

# 3 Instruktionen und Funktionen

## Übersicht nach Gruppen

Programmfluß	Ein- und Ausgabe	Puffer	Numerische Funktionen
<b>Unterprogramm</b>	<b>Seriell</b>	<b>Fifo</b>	<b>Bitweise</b>
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 siehe auch: Verschlüsselung
<b>Schleifen und Verzweigungen</b>	<b>Geräte</b>	<b>String als Puffer</b>	<b>Mathematisch</b>
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
<b>Taskverwaltung</b>	<b>Pins und Ports</b>	<b>Flash</b>	
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\$	

# Übersicht nach Gruppen

3

Numerische Funktionen
<b>Trigonometrische Funktionen</b>
ACOS
ACOT
ASIN
QUICK_WORD_COS
QUICK_WORD_SIN
ATAN
COS
COSH
COT
COTH
SIN
SINH
TAN
TANH

String-Operationen
<b>Allgemein</b>
LET\$
LEN
SET_LEN\$
MAX_LEN
FILL\$
LEFT\$
MID\$
RIGHT\$
REMOVE\$
REDOUBLE\$
ADD\$
AND\$
OR\$
XOR\$
SHIFTL\$
SHIFTM\$
UPPER\$
UPPER7\$
CONVERT\$
CONVERT7\$
<b>siehe auch:</b>
String als Puffer
Suchen
Spiegelung
Verschlüsselung
Komprimierung
Signalverarbeitung
<b>Formatierung</b>
FRAME
TABS
TRIM\$
USING
PRINT_USING
SPC
TAB
STRI\$
Formatstrings

Grafik
<b>Pixel</b>
OR\$, AND\$, XOR\$
GRAPHIC_MASK_COPY
GRAPHIC_MIRROR
GRAPHIC_COPY
INVERT
GRAPHIC_FILL_MASK
<b>Special</b>
GRAPHIC_EXP\$
FILL_AREA
<b>Vektor</b>
DRAW_LINE
DRAW_NEXT_LINE
CLOSE_LINE
SET_BASE
SET_SCALE
SET_ROTATION
SET_GRAREA
SET_DOT
DISTANCE

Spezielles
<b>Konvertierung</b>
VAL_NUM
VAL_REAL
STR\$
STRI\$
CHR\$
ASC
LTR
LLTOR
HREAL
LREAL
RTL
SIGNEXT
<b>Spiegelungen</b>
BIT_MIRR
BYTE_MIRR
BIT_MIRR\$
MIRROR\$
GRAPHIC_MIRROR
<b>Dateneingabe und – übertragung</b>
DEBOUNCE
DECODE
SCAN_TO_CHAR_
SETUP
SCAN_TO_CHAR
SCAN_TO_TOGGLE
siehe auch: Komp-
rimierung, Ver-
schlüsselung
<b>Suchen</b>
INSTR
INDEX
INDEX_1D
INDEX_2D
SELECT\$
<b>System</b>
SYSVARN
SYSVAR\$
SET_SYSVARN
RESTART_PROG
DELETE_PROG
THIS_VERS
DEVEN

# Übersicht nach Gruppen

<b>Spezielles</b>
<b>Komprimierung</b>
PACK_BYTE\$ UNPACK_BYTE\$ ASC_BCD\$ BCD_ASC\$ ASC_PBCD\$ PBCD_ASC\$ PBCD_BCD\$ BYTE_ANIB\$ ANIB_BYTE\$ BYTE_INIB\$ INIB_BYTE\$
<b>Zeit</b>
WAIT_DURATION WAIT_NEXT WAIT_CLOCK TICKS SET_TICKS
<b>Verschlüsselung</b>
BIT_MIX\$ BIT_DEMIX\$ BYTE_DEMIX\$ BYTE_DEMIX\$ SCRAMBLE\$ BYTE_MIX\$ CODE_1TON\$ DISTRIBUTE\$
<b>TAN – Trans-Aktions-Nummern</b>
TAN_CHK TAN_COUNT TAN_DEL TAN_FIRST

<b>Spezielles</b>
<b>Signalverarbeitung</b>
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

<b>Spezielles</b>
<b>Operatoren</b>
+ - * / BITAND BITOR BITXOR SHL SHR SHRA ROL ROR
<b>Boolsche Operatoren</b>
AND OR XOR NOT > >= < <= <> =

<b>Spezielles</b>
<b>Prä-Prozessor</b>
#DEFINE #INCLUDE #PROJECT_MODEL #COMMENT #ENDCOMMENT
<b>Compiler-Direktiven</b>
USER_VAR_STRICT USER_STACK_SIZE USER_STRING_SIZE USER_SECURITY USER_EPORT SET_DATA_ADDR DATA
<b>Deklarationen</b>
BYTE WORD LONG REAL STRING ARRAY FIFO DATALABEL

3

## Übersicht nach Gruppen

Leere Seite

3

## Kommentierte Übersicht

• = nur in Projektmodell ‚PM\_FULL‘

### Programmfluß: Allgemein

SUB...END	Unterprogramm definieren
CALL	Unterprogramm aufrufen
RETURN	Unterprogramm verlassen

### Programmfluß: Schleifen / Verzweigungen

LOOP...ENDLOOP	Anzahl von Schleifen unbedingt ausführen
FOR...NEXT	Schleifen mit Laufvariable: Start, Stopp, Schrittw.
WHILE...ENDWHILE	Schleife mit Abbruchbedingung am Anfang
IF..THEN..ELSE..ENDIF	Bedingte Programmausführung, ggf. mit Alternative
GOTO	Unbedingte Programmverzweigung
SWITCH..CASE..ENDCASE	Bedingte Programmausführung, Standard
SWITCHI..CASE..ENDCASE	Bedingte Programmausführung, high-speed

### Programmfluß: Multitasking

TASK ... END	Task Definition
RUN_TASK	Startet Task
STOP_TASK	Stoppt Task
CONT_TASK	Setze gestoppte Task wieder fort
EXIT_TASK	Beendet Ausführung einer Task
DISABLE_TSW	Schaltet Task-Switching Mechanismus AUS
ENABLE_TSW	Schaltet Task-Switching Mechanismus EIN
RELEASE_TASK	Rest der augenblicklichen Zeitscheibe freigeben
SET_TASK_PRIO	Setzt Priorität für Task-Switching
RESTART_PROG	• Startet BASIC-Programm neu, Warmstart
DELETE_PROG	• Löscht das aktuelle BASIC-Programm, PC-Mode
CUT_AND_PASTE	• Schneidet Wert heraus und kopiert
ON_ERROR_RESET	Setzt Fehler-Flag zurück nach der Bearbeitung
ON_ERRTASK_CALL	Ruft Unterprogramm auf bei Fehler in Task
ON_ERRTASK_GOTO	Springt zu Sprungmarke bei Fehler in Task

# Kommentierte Übersicht

• = nur in Projektmodell ‚PM\_FULL‘

## Ein- und Ausgabe: Seriell

SHIFT_IN	Getaktete, serielle Eingabe für externe Chips
SHIFT_OUT	Getaktete, serielle Ausgabe zu externen Chips
I2C_SETUP	I2C-Bus Einstellungen setzen
I2C_READ\$	Liest Bytes von I2C-Bus
I2C_WRITE	Schreibt Datenstring auf I2C-Bus
I2C_RESULT	Liest Status der letzten I2C-Bus Operation
I2C_ACK_POLL	ACK-Polling auf I2C-Bus ob Device bereit ist

3

## Ein- und Ausgabe: Geräte / Device-Treiber

INSTALL_DEVICE	Device Treiber installieren
GET	Unformatierte Daten-Eingabe von Device-Treiber
PUT	Unformatierte Daten-Ausgabe an Device-Treiber
INPUT	Daten-Eingabe von Dev.-Treiber, Trennzeichen
INPUT_LINE	Daten-Eingabe von Dev.-Treiber, Zeilen
PRINT	Formatierte Daten-Ausgabe an Dev.-Treiber
PRINT_USING	Daten-Ausgabe an Dev.-Treiber, + Formatierungen
USING	Format-Anweisung für PRINT_USING

## Ein- und Ausgabe: Pins und Ports

DIR_PIN	Setzt die Datenrichtung eines internen I/O-Pins
DIR_PORT	Setzt die Datenrichtung eines internen I/O-Ports
IN	Liest Bits von I/O-Port ein
OUT	Schreibt Bits auf I/O-Port
LL_IPORT_AND	Bitweise UND-Verknüpfung auf internem I/O-Port
LL_IPORT_IN	Einlesen von einem internem I/O-Port + BITAND
LL_IPORT_OR	Bitweise ODER-Verknüpfung auf internem I/O-Port
LL_IPORT_OUT	Schnelle Ausgabe auf internem I/O-Port
LL_IPORT_PULSE	Toggelt ein Bit eines internen I/O-Ports
LL_IPORT_XOR	Bitweise XOR-Verknüpfung auf internem I/O-Port
XSETUP	• Setzt Einstellung für XPort I/O-Erweiterung
XIN, XIN\$	• Liest Byte(s) von XPort I/O-Erweiterungs-System
XOUT	• Schreibt Byte(s) auf XPort I/O-Erweiterung
XBUS_INR\$	• Schnelles Einlesen von Daten über XPort-Bus
XBUS_OUTR	• Schnelles Ausgeben von Daten über Xport-Bus

• = nur in Projektmodell ‚PM\_FULL‘

## Puffer: Fifo

FIFO	Deklaration einer Variablen vom Typ Fifo
CLEAR_FIFO	Fifo löschen und in Anfangs-Zustand bringen
GET_FIFO	Hole Eintrag aus Fifo (wenn vorhanden)
PUT_FIFO	Schreibe Eintrag in Fifo (wenn Platz)
UNGET_FIFO	Mache GET_FIFO rückgängig (wenn möglich)
PUTU_FIFO	Schreibe Eintrag in Fifo (ältester Eintr. ev. verloren)
LEN_FIFO	Gesamt-Länge des Fifo melden
FREE_FIFO	Anzahl freier Fifo-Einträge melden
READ_FIFO	Beliebiges Element aus Fifo lesen
START_FIFO	Fifo löschen + vorbereiten für INTEGRAL_FIFO
INTEGRAL_FIFO	Fifo benutzen als Kurzzeit-Integrator (Window)

## Puffer: String als Puffer

NTOS\$	Schnell: Numerischer Wert (Integer) nach String.
NFROMS	Schnell: Numerischer Wert (Integer) aus String.
RTOS\$	Schnelle Ausführung: Real Wert nach String.
RFROMS	Schnelle Ausführung: Real Wert aus String.
STOS\$	Schnelle Ausführung: String nach String.

## Flash

DATA	Definiere Daten im FLASH: Bytes, Words, Files
SET_DATA_ADDR	Setzt Adresse in Data-FLASH Bereich
POKE_FLASH	Schreibt Daten in Daten-FLASH Bereich (standard)
PEEK_FLASH	• Liest Daten aus FLASH in Variable ein
POKEM_FLASH	• Schreibt Daten in Daten-FLASH Bereich
ERASE_FLASH	Löscht FLASH-Sektor(en), nur Daten-Bereich
SET_SERIAL_NO	• Schreibt Serien-Nummer, Installations-Code, etc.
SERIAL_NO\$	• Liest Serien-Nummer, Installations-Code
DELETE_PROG	• Löscht das aktuelle BASIC-Programm, PC-Mode
CHECK_FLASH	Findet letzte programmierte Stelle im User-Flash
LET\$	• Kopiert String (high-speed)
RND	Liefert Pseudo-Zufallswert
RANDOMIZE	Verändert Pseudo-Zufallswerte-Serien
LIN_APPROX	Zwischenschritte durch lineare Approximation

# Kommentierte Übersicht

• = nur in Projektmodell ‚PM\_FULL‘

## Numerische Funktionen: Bitweise

BIT	Testet Wert des gewünschten Bits in einem Wert
BIT_MIRR	Bit-Vertauschungen in einer Variablen (Spiegelung)
BITNOT	Invertiert alle Bits eines numerischen Wertes
CUT_BITS	• Schneidet Bit-Gruppe aus numerischem Wert aus
INV_BIT	Invertiert 1 Bit in numerischer Variable
RES_BIT	Setzt ein Bit in numerischer Variable auf ‚0‘
SET_BIT	Setze ein Bit in numerischer Variable auf ‚1‘
MASK	Gibt Maske mit einem gesetzten Bit zurück
IMASK	Gibt inverse Maske zurück, ein „0“ Bit
SIGNEXT	Vorzeichenrichtige Wert-Erweiterung
PARITY	• Berechnet Parity eines numerischen Wertes

## Numerische Funktionen: Mathematisch

ABS	Betrag eines Integer- oder Fließkomma-Wertes
SGN	Vorzeichen eines numerischen Wertes
CALC_CRC	• Berechnung einer CRC-Prüfsumme
EXP	Exponential-Funktion für Integer-Werte
EXPE	Exponential-Funktion zur Basis e, Real-Werte
FIX	Schneidet Nachkommastellen eines REALs ab
INT	Liefert nächst gelegene Ganzzahl eines REALs
LIMIT	• Begrenzt Integer Wert auf zulässigen Bereich
LD	Logarithmus Dualis, Basis 2
LN	Logarithmus Naturalis, Basis e
LOG	10-er Logarithmus
MOD	Modulo, Rest einer Integer-Division
SQRT	Wurzel
MODULO_INC	• Modulo Increment, steigend
MODULO_UPDO	• Umlaufendes Modulo-INC/DEC
PRIME	• Testet, ob Inteegerwert Primzahl ist.

• = nur in Projektmodell ‚PM\_FULL‘

<b>Numerische Funktionen: Trigonometrische Funktionen</b>	
SIN	SINUS, Real-Werte
COS	COSINUS, Real-Werte
TAN	TANGENS, Real-Werte
COT	COTANGENS, Real-Werte
ASIN	ARCUS SINUS, Real-Werte
ACOS	ARCUS COSINUS, Real-Werte
ATAN	ARCUS TANGENS, Real-Werte
ACOT	ARCUS COTANGENS, Real-Werte
SINH	SINUS hyperbolicus, Real-Werte
COSH	COSINUS hyperbolicus, Real-Werte
TANH	TANGENS hyperbolicus, Real-Werte
COTH	COTANGENS hyperbolicus, Real-Werte
QUICK_WORD_COS	<ul style="list-style-type: none"><li>• Integer COSINUS, schnelle Berechnung</li></ul>
QUICK_WORD_SIN	<ul style="list-style-type: none"><li>• Integer SINUS, schnelle Berechnung</li></ul>

# Kommentierte Übersicht

• = nur in Projektmodell ‚PM\_FULL‘

## String-Operationen: Allgemein

LET\$	• String-Zuweisung, high-speed
LEN	Ermittelt die aktuelle String-Länge
MAX_LEN	Ermittelt die maximale String-Länge
SET_LEN\$	Setzt die Länge eines Strings gezielt
FILL\$	Füllt String mit bel. Muster-String
LEFT\$	Liefert linken Teilstring
MID\$	Liefert mittleren Teilstring
RIGHT\$	Liefert rechten Teilstring
REMOVE\$	Entfernt Zeichen aus String
REMOUBLE\$	Entfernt doppelte Zeichen aus String
TRIM\$	Entfernt führende und/oder nachfolgende Zeichen
ADD\$	Addiert Wert auf alle Bytes eines Strings
AND\$	Maskiert alle Bytes eines Strings per AND
OR\$	Maskiert alle Bytes eines Strings per OR
XOR\$	Verknüpft alle Bytes eines Strings per XOR
SHIFTL\$	• Schiebt die Bits eines Strings um 1...8 Positionen
SHIFTM\$	• Schiebt die Bits eines Strings um 1...8 Positionen
UPPER\$	Konvertiert Buchstaben in String zu Großbuchst.
UPPER7\$	Dito, jedoch entferne ev. gesetztes Bit-7
CONVERT\$	Zeichen-zu-Zeichen Konversion lt. eigener Tabelle
CONVERT\$	Dito, jedoch entferne ev. gesetztes Bit-7
CODE_1TON\$	• Konvertiert in String: 1 Byte ==> N Bytes
DECODE	• Decodiert Stringdaten, (fehlerkorrigierende Codes)

## Grafik: Pixel-Grafik

OR2\$, OR3\$, OR4\$	• Vereint 2, 3 oder 4 Strings byteweise per OR
AND2\$, AND3\$, AND4\$	• Vereint 2, 3 oder 4 Strings byteweise per AND
XOR1\$	• Überlagert 1 String byteweise per XOR
GRAPHIC_MASK_COPY	• Vereinigt 2 Grafiken lt. Grafik-Make
GRAPHIC_COPY	• Kopiert Fenster aus einer Grafik in andere Grafik
GRAPHIC_MIRROR	• Spiegelt eine Pixel-Grafik
GRAPHIC_FILL_MASK	• Baut rechteckige Maske in Grafik-String auf
GRAPHIC_EXP\$	• Bit-Expansion zur Anpassung an Ausgabe-Geräte
INVERT	• Invertiert Anzahl Bytes in einem String
FILL_AREA	• Füllt Gebiete mit beliebiger Kontur
LET\$	• Kopiert String (high-speed)

• = nur in Projektmodell ‚PM\_FULL‘

## Grafik: Vektor-Grafik

DRAW_LINE	• Zeichnet Linie inklusive Linienanfang
DRAW_NEXT_LINE	• Fügt weiteres Linienelement an Linie an
CLOSE_LINE	• Schließt einen offenen Linienzug
SET_BASE	• Setzt den Koordinaten-Ursprung neu
SET_SCALE	• Setzt X-/Y-Skalierungs-Faktoren neu
SET_ROTATION	• Setzt Schreibstift Rotation
SET_GRAREA	• Setzt Grafik-Area
SET_DOT	• Setzt einzelnes Pixel in Grafik
DISTANCE	• Ermittelt den Abstand zweier Koordinaten-Paare

## Spezielles: Konvertierung

VAL_NUM	Liest Integer Wert aus einem String
VAL_REAL	Liest Real Wert aus einem String
STR\$	Wandelt numerischen Wert um in String
STRIS\$	Wandelt numerischen Wert in formatierten String
CHR\$	Typumwandlung: Byte -> 1 Stringzeichen
ASC	Liefert den numerischen Wert eines Stringzeichens
LTR	Typumwandlung: LONG to REAL
LLTOR	Kombiniert Bytes aus 2 LONGs zu 1 REAL-Var.
HREAL	Liefert high 32-Bits of REAL-Wert
LREAL	Liefert low 32-Bits of REAL-Wert
RTL	Typumwandlung: REAL to LONG
SIGNEXT	Vorzeichenrichtige Wert-Erweiterung

## Spiegelungen

BIT_MIRR	Kehrt Reihenfolge der Bits um in num. Variable
BYTE_MIRR	Kehrt Reihenfolge der Bits um in num. Variable
BIT_MIRR\$	Kehrt Reihenfolge der Bits um in String
MIRROR\$	Kehrt Reihenfolge der Zeichen in einem String um
GRAPHIC_MIRROR	• Spiegelt Pixel-Grafik an X- und/oder Y-Achse

# Kommentierte Übersicht

• = nur in Projektmodell ‚PM\_FULL‘

## Komprimierung / Konvertierung

PACK_BYTE\$	• Komprimiert Strings
UNPACK_BYTE\$	• De-Komprimiert Strings
BYTE_ANIB\$	• Komprimiert Bytes zu Nibbles, Datenreduktion
ANIB_BYTE\$	• De-Komprimiert Nibbles wieder zu Bytes
BYTE_INIB\$	• Komprimiert Bytes zu Incremental-Nibbles
INIB_BYTE\$	• De-Komprimiert I-Nibbles wieder zu Bytes
ASC_BCD\$	• Konvertiert: ASCII-HEX Ziffern -> BCD-Bytes
BCD_ASC\$	• Konvertiert: BCD -> ASCII-HEX
ASC_PBCD\$	• Konvertiert: ASCII-HEX Ziffern zu „packed BCD“
PBCD_ASC\$	• Konvertiert: packed BCD -> ASCII-HEX
BCD_PBCD\$	• Konvertiert: BCD -> „packed BCD“
PBCD_BCD\$	• Konvertiert: „packed BCD“ -> BCD

## Suchen / Quantifizieren

INSTR	Sucht Zeichenfolge in einem String, liefert Position
INDEX	Sucht Zeichenfolge in einem String, liefert Index
INDEX_1D	• 1D-Wertklassen, Widerst.-Tastaturen, Toleranzgr.
INDEX_2D	• 2D-Wertklassen, X/Y-Pos. (Index, Touchpanel)
SELECT\$	Selektiert String-Baustein an Hand eines Indexes

## Konvertierung / Chiffrierung

BIT_MIX\$	• Verwürfelt Bits in String lt. Verwürfelungs-String
BIT_DEMIX\$	• Entwürfelt Bits in String lt. Verwürfelungs-String
BYTE_MIX\$	• Verwürfelt Bytes in String lt. Verwürfelungs-String
BYTE_DEMIX\$	• Entwürfelt Bytes in String lt. Verwürfelungs-String
SCRAMBLE\$	• Macht Stringbytes unkenntlich lt. Scramble-String
CODE_1TON\$	• Konvertiert in String: 1 Byte ==> N Bytes
DISTRIBUTE	• Verteilt Bits in einem String (Bitstream)

## Transaktions-Nummern (TAN)

TAN_CHK	• Prüft ob gültige TAN, setz „benutzt“
TAN_COUNT	• Zählt TANs: freie, benutzte, gelöschte
TAN_DEL	• Löscht TAN
TAN_FIRST	• Sucht 1.TAN: freie, benutzte oder gelöschte

• = nur in Projektmodell ‚PM\_FULL‘

## Formatierungen

USING	Formatstr. f. formatierte Ausg. v. num. Werten
PRINT USING	Formatierte Daten-Ausgabe an Dev.-Treiber
SPC	Erzeugt feste Anz. Leerz. in der PRINT-Ausgabe
TAB	Leerz. in der PRINT-Ausgabe bis zu Tabulatorpos.
STRIS\$	Wandelt numerischen Wert in formatierten String
TRIM\$	Entferne führende und/oder nachfolgende Zeichen
FRAME	Erzeugt Ausgaberahmenstring f. PRINT_USING
TABS	Setzt Tabulatorpositionen für PRINT_USING

## Sonstige

SCAN_TO_CHAR_SETUP	• Tastatur-Scan einrichten
SCAN_TO_CHAR	• Scan Taste, die Code generiert in Keyboards
SCAN_TO_TOGGLE	• Scan Toggles in Keyboards
DEBOUNCE	• Entprellt:Tasten, Kontakte, Touchpanel, Lichtschr.
DECODE	• Decodiere lt. Table, z.B: fehlerkorrigierende Codes

## Systemnahe

RESTART_PROG	• Startet BASIC-Programm neu, Warmstart
DELETE_PROG	• Löscht das aktuelle BASIC-Programm, PC-Mode
READ_TSW	Liest akt. Einstellung des Task-Switches
THIS_VERS	Gibt die akt. Version dieses Runtime-Systems
DEVEN	Meldet letzten Device Error einer Task
SYSVARN	Liest numerische Systemvariable
SYSVAR\$	Liest String-Systemvariable
SET_SYSVARN	Setzt Systemparameter

## Zeit

TICKS	Liefert akt. Zeit des 1 ms System-Taktes
SET_TICKS	Setzt 1 ms System-Taktzähler auf neuen Wert
DIFF_TICKS	Liefert Zeitdifferenz in 1 ms System-Ticks
WAIT DURATION	Ausführung dieser Task wartet Zeit-DAUER
WAIT CLOCK	Ausf. dieser Task wartet bis TICKS Wert erreicht ist
WAIT NEXT	Ausführung dieser Task wartet Zeit-RASTER

# Kommentierte Übersicht

• = nur in Projektmodell ‚PM\_FULL‘

Signalverarbeitung	
SIGNAL_ADD	• Überlagert 2 Signale additiv
SIGNAL_AVERAGE	• Ermittelt den Signal-Mittelwert, (DC-Komponente)
SIGNAL_CUT_AMP	• Begrenzt Signal-Amplitude
SIGNAL_CUT_MAX	• Begrenzt Maximalwerte im Signal
SIGNAL_CUT_MIN	• Begrenzt Minimalwerte im Signal
SIGNAL_GAP_END	• Sucht eine Lücke im Signal
SIGNAL_NEXT_GAP	• Sucht nächste Lücke im Signal
SIGNAL_GAPI_END	• Sucht eine Lücke im Signal, Kurzzeitintegration
SIGNAL_NEXT_GAPI	• Sucht nächste Lücke im Signal, Kurzzeitintegration
SIGNAL_MAX	• Ermittelt Maximalwert eines Signals
SIGNAL_MIN	• Ermittelt Minimalwert eines Signals
SIGNAL_MAX_AMPL	• Ermittelt größte Amplitude eines Signals
SIGNAL_MUL	• Multipliziert 2 Signale
SIGNAL_OFFSET	• Verschiebt Signalpegel um einen Offset
SIGNAL_SCALE	• Verstärkt oder Dämpft dsas Signal
SIGNAL_BIGGER	• Ermittelt Samples die größer sind als Vergleichsw.
SIGNAL_SMALLER	• Ermittelt Samples die kleiner sind als Vergleichsw.
SIGNAL_SMOOTH	• Flacht die Enden eines Signals ab

Operatoren	
+	Addieren (String: anhängen)
-	subtrahieren
*	multiplizieren
/	Dividieren
BITAND	Bitweise UND
BITOR	Bitweise ODER
BITXOR	Bitweise Exklusiv ODER
SHL n	links schieben um n Bits
SHR n	rechts schieben um n Bits
SHRA n	arithmetisch rechts schieben
ROL n	links rotieren um n Bits
ROR n	rechts rotieren um n Bits

3

• = nur in Projektmodell ‚PM\_FULL‘

## Operatoren: Boolesche

AND	logisch UND
OR	logisch ODER
XOR	logisch exklusiv ODER
NOT	logisch NICHT
>	größer
>=	größer oder gleich
<	kleiner
<=	kleiner oder gleich
<>	ungleich
=	gleich

3

## Prä-Prozessor-Direktiven

#DEFINE	Definiert Symbol für Rest der Zeile
#INCLUDE	Bindet andere Datei in Quelltext ein
#PROJECT_MODEL	Legt Projektmodell fest (groß, klein)
#COMMENT	Beginnt Kommentarbereich in Quelltext
#ENDCOMMENT	Beendet Kommentarbereich in Quelltext

## Compiler-Direktiven

USER_STACK_SIZE	Setzt Task-Stackgröße
USER_VAR_STRICT	Erzwingt Variablendefinition
USER_STRING_SIZE	Setzt Standard-Stringgröße
USER_EPORT	Setzt Parameter des EPORT-Systems
USER_SECURITY	Setzt Sicherungswort
SET_DATA_ADDR	Legt Adresse im Daten-Flash fest
DATA	Bindet Daten in Flash ein (binär, BMP, WAV)

## Variablen-Deklarationen

BYTE	Kleine Werte ohne Vorzeichen (0...255)
WORD	Werte ohne Vorzeichen (0...65535)
LONG	Werte von -2.147.483.648 bis 2.147.483.647
REAL	Fließkommawerte von $4.19 \cdot 10^{-307}$ bis $1.67 \cdot 10^{308}$
STRING	Zeichenkette
DATALABEL	Adresse im Daten-Flash
ARRAY	Feld aus Bytes, WORDs, LONGs oder Strings
FIFO	First-in-First-out-Puffer