

3 Instructions and Functions

Overview according to groups

Program flow	Input/Output	Buffer	Numerical Functions
Subroutine	Serial	Fifo	Bitwise
SUB...END CALL RETURN	SHIFT_IN SHIFT_OUT I2C_ACK_POLL I2C_READ\$ I2C_RESULT I2C_SETUP I2C_WRITE	FIFO CLEAR_FIFO LEN_FIFO FREE_FIFO GET_FIFO READ_FIFO UNGET_FIFO PUT_FIFO PUTU_FIFO START_FIFO INTEGRAL_FIFO	BIT BIT_MIRR BITNOT CUT_BITS INV_BIT RES_BIT SET_BIT MASK IMASK SIGNEXT PARITY see also: Coding
Loops and Branches	Devices	String as buffer	Mathematical
LOOP...ENDLOOP FOR...NEXT WHILE...ENDWHILE IF...THEN ...ELSE...ENDIF GOTO SWITCH...CASE ...ENDSWITCH SWITCHI...CASE ...ENDSWITCH	INSTALL_DEVICE GET PUT INPUT INPUT_LINE PRINT PRINT_USING	NTOS\$ NFROMS RTOS\$ RFROMS STOS\$	ABS CALC_CRC EXP EXPE FIX INT LD LIMIT LN LOG MOD MODULO_INC MODULO_UPDO LIN_APPROX PRIME RND RANDOMIZE SGN SQRT
Task management	Pins and ports	Flash	
TASK...END RUN_TASK STOP_TASK CONT_TASK RESTART_PROG SET_TASK_PRIO RELEASE_TASK EXIT_TASK DISABLE_TSW ENABLE_TSW ON_ERROR_RESET ON_ERRTASK_CALL ON_ERRTASK_GOTO	DIR_PIN DIR_PORT IN OUT LL_IPOINT_AND LL_IPOINT_IN LL_IPOINT_OR LL_IPOINT_OUT LL_IPOINT_PULSE LL_IPOINT_XOR XSETUP XIN, XIN\$ XOUT XBUS_INR\$ XBUS_OUTR	PEEK_FLASH POKE_FLASH POKEM_FLASH CHECK_FLASH ERASE_FLASH SET_SERIAL_NO SERIAL_NO\$ DELETE_PROG SET_DATA_ADDR DATA LET\$	

3

Overview according to groups

3

Numerical functions	String operations	Graphic	Specials
Trigonometrical funktions	General	Pixel	Converting
ACOS ACOT ASIN QUICK_WORD_COS QUICK_WORD_SIN ATAN COS COSH COT COTH SIN SINH TAN TANH	LET\$ LEN SET_LEN\$ MAX_LEN FILL\$ LEFT\$ MID\$ RIGHT\$ REMOVE\$ REMDOUBLE\$ ADD\$ AND\$ OR\$ XOR\$ SHIFTL\$ SHIFTM\$ UPPER\$ UPPER7\$ CONVERT\$ CONVERT7\$	OR\$, AND\$, XOR\$ GRAPHIC_MASK_COPY GRAPHIC_MIRROR GRAPHIC_COPY INVERT GRAPHIC_FILL_MASK Special GRAPHIC_EXP\$ FILL_AREA Vector DRAW_LINE DRAW_NEXT_LINE CLOSE_LINE SET_BASE SET_SCALE SET_ROTATION SET_GRAREA SET_DOT DISTANCE	VAL_NUM VAL_REAL STR\$ STRI\$ CHR\$ ASC LTR LLTOR HREAL LREAL RTL SIGNEXT
	see also:		Mirror
	String as buffer Search Mirror Coding Compressing Signal processing		BIT_MIRR BYTE_MIRR BIT_MIRR\$ MIRROR\$ GRAPHIC_MIRROR
	Formatting		Data input and transmission
	FRAME TABS TRIM\$ USING PRINT_USING SPC TAB STRI\$ Format strings		DEBOUNCE DECODE SCAN_TO_CHAR_ SETUP SCAN_TO_CHAR SCAN_TO_TOGGLE see also: Compressing, Coding
			Search
			INSTR INDEX INDEX_1D INDEX_2D SELECT\$
			System
			SYSVARN SYSVAR\$ SET_SYSVARN RESTART_PROG DELETE_PROG THIS_VERS DEVEN

Overview according to groups

Specials
Compressing
PACK_BYTE\$ UNPACK_BYTE\$ ASC_BCD\$ BCD_ASC\$ ASC_PBCD\$ PBCD_ASC\$ PBCD_BCD\$ BYTE_ANIB\$ ANIB_BYTE\$ BYTE_INIB\$ INIB_BYTE\$
Time
WAIT_DURATION WAIT_NEXT WAIT_CLOCK TICKS SET_TICKS
Coding
BIT_MIX\$ BIT_DEMIX\$ BYTE_DEMIX\$ BYTE_DEMIX\$ SCRAMBLE\$ BYTE_MIX\$ CODE_1TON\$ DISTRIBUTE\$
TAN – Transaction numbers
TAN_CHK TAN_COUNT TAN_DEL TAN_FIRST

Specials
Signal processing
SIGNAL_ADD SIGNAL_AVERAGE SIGNAL_BIGGER SIGNAL_CUT_AMP SIGNAL_CUT_MAX SIGNAL_CUT_MIN SIGNAL_GAP_END SIGNAL_GAPI_END SIGNAL_MAX SIGNAL_MAX_AMPL SIGNAL_MIN SIGNAL_MUL SIGNAL_NEXT_GAP SIGNAL_NEXT_GAPI SIGNAL_OFFSET SIGNAL_SCALE SIGNAL_SMALLER SIGNAL_SMOOTH

Specials
Operators
+ - * / BITAND BITOR BITXOR SHL SHR SHRA ROL ROR
Boolean operators
AND OR XOR NOT > >= < <= <> =

Specials
Pre-Processor
#DEFINE #INCLUDE #PROJECT_MODEL #COMMENT #ENDCOMMENT
Compiler directives
USER_VAR_STRICT USER_STACK_SIZE USER_STRING_SIZE USER_SECURITY USER_EPORT SET_DATA_ADDR DATA
Declarations
BYTE WORD LONG REAL STRING ARRAY FIFO DATALABEL

3

Overview according to groups

Empty Page

3

Annotated overview

• = only in project model ,PM_FULL'

Program flow: General

SUB...END	Define subroutine
CALL	Call subroutine
Return	Quit subroutine

Program flow: Loops/branches

LOOP...ENDLOOP	Number of loops must be executed
FOR...NEXT	Loops with control variable: Start, stop, step
WHILE...ENDWHILE	Loop with abort condition at the beginning
IF..THEN..ELSE..ENDIF	Conditional program execution, possibly with alternative
GOTO	Unconditional program branch
SWITCH..CASE..ENDCASE	Conditional program execution, standard
SWITCHI..CASE..ENDCASE	Conditional programme execution, high speed

Program flow: Multitasking

Task ... END	Task definition
RUN_TASK	Starts task
STOP_TASK	Stops task
CONT_TASK	Continue stopped task
EXIT_TASK	End execution a task
DISABLE_TSW	Switches Task Switching mechanism OFF
ENABLE_TSW	Switches Task Switching mechanism ON
RELEASE_TASK	Release the current time slice
SET_TASK_PRIO	Sets priority for task switching
RESTART_PROG	• Re-start BASIC program, warm start
DELETE_PROG	• Deletes the current BASIC program, PC mode
CUT_AND_PASTE	• Cuts and pastes value
ON_ERROR_RESET	Resets Error flag after processing
ON_ERRTASK_CALL	Calls subroutine in the event of an error in the task
ON_ERRTASK_GOTO	Go to skip flag in the event of an error in the task

• = Only in project model ,PM_FULL'

Input and output: serial

SHIFT_IN	Clocked, serial input for external chip
SHIFT_OUT	Clocked, serial output to external chip
I2C_SETUP	Set I2b bus settings
I2C_READ\$	Reads bytes from I2C bus
I2C_WRITE	Write data string to I2C bus
I2C_RESULT	Read status of the last I2C bus operation
I2C_ACK_POLL	ACK-Polling on I2C bus whether device is ready

3

Input and output: Device / Device driver

INSTALL_DEVICE	Install device driver
GET	Unformatted data input from device driver
PUT	Unformatted data output to device driver
INPUT	Data input from dev. driver, separator
INPUT_LINE	Data input from dev. driver, line
PRINT	Formatted data output to dev. driver
PRINT_USING	Data output to dev. driver, + formatting
USING	Format instruction for PRINT_USING

• = only in project model ,PM_FULL'

Input and output: pins and ports

DIR_PIN		Sets the data direction of an internal I/O-pin
DIR_PORT		Sets the data direction of an internal I/O port
IN		Reads in bits from an I/O port
OUT		Write bits to I/O port
LL_IPOINT_AND		Bit for bit AND operation on internal I/O port
LL_IPOINT_IN		Read in from an internal I/O port + BITAND
LL_IPOINT_OR		Bit for bit OR operation on internal I/O port
LL_IPOINT_OUT		Fast output to internal I/O port
LL_IPOINT_pulse		Toggles a bit of an internal I/O port
LL_IPOINT_XOR		Bit for bit XOR operation on internal I/O port
XSETUP	•	Sets setting for XPort I/O extension
XIN, XIN\$	•	Reads byte(s) from XPort I/O extension system
XOUT	•	Writes byte(s) to XPort I/O extension
XBUS_INR\$	•	Fast import of data via XPort bus
XBUS_OUTR	•	Fast output of data via Xport bus

3

Buffer: Fifo

FIFO		Declaration of a variable of the type FIFO
CLEAR_FIFO		Delete Fifo and bring into starting condition
GET_FIFO		Fetch entry from Fifo (if present)
PUT_FIFO		Write entry in Fifo (if space)
UNGET_FIFO		Undo GET_FIFO (if possible)
PUTU_FIFO		Write entry in Fifo (oldest entry possibly lost)
LEN_FIFO		Report total length of Fifo
FREE_FIFO		Report number of free Fifo entries
READ_FIFO		Read arbitrary element from Fifo
START_FIFO		Delete Fifo + prepare for INTEGRAL_FIFO
INTEGRAL_FIFO		Use Fifo as short-term integrator (Window)

• = only in project model ,PM_FULL'

Buffer: String as buffer

NTOS\$	Fast: numerical value (integer) to string.
NFROMS	Fast: numerical value (integer) from string.
RTOS\$	Fast execution: real value to string.
RFROMS	Fast execution: real value from string.
STOS\$	Fast execution: string to string.

3

Flash

DATA	Define date in FLASH: bytes, words, files
SET_DATA_ADDR	Sets address in Data-FLASH area
POKE_FLASH	Write date in Data FLASH area (standard)
PEEK_FLASH	• Reads data from FLASH to a variable
POKEM_FLASH	• Write date in Data FLASH area
ERASE_FLASH	Deletes FLASH-sector(s), only data area
SET_SERIAL_NO	• Writes serial number, installation code, etc.
SERIAL_NO\$	• Reads serial number, installation code
DELETE_PROG	• Deletes the current BASIC program, PC mode
CHECK_FLASH	Finds last programmed position in User-FLASH
LET\$	• Copies string (high-speed)
RND	Delivers pseudo random value
RANDOMIZE	Changes pseudo random value series
LIN_APPROX	Intermediate stages by linear approximation

• = Only in project model 'PM_FULL'

Numerical functions: bit for bit

BIT	Test values of desired bit in a value
BIT_MIRR	Bit swaps in a variable (mirroring)
BITNOT	Inverts all bits of a numerical value
CUT_BITS	• Cuts bit groups from numerical value
INV_BIT	Inverts 1 bit in numerical variable
RES_BIT	Sets a bit in numerical variable to '0'
SET_BIT	Sets a bit in numerical variable to '1'
MASK	Returns a mask with a set bit
IMASK	Returns inverse mask, a "0" bit
SIGNEXT	Value extension with correct operational sign
PARITY	• Calculates parity of a numerical value

Numerical functions: mathematical

ABS	Amount of an integer or floating point value
SGN	Operational sign of a numerical value
CALC_CRC	• Calculation of a CRC checksum
EXP	Exponential function for integer value
EXPE	Exponential function for basis e, real value
FIX	Truncates fractional digits of a REAL
INT	Delivers next integer of a REAL
LIMIT	• Limits integer value to permissible range
LD	Logarithm Dualis, base 2
LN	Logarithm Naturalis, base E
LOG	Common logarithm
MOD	Modulo, remains of an integer division
SQRT	Root
MODULO_INC	• Modulo Increment, rising
MODULO_UPDO	• Circulating Modulo-INC/DEC
PRIME	• Tests whether integer value is a prime number.

• = Only in project model 'PM_FULL'

Numerical functions: trigonometrical functions	
SIN	SINE, real values
COS	COSINE, real values
TAN	TANGENT, real values
COT	COTANGENT, real values
ASIN	ARC SINE, real values
ACOS	ARC COSINE, real values
ATAN	ARC TANGENT, real values
ACOT	ARC COTANGENT, real values
SINH	SINE hyperbolicus, real values
COSH	COSINE hyperbolicus, real values
TANH	TANGENT hyperbolicus, real values
COTH	COTANGENT hyperbolicus, real values
QUICK_WORD_COS	• Integer COSINE, fast calculation
QUICK_WORD_SIN	• Integer SINE, fast calculation

3

• = Only in project model 'PM_FULL'

String operations: general	
LET\$	• String assignment, high-speed
LEN	Determines the current string length
MAX_LEN	Determines the maximum string length
SET_LEN\$	Sets the specific length a string
FILL\$	Fills string with assigned pattern string
LEFT\$	Delivers left partial string
MID\$	Delivers middle partial string
RIGHT\$	Delivers right partial string
REMOVE\$	Removes character from string
REMOUBLE\$	Removes double character from string
TRIM\$	Removes leading and/or following character
ADD\$	Adds up value of all bytes in a string
AND\$	Masks all bytes in a string by AND
OR\$	Masks all bytes in a string by OR
XOR\$	Links all bytes in a string by XOR
SHIFTL\$	• Shifts the bits in a string by 1...8 positions
SHIFTM\$	• Shifts the bits in a string by 1...8 positions
UPPER\$	Converts letters in a string to capitals
UPPER7\$	ditto, however removes any set bit 7
CONVERT\$	Character-to-character conversion acc. to own table
CONVERT\$	ditto, however removes any set bit 7
CODE_1TON\$	• Converts in a string 1 byte ==> N bytes
DECODE	• Decodes string data, (error-correcting codes)

• = only in project model ,PM_FULL'

3

Graphic: pixel graphic

OR2\$, OR3\$, OR4\$	• Combines 2, 3 or 4 strings byte-by-byte by OR
AND2\$, AND3\$, AND4\$	• Combines 2, 3 or 4 strings byte-by-byte by AND
XOR1\$	• Superimposes 1 string byte-by-byte by XOR
GRAPHIC_MASK_COPY	• Combines 2 graphics acc. to graphic mask
GRAPHIC_COPY	• Copy window from one graphic into another graphic
GRAPHIC_MIRROR	• Mirrors a pixel graphic
GRAPHIC_FILL_MASK	• Creates rectangular mask in graphic string
GRAPHIC_EXP\$	• Bit expansion for adjustment to output device
INVERT	• Inverts number of bytes in a string
FILL_AREA	• Fill areas with random contour
LET\$	• Copies string (high-speed)

Graphic: vector graphic

DRAW_LINE	• Draw line including line beginning
DRAW_NEXT_LINE	• Adds further line element to line
CLOSE_LINE	• Closes an open line
SET_BASE	• Sets the new coordinates origin
SET_SCALE	• Sets the new X-/Y scaling factor
SET_ROTATION	• Sets pen rotation
SET_GRAREA	• Sets graphic area
SET_DOT	• Sets single pixel in graphic
DISTANCE	• Determines the between two coordinate pairs

• = only in project model ,PM_FULL'

Special: conversion	
VAL_NUM	Reads integer value from a string
VAL_REAL	Reads real value from a string
STR\$	Converts numerical value in string
STRI\$	Converts numerical value in formatted string
CHR\$	Type conversion: byte -> 1 string character
ASC	Delivers the numerical value of a string character
LTR	Type conversion: LONG to REAL
LLTOR	Combines bytes from 2 LONGs to 1 REAL var.
HREAL	Delivers high 32-bit of REAL-value
LREAL	Delivers low 32-bit of REAL-value
RTL	Type conversion: REAL to Long
SIGNEXT	Value extension with correct sign

Mirrorings	
BIT_MIRR	Inverts sequence of bits in num. variable
BYTE_MIRR	Inverts sequence of bits in num. variable
BIT_MIRR\$	Inverts sequence of bits in string
MIRROR\$	Inverts sequence of characters in a string
GRAPHIC_MIRROR	• Mirrors pixel graphic on X and/or Y axis

• = Only in project model 'PM_FULL'

3

Compression / conversion

PACK_BYTE\$	• Compresses strings
UNPACK_BYTE\$	• De-compresses strings
BYTE_ANIB\$	• Compresses bytes to nibbles, data reduction
ANIB_BYTE\$	• De-Compresses nibbles back to bytes
BYTE_INIB\$	• Compresses bytes to incremental nibbles
INIB_BYTE\$	• De-Compresses I-nibbles back to bytes
ASC_BCD\$	• Converts: ASCII-HEX numbers -> BCD bytes
BCD_ASC\$	• Converts: BCD -> ASCII-HEX
ASC_PBCD\$	• Converts: ASCII-HEX numbers to "packed BCD"
PBCD_ASC\$	• Converts: packed BCD -> ASCII-HEX
BCD_PBCD\$	• Converts: BCD -> "packed BCD"
PBCD_BCD\$	• Converts: "packed BCD" -> BCD

Search / quantify

INSTR	Searches for character string in a string, delivers position
INDEX	Searches for character string in a string, delivers index
INDEX_1D	• 1D value classes, res. keyboards, tol. limit
INDEX_2D	• 2D value classes, X/Y-pos. (index, touchpanel)
SELECT\$	Selects string element on the basis of an index

Conversion / encoding

BIT_MIX\$	• Mixes bits in string acc. to mixing string
BIT_DEMIX\$	• Unmixes bits in string acc. to mixing string
BYTE_MIX\$	• Mixes bytes in string acc. to mixing string
BYTE_DEMIX\$	• Unmixes bytes in string acc. to mixing string
SCRAMBLE\$	• Makes string bytes indecipherable acc. to scramble string
CODE_1TON\$	• Converts in a string 1 byte ==> N bytes
DISTRIBUTE	• Distributes bit in a string (bitstream)

• = only in project model ,PM_FULLL'

Transaction numbers (TAN)	
TAN_CHK	• Checks whether valid TAN, sets "used"
TAN_COUNT	• Counts TANs: free, used, deleted
TAN_DEL	• Deletes TAN
TAN_FIRST	• Searches for 1st TAN: free, used or deleted

• = Only in project model 'PM_FULL'

Formats	
USING	Format str. for formatted output of num. value
PRINT USING	Formatted data output to dev. driver
SPC	Creates fixed no. of spaces in the PRINT output
TAB	Spaces in the PRINT output up to tabulator pos. .
STR!\$	Converts numerical value in formatted string
TRIM\$	Removes leading and/or following character
FRAME	Creates output frame string for PRINT_USING
TABS	Sets tabulator positions for PRINT_USING

3

Others	
SCAN_TO_CHAR_SETUP	• Setup keyboard scan
SCAN_TO_CHAR	• Scan button to generate the code in keyboards
SCAN_TO_TOGGLE	• Scan toggles in keyboards
DEBOUNCE	• Debounce: buttons, contacts, touchpanel, light barrier
DECODE	• Decode acc. to table, e.g.: error-correcting codes

System near	
RESTART_PROG	• Re-start BASIC program, warm start
DELETE_PROG	• Deletes the current BASIC program, PC mode
READ_TSW	Reads current setting of the task switch
THIS_VERS	Shows the current version of this runtime system
DEVEN	Reports last device error of a task
SYSVARN	Reads numerical system variable
SYSVAR\$	Reads string system variable
SET_SYSVARN	Sets system parameter

• = only in project model ,PM_FULL'

Time	
TICKS	Delivers current time of the 1 ms system clock pulse
SET_TICKS	Sets 1 ms system clock counter to new value
DIFF_TICKS	Delivers time difference in 1 ms system ticks
WAIT_DURATION	Execution of this task waits the time duration
WAIT_CLOCK	Execution of this task waits till TICKS value reached
WAIT_NEXT	Execution of this task waits the time GRID

3

Signal processing	
SIGNAL_ADD	• Superimposes 2 signals additive
SIGNAL_AVERAGE	• Determines the mean signal value, (DC components)
SIGNAL_CUT_AMP	• Limits signal amplitude
SIGNAL_CUT_MAX	• Limits maximum values in the signal
SIGNAL_CUT_MIN	• Limits minimum values in the signal
SIGNAL_GAP_END	• Searches for a gap in the signal
SIGNAL_NEXT_GAP	• Searches for next gap in the signal
SIGNAL_GAPI_END	• Searches for a gap in the signal, short-term integration
SIGNAL_NEXT_GAPI	• Searches for next gap in the signal, short-term integration
SIGNAL_MAX	• Determines maximum value of a signal
SIGNAL_MIN	• Determines minimum value of a signal
SIGNAL_MAX_AMPL	• Determines greatest amplitude of a signal
SIGNAL_MUL	• Multiplies 2 signals
SIGNAL_OFFSET	• Shifts signal level by an offset
SIGNAL_SCALE	• Amplifies or muffles the signal
SIGNAL_BIGGER	• Determines samples which are greater than comp. value
SIGNAL_SMALLER	• Determines samples which less than comp. value.
SIGNAL_SMOOTH	• Flattens the end of a signal

• = Only in project model 'PM_FULL'

Operators	
+	Add up (string: append)
-	subtract
*	multiply
/	divide
BITAND	bit for bit AND
BITOR	bit for bit OR
BITXOR	bit for bit exclusive OR
SHL n	Shift to the left by N bits
SHR n	Shift to the right by N bits
SHRA n	Arithmetical shift to the right
ROL n	Rotate to the left by N bits
ROR n	Rotate to the right by N bits

3

• = Only in project model 'PM_FULL'

Operators: Boolean

AND	logical AND
OR	logical OR
XOR	logical exclusive OR
NOT	logical NOT
>	greater than
>=	greater than or equal to
<	smaller than
<=	smaller than or equal to
<>	unequal
=	equal to

3

Preprocessor directives

#DEFINE	Defines symbol for remainder of line
#INCLUDE	Integrates other file in source text
#PROJECT_MODEL	Determines project model (large, small)
#COMMENT	Starts commentary area in source text
#ENDCOMMENT	Ends comment area in source text

Compiler directives

USER_STACK_SIZE	Sets task stack size
USER_VAR_STRICT	Forces variables definition
USER_STRING_SIZE	Sets standard string size
USER_EPORT	Sets parameter for the EPORT system
USER_SECURITY	Sets security word
SET_DATA_ADDR	Specifies address in data FLASH
DATA	Integrates data in Flash (binary, BMP, WAV)

• = Only in project model 'PM_FULL'

Variables declarations	
BYTE	Small value without operational sign (0...255)
WORD	Values without operational sign (0...65535)
LONG	Values from -2,147,483,648 to 2,147,483,647
REAL	Floating point values from $4,19 \cdot 10^{-307}$ to $1,67 \cdot 10^{308}$
STRING	Character string
DATALABEL	Address in data flash
ARRAY	Field of bytes, WORDs, LONGs or strings
FIFO	First-in-First-out buffer

3