WORD	Adds two or more WORD operands
01*7	ADD:DWORD	* summand0:DWORD	Adds two or more DWORD operands
01*9	ADD:REAL	* summand0:REAL	Adds two or more REAL operands
0201	SUB:SINT	0 minuend:SINT 1 subtrahend:SINT	Calculates subtract between first and second argument
0202	SUB:INT	0 minuend:INT 1 subtrahend:INT	Calculates subtract between first and second argument
0203	SUB:DINT	0 minuend:DINT 1 subtrahend:DINT	Calculates subtract between first and second argument
0205	SUB:BYTE	0 minuend:BYTE 1 subtrahend:BYTE	Calculates subtract between first and second argument
0206	SUB:WORD	0 minuend:WORD 1 subtrahend:WORD	Calculates subtract between first and second argument
0207	SUB:DWORD	0 minuend:DWORD 1 subtrahend:DWORD	Calculates subtract between first and second argument
0209	SUB:REAL	0 minuend:REAL 1 subtrahend:REAL	Calculates subtract between first and second argument
020B	SUB:TIME	0 minuend:TIME 1 subtrahend:TIME	Calculates subtract between first and second argument
03*1	MUL:SINT	* factor0:SINT	Multiplies two or more SINT factors
03*2	MUL:INT	* factor0:INT	Multiplies two or more INT factors
03*3	MUL:DINT	* factor0:DINT	Multiplies two or more DINT factors

03*5	MUL:BYTE	* factor0:BYTE	Multiplies two or more BYTE factors
03*6	MUL:WORD	* factor0:WORD	Multiplies two or more WORD factors
03*7	MUL:DWORD	* factor0:DWORD	Multiplies two or more DWORD factors
03*9	MUL:REAL	* factor0:REAL	Multiplies two or more REAL factors
0401	DIV:SINT	0 dividend:SINT 1 divisor:SINT	Divides dividend by divisor
0402	DIV:INT	0 dividend:INT 1 divisor:INT	Divides dividend by divisor
0403	DIV:DINT	0 dividend:DINT 1 divisor:DINT	Divides dividend by divisor
0405	DIV:BYTE	0 dividend:BYTE 1 divisor:BYTE	Divides dividend by divisor
0406	DIV:WORD	0 dividend:WORD 1 divisor:WORD	Divides dividend by divisor
0407	DIV:DWORD	0 dividend:DWORD 1 divisor:DWORD	Divides dividend by divisor
0409	DIV:REAL	0 dividend:REAL 1 divisor:REAL	Divides dividend by divisor
0411	MOD:SINT	0 dividend:SINT 1 divisor:SINT	Remainder of division dividend by divisor
0412	MOD:INT	0 dividend:INT 1 divisor:INT	Remainder of division dividend by divisor
0413	MOD:DINT	0 dividend:DINT 1 divisor:DINT	Remainder of division dividend by divisor
0415	MOD:BYTE	0 dividend:BYTE 1 divisor:BYTE	Remainder of division dividend by divisor
0416	MOD:WORD	0 dividend:WORD 1 divisor:WORD	Remainder of division dividend by divisor
0417	MOD:DWORD	0 dividend:DWORD 1 divisor:DWORD	Remainder of division dividend by divisor
0501	NEG:SINT	0 INP:SINT	Changes sign of the signed value
0502	NEG:INT	0 INP:INT	Changes sign of the signed value
0503	NEG:DINT	0 INP:DINT	Changes sign of the signed value
0507	NEG:REAL	0 INP:REAL	Changes sign of the signed value
0510	NOT:BOOL	0 INP:BOOL	Binary negation
0511	NOT:BYTE	0 INP:BYTE	Bitwise negation of unsigned value
0512	NOT:WORD	0 INP:WORD	Bitwise negation of unsigned value
0513	NOT:DWORD	0 INP:DWORD	Bitwise negation of unsigned value
0601	EXPT:DINT	0 X:SINT 1 Y:SINT	Returns X raised to the specified power (Y)
0602	EXPT:DINT	0 X:INT 1 Y:SINT	Returns X raised to the specified power (Y)
0603	EXPT:DINT	0 X:DINT 1 Y:SINT	Returns X raised to the specified power (Y)
0604	EXPT:REAL	0 X:REAL 1 Y:SINT	Returns X raised to the specified power (Y)
0605	EXPT:REAL	0 X:REAL 1 Y:INT	Returns X raised to the specified power (Y)
0606	EXPT:REAL	0 X:REAL 1 Y:DINT	Returns X raised to the specified power (Y)
0607	EXPT:REAL	0 X:REAL 1 Y:REAL	Returns X raised to the specified power (Y)
0611	ABS:SINT	0 INP:SINT	Returns the absolute value of a specified number

0612	ABS:INT	0 INP:INT	Returns the absolute value of a specified number
0613	ABS:DINT	0 INP:DINT	Returns the absolute value of a specified number
0615	ABS:BYTE	0 INP:BYTE	Returns the absolute value of a specified number
0616	ABS:WORD	0 INP:WORD	Returns the absolute value of a specified number
0617	ABS:DWORD	0 INP:DWORD	Returns the absolute value of a specified number
0619	ABS:REAL	0 INP:REAL	Returns the absolute value of a specified number
0620	SQRT:REAL	0 R:REAL	Returns the square root of a specified number
0622	LN:REAL	0 R:REAL	Returns the natural (base e) logarithm of a specified number
0624	LOG:REAL	0 R:REAL	Returns the base 10 logarithm of a specified number
0626	EXP:REAL	0 Y:REAL	Returns e raised to the specified power (Y)
0628	SIN:REAL	0 A:REAL	A - An angle, measured in radians Returns the sine of the specified angle
062A	COS:REAL	0 A:REAL	A - An angle, measured in radians Returns the cosine of the specified angle
062C	TAN:REAL	0 A:REAL	A - An angle, measured in radians Returns the tangent of the specified angle
062E	ASIN:REAL	0 R:REAL	R - A number representing a sine, where -1 <= R <= 1 Returns the angle PHI (measured in radians, such that -PI/2 <= PHI <= PI/2) whose sine is the specified number.
0630	ACOS:REAL	0 R:REAL	R - A number representing a sine, where -1 <= R <= 1 Returns the angle PHI (measured in radians, such that 0 <= PHI <= PI) whose cosine is the specified number.
0632	ATAN:REAL	0 R:REAL	R - A number representing a tangent Returns the angle PHI (measured in radians, such that -PI/2 <= PHI <= PI/2) whose tangent is the specified number.
0634	TRUNC:DINT	0 R:REAL	Calculates the integral part of REAL number as DINT value
0636	ROUND:DINT	0 R:REAL	Rounds a value to the nearest integer
08*0	AND:BOOL	* arg0:BOOL	Binary AND of BOOL operands
08*1	AND:BYTE	* arg0:BYTE	Bitwise AND of BYTE operands
08*2	AND:WORD	* arg0:WORD	Bitwise AND of WORD operands
08*3	AND:DWORD	* arg0:DWORD	Bitwise AND of DWORD operands
09*0	OR:BOOL	* arg0:BOOL	Binary OR of BOOL operands
09*1	OR:BYTE	* arg0:BYTE	Bitwise OR of BYTE operands
09*2	OR:WORD	* arg0:WORD	Bitwise OR of WORD operands

09*3	OR:DWORD	* arg0:DWORD	Bitwise OR of DWORD operands
0A*0	XOR:BOOL	* arg0:BOOL	Binary XOR of BOOL operands
0A*1	XOR:BYTE	* arg0:BYTE	Bitwise XOR of BYTE operands
0A*2	XOR:WORD	* arg0:WORD	Bitwise XOR of WORD operands
0A*4	XOR:DWORD	* arg0:DWORD	Bitwise XOR of DWORD operands
0B01	SHL:BYTE	0 arg:BYTE 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B02	SHL:WORD	0 arg:WORD 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B03	SHL:DWORD	0 arg:DWORD 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B11	SHR:BYTE	0 arg:BYTE 1 num:INT	Shifts first argument right by the number of bits specified by second argument
0B12	SHR:WORD	0 arg:WORD 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B13	SHR:DWORD	0 arg:DWORD 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B21	ROL:BYTE	0 arg:BYTE 1 num:INT	Rotate the input values to the left to the most significant bit (MSB) by a specified number of bit positions
0B22	ROL:WORD	0 arg:WORD 1 num:INT	Rotate the input values to the left to the most significant bit (MSB) by a specified number of bit positions
0B23	ROL:DWORD	0 arg:DWORD 1 num:INT	Rotate the input values to the left to the most significant bit (MSB) by a specified number of bit positions
0B31	ROR:BYTE	0 arg:BYTE 1 num:INT	Rotate the input values to the right to the least significant bit (LSB) by a specified number of bit positions
0B32	ROR:WORD	0 arg:WORD 1 num:INT	Rotate the input values to the right to the least significant bit (LSB) by a specified number of bit positions
0B33	ROR:DWORD	0 arg:DWORD 1 num:INT	Rotate the input values to the right to the least significant bit (LSB) by a specified number of bit positions
0C00	SEL:BOOL	0 selector:BOOL 1 case_false:BOOL 2 case_true:BOOL	Selects one of two arguments
0C01	SEL:SINT	0 selector:BOOL 1 case_false:SINT 2 case_true:SINT	Selects one of two arguments
0C02	SEL:INT	0 selector:BOOL 1 case_false:INT 2 case_true:INT	Selects one of two arguments
0C03	SEL:DINT	0 selector:BOOL 1 case_false:DINT 2 case_true:DINT	Selects one of two arguments
0C05	SEL:BYTE	0 selector:BOOL 1 case_false:BYTE 2 case_true:BYTE	Selects one of two arguments

0C06	SEL:WORD	0 selector:BOOL 1 case_false:WORD 2 case_true:WORD	Selects one of two arguments
0C07	SEL:DWORD	0 selector:BOOL 1 case_false:DWORD 2 case_true:DWORD	Selects one of two arguments
0C09	SEL:REAL	0 selector:BOOL 1 case_false:REAL 2 case_true:REAL	Selects one of two arguments
0C0B	SEL:TIME	0 selector:BOOL 1 case_false:TIME 2 case_true:TIME	Selects one of two arguments
0C0C	SEL:DATE	0 selector:BOOL 1 case_false:DATE 2 case_true:DATE	Selects one of two arguments
0C0D	SEL:TIME_OF_DAY	0 selector:BOOL 1 case_false:TIME_OF_DAY 2 case_true:TIME_OF_DAY	Selects one of two arguments
0C0E	SEL:DATE_AND_TIME	0 selector:BOOL 1 case_false:DATE_AND_TIME 2 case_true:DATE_AND_TIME	Selects one of two arguments
0D00	LIMIT:BOOL	0 value:BOOL 1 min:BOOL 2 max:BOOL	Limits the value between two bounds
0D01	LIMIT:SINT	0 value:SINT 1 min:SINT 2 max:SINT	Limits the value between two bounds
0D02	LIMIT:INT	0 value:INT 1 min:INT 2 max:INT	Limits the value between two bounds
0D03	LIMIT:DINT	0 value:DINT 1 min:DINT 2 max:DINT	Limits the value between two bounds
0D05	LIMIT:BYTE	0 value:BYTE 1 min:BYTE 2 max:BYTE	Limits the value between two bounds
0D06	LIMIT:WORD	0 value:WORD 1 min:WORD 2 max:WORD	Limits the value between two bounds
0D07	LIMIT:DWORD	0 value:DWORD 1 min:DWORD 2 max:DWORD	Limits the value between two bounds
0D09	LIMIT:REAL	0 value:REAL 1 min:REAL 2 max:REAL	Limits the value between two bounds
0D0B	LIMIT:TIME	0 value:TIME 1 min:TIME 2 max:TIME	Limits the value between two bounds
0D0C	LIMIT:DATE	0 value:DATE 1 min:DATE 2 max:DATE	Limits the value between two bounds
0D0D	LIMIT:TIME_OF_DAY	0 value:TIME_OF_DAY 1 min:TIME_OF_DAY 2 max:TIME_OF_DAY	Limits the value between two bounds
0D0E	LIMIT:DATE_AND_TIME	0 value:DATE_AND_TIME 1 min:DATE_AND_TIME 2 max:DATE_AND_TIME	Limits the value between two bounds
0E00	MAX:BOOL	0 in0:BOOL 1 in1:BOOL	Selects maximum between two values
0E01	MAX:SINT	0 in0:SINT 1 in1:SINT	Selects maximum between two values
0E02	MAX:INT	0 in0:INT 1 in1:INT	Selects maximum between two values

0E03	MAX:DINT	0 in0:DINT 1 in1:DINT	Selects maximum between two values
0E05	MAX:BYTE	0 in0:BYTE 1 in1:BYTE	Selects maximum between two values
0E06	MAX:WORD	0 in0:WORD 1 in1:WORD	Selects maximum between two values
0E07	MAX:DWORD	0 in0:DWORD 1 in1:DWORD	Selects maximum between two values
0E09	MAX:REAL	0 in0:REAL 1 in1:REAL	Selects maximum between two values
0E0B	MAX:TIME	0 in0:TIME 1 in1:TIME	Selects maximum between two values
0E0C	MAX:DATE	0 in0:DATE 1 in1:DATE	Selects maximum between two values
0E0D	MAX:TIME_OF_DAY	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Selects maximum between two values
0E0E	MAX:DATE_AND_TIME	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Selects maximum between two values
0F00	MIN:BOOL	0 in0:BOOL 1 in1:BOOL	Selects minimum between two values
0F01	MIN:SINT	0 in0:SINT 1 in1:SINT	Selects minimum between two values
0F02	MIN:INT	0 in0:INT 1 in1:INT	Selects minimum between two values
0F03	MIN:DINT	0 in0:DINT 1 in1:DINT	Selects minimum between two values
0F05	MIN:BYTE	0 in0:BYTE 1 in1:BYTE	Selects minimum between two values
0F06	MIN:WORD	0 in0:WORD 1 in1:WORD	Selects minimum between two values
0F07	MIN:DWORD	0 in0:DWORD 1 in1:DWORD	Selects minimum between two values
0F09	MIN:REAL	0 in0:REAL 1 in1:REAL	Selects minimum between two values
0F0B	MIN:TIME	0 in0:TIME 1 in1:TIME	Selects minimum between two values
0F0C	MIN:DATE	0 in0:DATE 1 in1:DATE	Selects minimum between two values
0F0D	MIN:TIME_OF_DAY	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Selects minimum between two values
0F0E	MIN:DATE_AND_TIME	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Selects minimum between two values
1000	GT:BOOL	0 in0:BOOL 1 in1:BOOL	Checks if first argument is grater than second
1001	GT:BOOL	0 in0:SINT 1 in1:SINT	Checks if first argument is grater than second
1002	GT:BOOL	0 in0:INT 1 in1:INT	Checks if first argument is grater than second
1003	GT:BOOL	0 in0:DINT 1 in1:DINT	Checks if first argument is grater than second
1005	GT:BOOL	0 in0:BYTE 1 in1:BYTE	Checks if first argument is grater than second
1006	GT:BOOL	0 in0:WORD 1 in1:WORD	Checks if first argument is grater than second
1007	GT:BOOL	0 in0:DWORD 1 in1:DWORD	Checks if first argument is grater than second
1009	GT:BOOL	0 in0:REAL 1 in1:REAL	Checks if first argument is grater than second
100B	GT:BOOL	0 in0:TIME 1 in1:TIME	Checks if first argument is grater than second
100C	GT:BOOL	0 in0:DATE 1 in1:DATE	Checks if first argument is grater than second

100D	GT:BOOL	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Checks if first argument is grater than second
100E	GT:BOOL	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Checks if first argument is grater than second
1100	GE:BOOL	0 in0:BOOL 1 in1:BOOL	Checks if first argument is grater or equal than second
1101	GE:BOOL	0 in0:SINT 1 in1:SINT	Checks if first argument is grater or equal than second
1102	GE:BOOL	0 in0:INT 1 in1:INT	Checks if first argument is grater or equal than second
1103	GE:BOOL	0 in0:DINT 1 in1:DINT	Checks if first argument is grater or equal than second
1105	GE:BOOL	0 in0:BYTE 1 in1:BYTE	Checks if first argument is grater or equal than second
1106	GE:BOOL	0 in0:WORD 1 in1:WORD	Checks if first argument is grater or equal than second
1107	GE:BOOL	0 in0:DWORD 1 in1:DWORD	Checks if first argument is grater or equal than second
1109	GE:BOOL	0 in0:REAL 1 in1:REAL	Checks if first argument is grater or equal than second
110B	GE:BOOL	0 in0:TIME 1 in1:TIME	Checks if first argument is grater or equal than second
110C	GE:BOOL	0 in0:DATE 1 in1:DATE	Checks if first argument is grater or equal than second
110D	GE:BOOL	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Checks if first argument is grater or equal than second
110E	GE:BOOL	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Checks if first argument is grater or equal than second
1200	EQ:BOOL	0 in0:BOOL 1 in1:BOOL	Checks if first argument is equal to second
1201	EQ:BOOL	0 in0:SINT 1 in1:SINT	Checks if first argument is equal to second
1202	EQ:BOOL	0 in0:INT 1 in1:INT	Checks if first argument is equal to second
1203	EQ:BOOL	0 in0:DINT 1 in1:DINT	Checks if first argument is equal to second
1205	EQ:BOOL	0 in0:BYTE 1 in1:BYTE	Checks if first argument is equal to second
1206	EQ:BOOL	0 in0:WORD 1 in1:WORD	Checks if first argument is equal to second
1207	EQ:BOOL	0 in0:DWORD 1 in1:DWORD	Checks if first argument is equal to second
1209	EQ:BOOL	0 in0:REAL 1 in1:REAL	Checks if first argument is equal to second
120B	EQ:BOOL	0 in0:TIME 1 in1:TIME	Checks if first argument is equal to second
120C	EQ:BOOL	0 in0:DATE 1 in1:DATE	Checks if first argument is equal to second
120D	EQ:BOOL	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Checks if first argument is equal to second
120E	EQ:BOOL	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Checks if first argument is equal to second
1300	LE:BOOL	0 in0:BOOL 1 in1:BOOL	Checks if first argument is less or equal than second
1301	LE:BOOL	0 in0:SINT 1 in1:SINT	Checks if first argument is less or equal than second
1302	LE:BOOL	0 in0:INT 1 in1:INT	Checks if first argument is less or equal than second
1303	LE:BOOL	0 in0:DINT 1 in1:DINT	Checks if first argument is less or equal than second
1305	LE:BOOL	0 in0:BYTE 1 in1:BYTE	Checks if first argument is less or equal than second

1306	LE:BOOL	0 in0:WORD 1 in1:WORD	Checks if first argument is less or equal than second
1307	LE:BOOL	0 in0:DWORD 1 in1:DWORD	Checks if first argument is less or equal than second
1309	LE:BOOL	0 in0:REAL 1 in1:REAL	Checks if first argument is less or equal than second
130B	LE:BOOL	0 in0:TIME 1 in1:TIME	Checks if first argument is less or equal than second
130C	LE:BOOL	0 in0:DATE 1 in1:DATE	Checks if first argument is less or equal than second
130D	LE:BOOL	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Checks if first argument is less or equal than second
130E	LE:BOOL	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Checks if first argument is less or equal than second
1400	LT:BOOL	0 in0:BOOL 1 in1:BOOL	Checks if first argument is less than second
1401	LT:BOOL	0 in0:SINT 1 in1:SINT	Checks if first argument is less than second
1402	LT:BOOL	0 in0:INT 1 in1:INT	Checks if first argument is less than second
1403	LT:BOOL	0 in0:DINT 1 in1:DINT	Checks if first argument is less than second
1405	LT:BOOL	0 in0:BYTE 1 in1:BYTE	Checks if first argument is less than second
1406	LT:BOOL	0 in0:WORD 1 in1:WORD	Checks if first argument is less than second
1407	LT:BOOL	0 in0:DWORD 1 in1:DWORD	Checks if first argument is less than second
1409	LT:BOOL	0 in0:REAL 1 in1:REAL	Checks if first argument is less than second
140B	LT:BOOL	0 in0:TIME 1 in1:TIME	Checks if first argument is less than second
140C	LT:BOOL	0 in0:DATE 1 in1:DATE	Checks if first argument is less than second
140D	LT:BOOL	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Checks if first argument is less than second
140E	LT:BOOL	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Checks if first argument is less than second
1500	NE:BOOL	0 in0:BOOL 1 in1:BOOL	Checks if first argument is not equal to second
1501	NE:BOOL	0 in0:SINT 1 in1:SINT	Checks if first argument is not equal to second
1502	NE:BOOL	0 in0:INT 1 in1:INT	Checks if first argument is not equal to second
1503	NE:BOOL	0 in0:DINT 1 in1:DINT	Checks if first argument is not equal to second
1505	NE:BOOL	0 in0:BYTE 1 in1:BYTE	Checks if first argument is not equal to second
1506	NE:BOOL	0 in0:WORD 1 in1:WORD	Checks if first argument is not equal to second
1507	NE:BOOL	0 in0:DWORD 1 in1:DWORD	Checks if first argument is not equal to second
1509	NE:BOOL	0 in0:REAL 1 in1:REAL	Checks if first argument is not equal to second
150B	NE:BOOL	0 in0:TIME 1 in1:TIME	Checks if first argument is not equal to second
150C	NE:BOOL	0 in0:DATE 1 in1:DATE	Checks if first argument is not equal to second
150D	NE:BOOL	0 in0:TIME_OF_DAY 1 in1:TIME_OF_DAY	Checks if first argument is not equal to second
150E	NE:BOOL	0 in0:DATE_AND_TIME 1 in1:DATE_AND_TIME	Checks if first argument is not equal to second

16*0	MUX:BOOL	0 in0:INT * in1:BOOL	Selects one of the values
16*1	MUX:SINT	0 in0:INT * in1:SINT	Selects one of the values
16*2	MUX:INT	0 in0:INT * in1:INT	Selects one of the values
16*3	MUX:DINT	0 in0:INT * in1:DINT	Selects one of the values
16*5	MUX:BYTE	0 in0:INT * in1:BYTE	Selects one of the values
16*6	MUX:WORD	0 in0:INT * in1:WORD	Selects one of the values
16*7	MUX:DWORD	0 in0:INT * in1:DWORD	Selects one of the values
16*9	MUX:REAL	0 in0:INT * in1:REAL	Selects one of the values
16*B	MUX:TIME	0 in0:INT * in1:TIME	Selects one of the values
16*C	MUX:DATE	0 in0:INT * in1:DATE	Selects one of the values
16*D	MUX:TIME_OF_DAY	0 in0:INT * in1:TIME_OF_DAY	Selects one of the values
16*E	MUX:DATE_AND_TIME	0 in0:INT * in1:DATE_AND_TIME	Selects one of the values
1900	INT_TO_REAL:REAL	0 INP:INT	Converts INT value to REAL value
1901	DINT_TO_REAL:REAL	0 INP:DINT	Converts DINT value to REAL value
1902	TIME_TO_DINT:DINT	0 INP:TIME	Returns number represented by TIME
1903	DINT_TO_TIME:TIME	0 INP:DINT	Returns TIME represented by number
1904	TIME_TO_REAL:REAL	0 INP:TIME	Returns REAL number represented by TIME. Superposition of conversions TIME->DINT->REAL
1905	REAL_TO_TIME:TIME	0 INP:REAL	Returns TIME represented by REAL number. Superposition of conversions REAL->DINT->TIME
1906	BCD_TO_INT:INT	0 INP:BYTE	Converts BCD stored at BYTE to INT value
1907	BCD_TO_INT:INT	0 INP:WORD	Converts BCD stored at WORD to INT value
1908	INT_TO_BYTE_BCD:BYTE	0 INP:INT	Converts INT value to BCD stored at BYTE
1909	INT_TO_WORD_BCD:WORD	0 INP:INT	Converts INT value to BCD stored at WORD
190A	REAL_TO_INT:INT	0 INP:REAL	Converts REAL to INT value with truncation
190B	INT_TO_BOOL:BOOL	0 INP:INT	Converts BOOL to INT value
190C	INT_TO_DINT:DINT	0 INP:INT	Converts INT to DINT value
1C17	CUR_TIME:TIME		Returns current system time
1C20	READ_RTC:DATE_AND_TIME		Returns current date and time from RTC clock
1C21	WRITE_RTC:BOOL	0 TIME_TO_SAVE:DATE_AND_TIME	Sets the RTC clock. Returns clock update status.
1C30	GET_TST:DATE_AND_TIME		Returns task cycle start time.
1C22	RANDOML:REAL		Returns random value with linear probability with range 0 to 1
			Returns value of status word 1. Meaning of the bitwise mask:

1C2F	GET_STATUS_WORD1:WORD		16#01 - Last cycle execution with overrun. 16#02 - Access to array index was over bounds. 16#04 - If set then its mean initial (cold) statup (1). If reset (0) then normal execution. 16#08 - If set then its mean first startup after uploading xcp code.
1C32	GET_VMACH_VERSION:WORD		Returns current virtual machine number in nibble Little Endian order (ie. 2.5.6.7 as 0x6725).
1C31	TASK_CYCLE:TIME		Task interval of current task
01*4	ADD:LINT	* summand0:LINT	Adds two or more LINT operands
01*8	ADD:LWORD	* summand0:LWORD	Adds two or more LWORD operands
01*B	ADD:TIME	* summand0:TIME	Adds two or more TIME operands
0204	SUB:LINT	0 minuend:LINT 1 subtrahend:LINT	Calculates subtract between first and second argument
0208	SUB:LWORD	0 minuend:LWORD 1 subtrahend:LWORD	Calculates subtract between first and second argument
03*4	MUL:LINT	* factor0:LINT	Multiplies two or more LINT factors
03*8	MUL:LWORD	* factor0:LWORD	Multiplies two or more LWORD factors
0404	DIV:LINT	0 dividend:LINT 1 divisor:LINT	Divides dividend by divisor
0408	DIV:LWORD	0 dividend:LWORD 1 divisor:LWORD	Divides dividend by divisor
0414	MOD:LINT	0 dividend:LINT 1 divisor:LINT	Remainder of division dividend by divisor
0418	MOD:LWORD	0 dividend:LWORD 1 divisor:LWORD	Remainder of division dividend by divisor
0504	NEG:LINT	0 in0:LINT	Changes sign of the signed value
0514	NOT:LWORD	0 in0:LWORD	Bitwise negation of unsigned value
0614	ABS:LINT	0 in0:LINT	Returns the absolute value of a specified number
0618	ABS:LWORD	0 in0:LWORD	Returns the absolute value of a specified number
08*4	AND:LWORD	* in0:LWORD	Bitwise AND of LWORD operands
09*4	OR:LWORD	* in0:LWORD	Bitwise OR of LWORD operands
0A*5	XOR:LWORD	* in0:LWORD	Bitwise XOR of LWORD operands
0B04	SHL:LWORD	0 arg:LWORD 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B14	SHR:LWORD	0 arg:LWORD 1 num:INT	Shifts first argument left by the number of bits specified by second argument
0B24	ROL:LWORD	0 arg:LWORD 1 num:INT	Rotate the input values to the left to the most significant bit (MSB) by a specified number of bit positions
0B34	ROR:LWORD	0 arg:LWORD 1 num:INT	Rotate the input values to the right to the least significant bit (LSB) by a specified number of bit positions
0C04	SEL:LINT	0 selector:BOOL 1 case_false:LINT 2 case_true:LINT	Selects one of two arguments

0C08	SEL:LWORD	0 selector:BOOL 1 case_false:LWORD 2 case_true:LWORD	Selects one of two arguments
0D04	LIMIT:LINT	0 in0:LINT 1 in1:LINT 2 in2:LINT	Limits the value between two bounds
0D08	LIMIT:LWORD	0 in0:LWORD 1 in1:LWORD 2 in2:LWORD	Limits the value between two bounds
0E04	MAX:LINT	0 in0:LINT 1 in1:LINT	Selects maximum between two values
0E08	MAX:LWORD	0 in0:LWORD 1 in1:LWORD	Selects maximum between two values
0F04	MIN:LINT	0 in0:LINT 1 in1:LINT	Selects minimum between two values
0F08	MIN:LWORD	0 in0:LWORD 1 in1:LWORD	Selects minimum between two values
1004	GT:BOOL	0 in0:LINT 1 in1:LINT	Checks if first argument is grater than second
1008	GT:BOOL	0 in0:LWORD 1 in1:LWORD	Checks if first argument is grater than second
1104	GE:BOOL	0 in0:LINT 1 in1:LINT	Checks if first argument is grater or equal than second
1108	GE:BOOL	0 in0:LWORD 1 in1:LWORD	Checks if first argument is grater or equal than second
1204	EQ:BOOL	0 in0:LINT 1 in1:LINT	Checks if first argument is equal to second
1208	EQ:BOOL	0 in0:LWORD 1 in1:LWORD	Checks if first argument is equal to second
1304	LE:BOOL	0 in0:LINT 1 in1:LINT	Checks if first argument is less or equal than second
1308	LE:BOOL	0 in0:LWORD 1 in1:LWORD	Checks if first argument is less or equal than second
1404	LT:BOOL	0 in0:LINT 1 in1:LINT	Checks if first argument is less than second
1408	LT:BOOL	0 in0:LWORD 1 in1:LWORD	Checks if first argument is less than second
1504	NE:BOOL	0 in0:LINT 1 in1:LINT	Checks if first argument is not equal to second
1508	NE:BOOL	0 in0:LWORD 1 in1:LWORD	Checks if first argument is not equal to second
16*4	MUX:LINT	0 in0:INT * in1:LINT	Selects one of the values
16*8	MUX:LWORD	0 in0:INT * in1:LWORD	Selects one of the values
0640	GET_DAYOFWEEK:INT	0 PDATE:DATE	Returns day of week with DATE argument. 0 - Sunday, 1 - Monday, 2 - Tuesday, 3 - Wednesday, 4 - Thursday, 5 - Friday, 6 - Saturday.
0641	GET_DAYOFWEEK:INT	0 PDATETIME:DATE_AND_TIME	Returns day of week with DATE_AND_TIME argument. 0 - Sunday, 1 - Monday, 2 - Tuesday, 3 - Wednesday, 4 - Thursday, 5 - Friday, 6 - Saturday.

Specific *in-line* functions

Name:Type	Arguments	Description

Specific function blocks

Name	Input parameters	Output parameters	Description

Export date: **2009.08-17 15:01:56**

Virtual machine little endian in line functions

Version: **2.6.0** from date **2009.07-13 21:20**

Extended types

Type	Implementation	Details	Comment

Removed elementary types

No	Type	Comment

Dependent files

Type	Order	Name

Implemented function

Code	Name:Type	Arguments	Description

Specific *in-line* functions

Name:Type	Arguments	Description
GET_YEAR:INT	0 PDATE_TIME:DATE_AND_TIME	Returns year value from DATE_AND_TIME parameter
GET_YEAR:INT	0 PDATE:DATE	Returns year value from DATE parameter
GET_MONTH:INT	0 PDATE_TIME:DATE_AND_TIME	Returns month value from DATE_AND_TIME parameter
GET_MONTH:INT	0 PDATE:DATE	Returns month value from DATE parameter
GET_DAY:INT	0 PDATE:DATE	Returns day value from DATE parameter
GET_DAY:INT	0 PDATETIME:DATE_AND_TIME	Returns day value from DATE_AND_TIME parameter
GET_HOUR:INT	0 PTIMEOFDAY:TIME_OF_DAY	Returns hour value from TIME_OF_DAY parameter
GET_HOUR:INT	0 PDATETIME:DATE_AND_TIME	Returns hour value from DATE_AND_TIME parameter
GET_MINUTE:INT	0 PTIMEOFDAY:TIME_OF_DAY	Returns minute value from TIME_OF_DAY parameter
GET_MINUTE:INT	0 PDATETIME:DATE_AND_TIME	Returns minute value from DATE_AND_TIME parameter
GET_SECOND:INT	0 PTIMEOFDAY:TIME_OF_DAY	Returns second value from TIME_OF_DAY parameter
GET_SECOND:INT	0 PDATETIME:DATE_AND_TIME	Returns second value from DATE_AND_TIME parameter
GET_HUNDSEC:INT	0 PTIMEOFDAY:TIME_OF_DAY	Returns 1/100 second value from TIME_OF_DAY parameter
GET_HUNDSEC:INT	0 PDATETIME:DATE_AND_TIME	Returns 1/100 second value from DATE_AND_TIME parameter
DT_TO_TOD:TIME_OF_DAY	0 PDATETIME:DATE_AND_TIME	Extracts TIME_OF_DAY from DATE_AND_TIME parameter
BOOL_TO_INT:INT	0 INP:BOOL	Conversion BOOL parameter to INT value
DEPR_INT_TO_DINT:DINT	0 INP:INT	[Depreciate] Converts INT parameter to DINT value by duplicating the most significant bit
DINT_TO_INT:INT	0 INP:DINT	Converts DINT parameter to INT value by omitting the most significant bits

BYTE_TO_SINT:SINT	0 INP:BYTE	Converts BYTE parameter to SINT value without affection
SINT_TO_BYTE:BYTE	0 INP:SINT	Converts SINT parameter to BYTE value without affection
WORD_TO_INT:INT	0 INP:WORD	Converts WORD parameter to INT value without affection
INT_TO_WORD:WORD	0 INP:INT	Converts INT parameter to WORD value without affection
DWORD_TO_DINT:DINT	0 INP:DWORD	Converts DWORD parameter to DINT value without affection
DINT_TO_DWORD:DWORD	0 INP:DINT	Converts DWORD parameter to DINT value without affection
LWORD_TO_LINT:LINT	0 INP:LWORD	Converts LWORD parameter to LINT value without affection
LINT_TO_LWORD:LWORD	0 INP:LINT	Converts LINT parameter to LWORD value without affection
DINT_TO_LINT:LINT	0 INP:DINT	Converts DINT parameter to LINT value by duplicating the most significant bit
LINT_TO_DINT:DINT	0 INP:LINT	Converts LINT parameter to DINT value by omitting the most significant bits
DWORD_TO_LWORD:LWORD	0 INP:DWORD	Converts DWORD parameter to LWORD value by filling the most significant bits with zeroes
LWORD_TO_DWORD:DWORD	0 INP:LWORD	Converts LWORD parameter to DWORD value by omitting the most significant bits
WORD_TO_DWORD:DWORD	0 INP:WORD	Converts WORD parameter to DWORD value by filling the most significant bits with zeroes
WORD_TO_LWORD:LWORD	0 INP:WORD	Converts WORD parameter to LWORD value by filling the most significant bits with zeroes
INT_TO_LINT:LINT	0 INP:INT	Converts DINT parameter to LINT value by duplicating the most significant bit

Specific function blocks

Name	Input parameters	Output parameters	Description

Export date: **2009.08-17 15:01:56**

Specyfikacja maszyny wirtualnej dla sterownika SMC

Wersja: **2.6.0** z dnia **2009.07-09 14:14**

Rozszerzenia typów

Typ	Implementacja	Szczegóły	Komentarz

Usunięte elementarne typy

Nr	Typ	Komentarz
1	**LREAL**	
2	**STRING**	

Zależne pliki

Typ	Kolejność	Nazwa
PRE	**0**	VMCore.xml
POST	**0**	le-IF.xml

Zaimplementowane funkcje

Kod	Nazwa:Typ	Argumenty	Opis

Specyficzne funkcje *in-line*

Nazwa:Typ	Argumenty	Opis

Specyficzne bloki funkcjonalne

Nazwa	Parametry WE	Parametry WY	Opis
COM_SM4	EN:BOOL NO:BYTE TOUT:TIME OUT1:BOOL OUT2:BOOL OUT3:BOOL OUT4:BOOL OUT5:BOOL OUT6:BOOL OUT7:BOOL OUT8:BOOL	C:BOOL	Blok komunikacyjny dla SM4
COM_SM1	EN:BOOL NO:BYTE TOUT:TIME	C:BOOL IN1:REAL IN2:REAL	Blok komunikacyjny dla SM1
COM_SM2	EN:BOOL NO:BYTE TOUT:TIME	C:BOOL IN1:REAL IN2:REAL IN3:REAL IN4:REAL	Blok komunikacyjny dla SM2
COM_SM3	EN:BOOL NO:BYTE TOUT:TIME	C:BOOL IN1:BOOL IN2:BOOL	Blok komunikacyjny dla SM3
		C:BOOL IN1:BOOL	

COM_SM5	EN:BOOL NO:BYTE TOUT:TIME	IN2:BOOL IN3:BOOL IN4:BOOL IN5:BOOL IN6:BOOL IN7:BOOL IN8:BOOL	Blok komunikacyjny dla SM5
APON	R:BOOL	Q:BOOL	R - Wejście kasujące alarm Alarm restartu sterownika
ASTR	R:BOOL	Q:BOOL	R - Wejście kasujące alarm Alarm załadowania nowej konfiguracji lub utraty zmiennych RETAIN

Data eksportu: **2009.08-17 15:01:56**

Functions for LREAL type support

Version: **2.6.0** from date **2009.07-13 21:24**

Extended types

Type	Implementation	Details	Comment

Removed elementary types

No	Type	Comment

Dependent files

Type	Order	Name

Implemented function

Code	Name:Type	Arguments	Description
190D	REAL_TO_LREAL:LREAL	0 INP:REAL	Converts REAL value to LREAL value
190E	LREAL_TO_REAL:REAL	0 INP:LREAL	Converts LREAL value to REAL value
01*A	ADD:LREAL	* summand0:LREAL	Adds two or more LREAL operands
020A	SUB:LREAL	0 minuend:LREAL 1 subtrahend:LREAL	Calculates subtract between first and second argument
03*A	MUL:LREAL	* factor0:LREAL	Multiplies two or more LREAL factors
040A	DIV:LREAL	0 dividend:LREAL 1 divisor:LREAL	Divides dividend by divisor
0509	NEG:LREAL	0 in0:LREAL	Changes sign of the signed value
061A	ABS:LREAL	0 in0:LREAL	Returns the absolute value of a specified number
0621	SQRT:LREAL	0 D:LREAL	Returns the square root of a specified number
0623	LN:LREAL	0 D:LREAL	Returns the natural (base e) logarithm of a specified number
0625	LOG:LREAL	0 D:LREAL	Returns the base 10 logarithm of a specified number
0627	EXP:LREAL	0 Y:LREAL	Returns e raised to the specified power (Y)
0629	SIN:LREAL	0 A:LREAL	A - An angle, measured in radians Returns the sine of the specified angle
062B	COS:LREAL	0 A:LREAL	A - An angle, measured in radians Returns the cosine of the specified angle
062D	TAN:LREAL	0 A:LREAL	A - An angle, measured in radians Returns the tangent of the specified angle
062F	ASIN:LREAL	0 D:LREAL	D - A number representing a sine, where -1 <= D <= 1 Returns the angle PHI (measured in radians, such that -PI/2 <= PHI <= PI/2) whose sine is the specified number.
0631	ACOS:LREAL	0 D:LREAL	D - A number representing a sine, where -1 <= D <= 1 Returns the angle PHI (measured in radians, such that 0 <= PHI <= PI) whose cosine is the specified number.
0633	ATAN:LREAL	0 D:LREAL	D - A number representing a tangent Returns the angle PHI (measured in radians, such that -PI/2 <= PHI <= PI/2) whose tangent is the specified number.
		0 D:LREAL	Calculates the integral part of LREAL number to

0635	TRUNC:LINT		LINT value
0637	ROUND:LINT	**0** in0:LREAL	Rounds a value to the nearest integer
0C0A	SEL:LREAL	**0** selector:BOOL **1** case_false:LREAL **2** case_true:LREAL	Selects one of two arguments
0D0A	LIMIT:LREAL	**0** value:LREAL **1** min:LREAL **2** max:LREAL	Limits the value between two bounds
0E0A	MAX:LREAL	**0** in0:LREAL **1** in1:LREAL	Selects maximum between two values
0F0A	MIN:LREAL	**0** in0:LREAL **1** in1:LREAL	Selects minimum between two values
100A	GT:BOOL	**0** in0:LREAL **1** in1:LREAL	Checks if first argument is grater than second
110A	GE:BOOL	**0** in0:LREAL **1** in1:LREAL	Checks if first argument is grater or equal than second
120A	EQ:BOOL	**0** in0:LREAL **1** in1:LREAL	Checks if first argument is equal to second
130A	LE:BOOL	**0** in0:LREAL **1** in1:LREAL	Checks if first argument is less or equal than second
140A	LT:BOOL	**0** in0:LREAL **1** in1:LREAL	Checks if first argument is less than second
150A	NE:BOOL	**0** in0:LREAL **1** in1:LREAL	Checks if first argument is not equal to second
16*A	MUX:LREAL	**0** in0:INT * in1:LREAL	Selects one of the values

Specific *in-line* functions

Name:Type	Arguments	Description

Specific function blocks

Name	Input parameters	Output parameters	Description

Export date: **2009.08-17 15:01:56**

Virtual Machine Specification for Praxis Automation Systems

Version: **2.6.0** from date **2009.07-13 22:32**

Extended types

Type	Implementation	Details	Comment

Removed elementary types

No	Type	Comment
1	**STRING**	

Dependent files

Type	Order	Name
PRE	**0**	VMCore.xml
POST	**0**	lreals.xml
POST	**1**	le-IF.xml

Implemented function

Code	Name:Type	Arguments	Description

Specific *in-line* functions

Name:Type	Arguments	Description

Specific function blocks

Name	Input parameters	Output parameters	Description
APON	R:BOOL	Q:BOOL	R - Reset input Warm restart alarm
ASTR	R:BOOL	Q:BOOL	R - Reset input Cold start alarm

Export date: **2009.08-17 15:01:56**

Experimental Virtual Machine Specification for University

Version: **2.6.0** from date **2009.07-26 20:22**

Extended types

Type	Implementation	Details	Comment

Removed elementary types

No	Type	Comment

Dependent files

Type	Order	Name
PRE	**0**	VMCore.xml
POST	**0**	lreals.xml
POST	**1**	le-IF.xml
POST	**2**	flash.xml

Implemented function

Code	Name:Type	Arguments	Description

Specific *in-line* functions

Name:Type	Arguments	Description

Specific function blocks

Name	Input parameters	Output parameters	Description
HW_RS	R:BOOL S:BOOL	Q:BOOL	R - Reset input S - Set input Hardware RS function block for testing purpose

Export date: **2009.08-17 15:01:57**