Index

A/D inputs with Analog3x

The device driver 'ANALOG3x' reads in analog values from the external A/D-module EP11...EP14.

Further information on ANALOG3x.TDD:

- Secondary addresses of the ANALOG3x
- User-Function-Codes of the ANALOG3x
- Timing of ANALOG3x
- Scale measurement range
- Adjust input voltage range
- Define channel groups
- Delete channel group
- Read A/D-value and read from a channel group

File name: ANALOG3x8.TDD
ANALOG3x64.TDD

INSTALL DEVICE #D, "ANALOG3x8.TDD", P1,..., P13
INSTALL DEVICE #D, "ANALOG3x64.TDD", P1,..., P13

D is a constant, variable or expression of the data type BYTE, WORD or LONG in the range 0...63 and stands for the device number of the driver.

P1...P13 are further parameters which determine the connection of the EP11-module to BASIC-Tiger®. These parameters are described in the following table.

All parameters are bytes and can leave the standard value unchanged by specifying 0 or 0EEH (=238).

	Standard	leave unchanged	Description of parameter
P1	6	0	Logical BUS address for EP11
P2	8	0	Logical port address for: -RD, -WR, HBEN, -CE
P3	0	0eeh	Bit number for Signal '-RD'
P4	1	0eeh	Bit number for signal '-WR'
P5	3	0eeh	Bit number for signal '-HBEN'
P6	5	0eeh	Bit number for signal '-CE'
P7	1	0eeh	I/O Access-Speed-Reduction (1=no,2...120) Tiger-1: 1 Tiger-2: 40
P8	0	0eeh	Always 0, reserved parameter
P9	0	0eeh	0 = no address signals used 1 = an address line for 2 A/D ports 2 = two address lines for 4 A/D ports 3 = three address lines for 8 A/D-Ports
P10	7	0	Logical port address for address signals from EP11
P11	0	0eeh	First bit-position for address signals (0...7) e.g. two address signals at port 8, first position 6: L86 and L87 are address signals
P12	6	0eeh	Integration width: 0 = 1 sample 1 = 2 samples 2 = 4 samples 3 = 8 samples 4 = 16 samples 5 = 32 samples 6 = 64 samples 7 = 128 samples
P13	1	0eeh	No. of channels measured per 1ms

The device driver ANALOG3x is an enhancement of ANANLOG3.TDD. The measurements of ANALOG3x are not executed with the GET instruction. The device driver permanently samples the analog inputs of the EP11...EP14 time-controlled and adds the values into the integration buffer. The size of the integration buffer is passed with *P12* with the *install_device* instruction. With the GET instruction the current integrated value is read out. Please read *Timing of ANALOG3x* for detailed information.

I/O Access-Speed-Reduction must not be changed normally. The correct values are given by the device driver. If you don't get correct values, try to increase the standard value. This will slow down the sample rate.

The control lines may partly be used together with other device drivers, particularly with the control lines of the graphic LCD.

The device driver ANALOG3x reads in analog measured values from the external analog channels of the analog extension module EP11...EP14. The measurements are executed in the background permanently. The current integrated value is read out with the GET instruction.

The resolution is 16 bit (word). The driver supports the different modes of the A/D inputs of the EP11...EP14 module:

- Input voltage 0...5V
- Input voltage -5V...+ 5V
- Input voltage 0...10V
- Input voltage -10V...+ ..+10V

Two different read modes are available:

- from a random channel
- from a channel compiled beforehand

The settings for the analog measuring system are carried out by an output of strings to certain secondary addresses. Settings which have been made once are retained and do not have to be repeated before each measurement.

The following table shows an overview of the special functions of the driver on the secondary address.

Secondary addresses of ANALOG3X

Secondary address	Description
0...63	PUT: Set the input voltage ranges for channels 0 to 63 GET: Read from a single channel
64...71	PUT: Define channel groups (8 groups are possible) GET: Read from channel group of all defined channels

User-Function-Codes of ANALOG3X

User-Function-Codes of the ANALOG3x to set parameters (PUT):

No	Symbol Prefix: UFCO_	Description
144	UFCO_AD3_FACTOR	Sets the value which is delivered with a maximum A/D measured value of the corresponding channel (WORD). 65536 corresponds to factor 1

Timing of ANALOG3X

The exact timing of the Analog3x can be set individually by the Tiger-BASIC® program. Analog3x samples the channels of the EP11...EP14 time-controlled every millisecond. The number of measured channels per millisecond is set with *P13* from *install_device*. If you choose a high value, the sampling cycle is faster, but the CPU load is much bigger, too. The following table illustrates the CPU load:

No. of channels / ms	Tiger-1	Tiger-2
1	64µs	30µs
2	120µs	56µs
3	176µs	84µs
4	228µs	112µs
8	---	222µs

The Tiger-1 has a CPU load of about 6% per channel, the Tiger-2 generate a CPU load of about 3% per channel. The parameter is limited to 8 (Tiger-2: 16) to avoid too much CPU load.

The next table will demonstrate some timing examples. The parameters are the channels/ms, size of integration buffer and the number of channels. The result is the time of one complete cycle to fill the integration buffer. The measured values are correct after the integration buffer has been filled once.

The accuracy of integrated values improves with the size of the integration buffer used, i.e. bigger array gives higher accuracy. However, the inherent low-pass filter effect of an integrator will remove the higher frequency components form the recorded signal spectrum.

No. of channels / ms	Size of buffer	No. of channels	Cycle to fill
1	128	64	8 sec
2	128	64	4 sec
4	128	64	2 sec
8	128	64	1 sec
1	128	8	1 sec
2	128	8	512 ms
4	128	8	256 ms
8	128	8	128 ms
1	64	64	4 sec
2	64	64	2 sec
4	64	64	1 sec
8	64	64	512 ms
1	64	8	512 ms
2	64	8	256 ms
4	64	8	128 ms
8	64	8	64 ms
1	16	64	1 sec
2	16	64	512 ms
4	16	64	256 ms
8	16	64	128 ms
1	16	8	128 ms
2	16	8	64 ms
4	16	8	32 ms
8	16	8	16 ms

Scale measurement range

A value can be set which is then used to scale the measured value with the User-Function-code UFCO_AD3_FACTOR. The scaling factor can be set individually for every channel by writing to the secondary address of the channel. If several values are written to a secondary address the factors will be issued as of the secondary address and for the following addresses. After a Reset the values are set to the factor 1, corresponding to the factor value 65536.

The A/D- converter delivers as maximum value 65536. The following calculation is carried out with the factor:

Measurement result = (measurement * factor + 32768) / 65536

2048 are added after multiplication for rounding off. A factor value of e.g. 100 leads to the following calculation at a maximum value of the A/D converter:

Measurement result = (65536 * 100 + 32768) / 65536

leading to a measurement result of 100. The factor value thus also specifies the desired value for end scale deflection.

The following example shows the scale measurement:

```
user_var_strict
#include ufunc4.inc

TASK Main
  BYTE EVER                              ' endless loop
  WORD W                                 ' analog value
  REAL V                                 ' voltage
  string scale$                          ' scale measurement range

  DIR_PORT 8,0                           '
  INSTALL_DEVICE #1, "LCD1.TDD"

#ifdef TIGER_1
  INSTALL DEVICE #4, "ANALOG3x64.TDD", &   ' install Analog3x64 device
driver
    6, &      ' Por address databus
    8, &      ' Port address control lines
    3, &      ' Pin-Nr. -RD
    4, &      ' Pin-Nr. -WR
    5, &      ' Pin-Nr. HBEN
    7, &      ' Pin-Nr. -CE
    0, &      ' speed reduction = Tiger-1
    0, &      ' reserved
    3, &      ' address lines
    8, &      ' Port address adsress signals
    0, &      ' Bitposition of asdress signals
    7, &      ' size of buffer: 0=1; 1=2; 2=4; 3=8; 4=16; ...; 7=128
    4         ' No. of channles measured per 1ms
#endif


#ifdef TIGER_2
  INSTALL DEVICE #4, "ANALOG3x64.TD2", &   ' install Analog3x64 device
driver
    6, &      ' Por address databus
    8, &      ' Port address control lines
    3, &      ' Pin-Nr. -RD
    4, &      ' Pin-Nr. -WR
    5, &      ' Pin-Nr. HBEN
    7, &      ' Pin-Nr. -CE
    40, &      ' speed reduction = Tiger-2
    0, &      ' reserved
    3, &      ' address lines
    8, &      ' Port address adsress signals
    0, &      ' Bitposition of asdress signals
    7, &      ' size of buffer: 0=1; 1=2; 2=4; 3=8; 4=16; ...; 7=128
    8         ' No. of channles measured per 1ms
#endif



  PRINT #1,"EP11-EP14 Example:"
  PUT #4, #0, fill$("<0>", 64)           ' ch0...63: alle 0...5V
  scale$=""                              ' init scale$
  scale$ = ntos$(scale$, 0, 4, 5000)     ' scale 0...5000 (used as mv)
  PUT #4, #0, #UFCO_AD3_FACTOR, scale$   ' set scale measurement
```

```
  FOR EVER = 0 TO 0 STEP 0              ' endless loop
    GET #4, #0, 2, W                    ' read value channel 0
    PRINT #1, "<1Bh>A<0><1><F0h>Result:"; "          ";
    PRINT #1, "<1Bh>A<0><1><F0h>Result:"; W; " mv"
    WAIT_DURATION 100                   ' wait 100 ms
  NEXT
END                                     ' Ende Task MAIN
```

Adjust input voltage range

An input voltage range for the A/D converter can be set at the extension modules EP11...EP14 for every channel. The desired area is coded in the lower two bits of a byte. A byte is transferred to the device driver with PUT for every channel to be set. The first channel to be set is selected by writing to a certain secondary address. The following bytes are used accordingly to the following channels. The table shows the coding for the areas:

Binary coding	HEX coding	Input voltage range
00000000b	0	unipolar 0V...+5V
00000001b	1	bipolar –5V...+5V
00000010b	2	unipolar 0V...+10V
00000011b	3	bipolar –10V...+10VV

The following example sets the input voltage range of the channels 17 to 22:

```
'               +---------------------- Kanal 17:   0V... +5V
'               !  +------------------- Kanal 18:   0V...+10V
'               !  !  +---------------- Kanal 19:  -5V... +5V
'               !  !  !  +------------- Kanal 20: -10V...+10V
'               !  !  !  !  +---------- Kanal 21: -10V...+10V
'               !  !  !  !  !  +--------Kanal 22:  -5V... +5V
PUT #AD3, #17, "00 02 01 03 03 01"%
```

The following example sets the voltage range to -5V...+5V:

```
user_var_strict
#include ufunc4.inc

TASK Main
  BYTE EVER                              ' endless loop
  WORD W                                 ' analog value
  REAL V                                 ' voltage

  DIR_PORT 8,0                           '
  INSTALL_DEVICE #1, "LCD1.TDD"

#ifdef TIGER_1
  INSTALL DEVICE #4, "ANALOG3x64.TDD", &   ' install Analog3x64 device
driver
    6, &       ' Por address databus
    8, &       ' Port address control lines
    3, &       ' Pin-Nr. -RD
    4, &       ' Pin-Nr. -WR
    5, &       ' Pin-Nr. HBEN
    7, &       ' Pin-Nr. -CE
    0, &       ' speed reduction = Tiger-1
    0, &       ' reserved
    3, &       ' address lines
    8, &       ' Port address adsress signals
    0, &       ' Bitposition of asdress signals
    7, &       ' size of buffer: 0=1; 1=2; 2=4; 3=8; 4=16; ...; 7=128
    4          ' No. of channles measured per 1ms
#endif


#ifdef TIGER_2
  INSTALL DEVICE #4, "ANALOG3x64.TD2", &   ' install Analog3x64 device
driver
    6, &       ' Por address databus
    8, &       ' Port address control lines
    3, &       ' Pin-Nr. -RD
    4, &       ' Pin-Nr. -WR
    5, &       ' Pin-Nr. HBEN
    7, &       ' Pin-Nr. -CE
    40, &       ' speed reduction = Tiger-2
    0, &       ' reserved
    3, &       ' address lines
    8, &       ' Port address adsress signals
    0, &       ' Bitposition of asdress signals
    7, &       ' size of buffer: 0=1; 1=2; 2=4; 3=8; 4=16; ...; 7=128
    8          ' No. of channles measured per 1ms
#endif



  PRINT #1,"EP11-EP14 Example:"
  PUT #4, #0, fill$("<01>", 64)                 ' ch0...63: all -5V...5V



  FOR EVER = 0 TO 0 STEP 0               ' endless loop
    GET #4, #0, 2, W                     ' read value channel 0
```

```
    V = (LTR(SIGNEXT(W, 16))/32768.0 * 5000.0)/1000.0 ' calculate voltage
    USING "NF<1><1> <2>V0 0 0 0 0 0 0 1.2 0 0 0 0 0 0 0"
    PRINT_USING #1, "<1Bh>A<0><1><F0h>Analog-0:";V;"V"
    PRINT #1, "<1Bh>A<0><2><F0h>Digital:"; "          ";
    PRINT #1, "<1Bh>A<0><2><F0h>Digital:"; W;
    USING "UH<8><8>0k 0 0 0 0 8"
    PRINT_USING #1, "<1Bh>A<0><3><F0h>Hex:"; W;
    WAIT_DURATION 100                   ' wait 100 ms
  NEXT
END
```

Define channel groups

If certain A/D channels are read more often and other channels only rarely, channel groups can be formed to compile the reading of certain channels. The device driver ANALOG3x supports up to 8 groups. Each group can contain up to 64 channels. The groups are defined by an output of strings with the instruction PUT to the secondary addresses 64...71.

Secondary address	Channel group
64	Channel group 0
65	Channel group 1
66	Channel group 2
67	Channel group 3
68	Channel group 4
69	Channel group 5
70	Channel group 6
71	Channel group 7

The output strings contain the channel numbers. Channel numbers outside the valid range will be ignored. Channel may be freely compiled into a group. Double usage as well as overlapping with other groups is also allowed. The channels will later be read in the order in which they have been defined. A group can be recompiled at any time by simple resetting it.

```
PUT #AD3, #71, "3F 00 01 2B 03"%        ' Kanalgruppe 7: 5 Kanaele
PUT #AD3, #65, "<0><1><29><17>"    ' Kanalgruppe 1: 4 Kanaele
PUT #AD3, #66, FILL$(64,"<4>")     ' Kanalgruppe 2: 64 Kanaele
```

Delete channel group

To delete a group, set a channel outside the range of values, e.g. channel *99*.

```
PUT #AD3, #64, "<99>"              ' Gruppe 0: leer
```

Read A/D-value singly and from a channel group

An A/D value is read from a channel by simply reading from the corresponding secondary address. Since the resolution is 16 bit, a WORD (2 bytes) will be needed for every analog value.

```
GET #AD3, #10, 2, wVar            ' liest Wert von Kanal 10
```

Reading with the instruction GET on the secondary addresses 64 to 71 reads all channels in this group, though the maximum number of bytes that the variable can contain. The channels will be read in the order in which they have been defined beforehand in the group.

The following example sets channel groups:

```
user_var_strict
#include ufunc4.inc

TASK Main
  BYTE EVER                              ' endless loop
  WORD W                                 ' analog value
  REAL V                                 ' voltage
  string Group$(4)                       '

  DIR_PORT 8,0                           '
  INSTALL_DEVICE #1, "LCD1.TDD"

#ifdef TIGER_1
  INSTALL DEVICE #4, "ANALOG3x64.TDD", &   ' install Analog3x64 device
driver
    6, &      ' Por address databus
    8, &      ' Port address control lines
    3, &      ' Pin-Nr. -RD
    4, &      ' Pin-Nr. -WR
    5, &      ' Pin-Nr. HBEN
    7, &      ' Pin-Nr. -CE
    0, &      ' speed reduction = Tiger-1
    0, &      ' reserved
    3, &      ' address lines
    8, &      ' Port address adsress signals
    0, &      ' Bitposition of asdress signals
    7, &      ' size of buffer: 0=1; 1=2; 2=4; 3=8; 4=16; ...; 7=128
    4         ' No. of channles measured per 1ms
#endif


#ifdef TIGER_2
  INSTALL DEVICE #4, "ANALOG3x64.TD2", &   ' install Analog3x64 device
driver
    6, &      ' Por address databus
    8, &      ' Port address control lines
    3, &      ' Pin-Nr. -RD
    4, &      ' Pin-Nr. -WR
    5, &      ' Pin-Nr. HBEN
    7, &      ' Pin-Nr. -CE
    40, &      ' speed reduction = Tiger-2
    0, &      ' reserved
    3, &      ' address lines
    8, &      ' Port address adsress signals
    0, &      ' Bitposition of asdress signals
    7, &      ' size of buffer: 0=1; 1=2; 2=4; 3=8; 4=16; ...; 7=128
    8         ' No. of channles measured per 1ms
#endif



  PRINT #1,"EP11-EP14 Example:"
  PUT #4, #0, fill$("<0>", 64)           ' ch0...63: alle 0...5V
  PUT #4, #64, "<7><6>"                  ' first channel 7, second channel 6

  FOR EVER = 0 TO 0 STEP 0               ' endless loop
    GET #4, #64, 4, Group$               ' Wert aus AD-Wandler lesen
```

```
    W = NFROMS(Group$, 0, 2)             ' W <- channel-7
    V = (LTR(W)/65536.0 * 5000.0)/1000.0' calculate voltage
    USING "NF<1><1> <2> 0 0 0 0 0 0 0 1.2 0 0 0 0 0 0 0 0"
    PRINT_USING #1, "<1Bh>A<0><1><F0h>Analog-7:";V;"V"

    W = NFROMS(Group$, 2, 2)             ' W <- channel-6
    V = (LTR(W)/65536.0 * 5000.0)/1000.0' calculate voltage
    USING "NF<1><1> <2> 0 0 0 0 0 0 0 1.2 0 0 0 0 0 0 0 0"
    PRINT_USING #1, "<1Bh>A<0><2><F0h>Analog-6:";V;"V"
    WAIT_DURATION 100                    ' 100 ms warten
  NEXT
END                                      ' Ende Task MAIN
```

Documentation History

Version of Docmentation	Version of driver	Description / Changes
001	1.00a	First release