Application Note AN0902
Communication Options on bentrup Controllers

_____

1.) General Informations

Commonly customers using bentrup WinControl software package to support controllers operation. However, if you have to link the bentrup controller to an own application programme the following communication options are available:

A.) Communications via Windows DDE: Simple, easy to understand DDE commands being served by the bentrup DDE server driver which returns the data in a directly understandable syntax. Contact us to obtain a software license for windows DDE. Requirements: bentrup RS485 or RS232 interface on the controller and corresponding interface board on the PC, bentrup DDE server software

B.) Communications via low level data IO: send binary commands as defined by the bentrup low level IO protocol directly via RS232 or RS485. Requirements: bentrup RS485 or RS232 interface on the controller and corresponding interface board on the PC

C.) Communications via PROFIBUS: Use Windows Data Socket services to easily integrate communication in your application. The OPC server provide a variety of easy to use options. Features like worldwide controller access via the internet can be done easily. Requirements: PROFIBUS network and PROFIBUS interface on the bentrup controller (option on TC-S1 and TC-M2).

D.) Communications via Windows COM (Component Object Model): Contact us to checkout this new option

_____

A.) Communications via Windows DDE

A1.) General Informations

DDE is a standardized interface to exchange data between MS-Windows applications. bentrup supplies a DDE server that handles the entire data communication between PC and all bentrup units connected to the bentrup SYSTEMBUS. This frees the user from all timing critical questions, system errors and complex high-speed and multitasking operations on the SYSTEMBUS.

The user simply connects its own application as a DDE client and communicates with the bentrup unit(s) using easy-to-understand commands. This manual describes the complete command set; if you are

interested in basic technical informations about DDE refer to the *Microsoft Programmer´s Documentation for Windows.*

A2.) SYSBUS test CLIENT Example Programme

Togather with the bentrup SYSBUS DDE Driver comes an example programme. This programme is useful to become familiar with the syntax of the commands quickly. It provides the possibility to manual entering commands and monitoring actions and reply on-line.

The source code of the example programme is also included and may be helpful as a reference when you start with your application.

A3.) Loading the SYSBUS server

After the installation is completed start the SYSBUS server. On the SYSBUS server you get a display of the actual status and communications the server presently performs. The indicator [server] comes up if the DDE server registration completed successfully; the indicator [access] lights up while a task of a client is being performed. The [server action] window gives an abbreviated message what kind of request is actually processed. Refer to the window [connections] to check how many clients are presently connected to the server (max. 32). Finally, the window [status] reads the result of the last request performed, zero means success. For these SYSBUS Error Codes refer to appendix A.

A4.) Working with the test CLIENT

Load the test CLIENT. If the SYSBUS server has not beeing loaded yet, the CLIENT will do this automatically. Set the [ID] window to the ID of the bentrup unit you want to play with and [LogIn] the CLIENT (check the changes and reports on SERVER panel). Now, you can perform the commands listed below. Just enter the command in the upper left window and launch it by pressing the corresponding button. You can also load several test CLIENTs and login at the server either at the same or a different ID and then do task as you like.

A5.) Available DDE SYSBUS Commands

The bentrup DDE server supports the following set of DDE commands for its clients (for detailed description of each command refer to the the *Microsoft Programmer´s Documentation for Windows*).

- Connect To DDE Server (Called by the client to login with a certain ID)
- Client DDE Execute (to execute a command on the bentrup unit)
- Client DDE Read (to read a programme, process parameters or configuration) from the bentrup unit
- Client DDE Write (to write a programme or configuration to the bentrup unit)
- Disconnet From DDE Server (to logout from this connection)

Warm and Hot links are not supported because they are not useful for this type of communication. Since many DDE connections can be established simultaneously, it is not required to login/logout for each task. The connection can be established until the application programme quits.

A6.) SYSBUS DDE command syntax

A6.1.) Execute Commands

Following are the valid **executable** commands of the SYSBUS DDE server:

Command	Explanation	Example
RESET	triggers a HARDWARE reset on the unit	#RESET
REMOTE ON	sets the unit in REMOTE mode	#REMOTE_ON
REMOTE OFF	resets the REMOTE mode	#REMOTE_OFF
START	starts the presently selected programm	#START
STOP	interrupts programme execution	#STOP
SKIP	causes the unit to skip to the next segment	#SKIP
HOLD ON	sets the unit to programme hold	#HOLD_ON
HOLD OFF	resets the programme hold	#HOLD_OFF
ENTER IN-STALLATION	causes the unit to quit operation to be prepared to receive configuration data	#ENTER_INSTALL
LEAVE IN-STALLATION	causes the unit to quit the installation and to save the new data permanently	#LEAVE_INSTALL
PROGxx	Loads Programme Number xx	#PROG 85

A6.2.) Read Commands

Following are the valid **read** commands of the SYSBUS DDE server:

Command	Explanation	Example
UNIT DATA	returns system relevant informations: 0: manufacturer 1: model 2: version	#SY1 <returns TC-M1>
SYSTEM STATUS	returns the actual operation status, programme and segment number	#ST0 <returns #IDLE #PROG1#SEG00>
SEGM. TIME	returns the present segment time	#SL0 <returns #01:20:00>
INPUT VALUEx	returns the actual value of an input	#IN 0 <returns #23.25°C>
SETPNT VALx	returns the actual value of an setpoint	#SP 0 <returns #24.10°C>
CHANNEL x	returns the actual value of an channel	#CH 2 <returns #85.5%>
DIGITAL OUTPUT	returns the actual state of a set of 8 digital outputs, starting from lowest output, 8 outputs each per request	#DO00 <returns #10000000 which means DO0.0 on, all others off>

DIGITAL INPUT	returns the actual state of a set of 8 digital inputs, starting from lowest intput, 8 inputs each per request	#DI00 <returns #10000000 which means DI0.0 on, all others off>
ANALOG OUTPUT	returns the actual value of an analog output	#AO0.2 <returns #4.76%>
ANALOG INPUT	returns the actual value of an analog input	#AI0.2 <returns #20.45mV>
SERVOMOTORPOSITION	returns the position of an servomotor output	#SM 1 <returns #100.0&*> >
READ PROGRAMME	returns the value of one programme parameter, syntax #P progNo, segNo, parNo	#P 1 0 0 <returns #120 as the value of programme 1, segment 1, time>
READ CONFIGURATION	returns the value of the selected configuration value or its limits, syntax #I page,number, parameter, selection*)	#I 0 0 0 0 <returns #0>
GET LAST ERR	returns the error code of the last operation (same as displayed on server). This information is read out of the server, so no access to the unit is done	ERROR? (no preceeding "#" !!) <returns #ERR:22>

*) Reading configuration data: The 1st parameter specifies the type of data, e.g. 0=input configuration, 1=setpoint configuration etc. The 2nd parameter selects the number of this data record, ie. input number 1, 2, 3 an so on. The 3rd parameter, selects the parameter row. Please refer to the tables show in the manual.

The 4th parameter selects whether the actual parameter is requested (0), the type of value (1, see also appendix B), the lower value limit (2) or the upper value limit (3) is requested to be returned. These informations (1) to (3) must be used when entering data on the configuration menu ! Otherwise the user might enter values out of bounds which will not be recognized as faulty until send back to the unit.

A6.3.) Write Commands

Following are the valid **write** commands of the SYSBUS DDE server:

Command	Explanation	Example
WRITE PROGRAMME	writes the given value to the selected programme number / segment / param.	#P 1 0 1 <value #100> writes "100" to programme 1, seg 0, column 1
WRITE CONFIGURATION	writes the given value to the selected configuration type, number, parameter	#I 0 1 8 <value #-50> sets the "min value" of input config. #00 to -50
WRITE DIGITAL OUTPUT (DOx)	Switches the specified digital output to ON/OFF. Only allowed on DOx **not** used by the configuration	#DO1 4 <value #1> turns DO1.4 ON; requires firmware >= V 7.17
REMOTE	sets the selected setpoint to remote and	#S 0 <value #150.75>

SETPOINT	to the given value (set to any value < -10000 to switch back to automatic). Don´t forget to #START the controller !	sets the setpoint #00 to 150.75°C

A7.) Limits on Data Accesses

There are some limitations which data are allowed to be accessed . These limits are to avoid that data corruption occurs due to simultaneous access etc. Please refer to the following rules for reading and writing configuration and programme data:

- read / write programmes: The unit must be set to "REMOTE ON" before (this updates internal data and lockes the keyboard, as indicated by the little key appearing on the very left top of the TC-M1 window). After the read / write process, reset the unit to normal by "REMOTE OFF". This causes unlocking the keyboard as well as an update of the display if required.
- write configuration: The unit must receive the "ENTER INSTALLATION" command to quit usual operation (otherwise severe process faults may occur because of inconsistent configuration). The display prompts this command and will report the following accesses.

If you try to access without performing the required steps the access will be rejected and an error code is returned.

A8.) Multiple Requests

To save time, it is possible to chain commands of the same type (ie. EXECUTEs, READs or WRITEs) togather and commence them as one command. The single values, if required, are send / received in the same order as the commands. Up to 10 requests can be done as one job. This function is especially helpful for the cyclic repeating read of process parameters.

Example: #IN0#IN1#CH0#CH1

   returns the values of the inputs 0 and 1 and its control channel values as on string

#23.25°C#24.55°C#55.4%#43.1%>

A9.) Error Codes

The bentrup DDE server checks carefully all receiving requests and denies processing if there is any error. The error messages of the bentrup DDE server can be devided in 3 cathegories:

- Errors caused by a faulty syntax, parameter number or value (e.g. #ABC or #P 0 0). The error codes 16 to 21 point to errors that have been found by the bentrup DDE server before sending out any request on the SYSBUS
- Errors caused by a data communication problem, e.g the unit did not reply, the electrical connections were faulty or the ID did not fit. The corresponding error codes are 22 to 27
- Errors because the EXECUTE/WRITE/READ could not be completed successfully in the unit. These are the errors codes from 1 to 7.

Don´t mix up these error codes with the errors your DDE functions return ! If one of the above listed error occurs, the DDE function just returns a "NotProcessed" error message (native DDEML error). To get

the exact cause of the error, use the READ command "ERROR?" to get the SYSBUS error code from the DDE Server.

Error	Explanation
0	no Error
1	request not allowed (parameter out of actual bounds)
2	unauthorized programme access (read/write)
3	programme parameter write out of value limits
4	configuration parameter write out of value limits
5	bad command
6	unauthorized configuration write
7	reserved
16	bad write syntax command
17	bad read syntax command
18	bad execute syntax command
19	bad value syntax
20	no command specified
21	bad parameter
22	bad command reply
23	checksum error
24	no physical reply
25	missing command
26	reply not for ME
27	reply not from ID

A10.) Codes for Installation values

Some values of the configuration are coded because their information do not correspond to a certain values (e.g. mV range on input configuration: 0 stands for 50mV, 1 for 20mV ...) The data type value is returned on reading the configuration data (e.g. #I 0 0 0 1). Using the table following below you can convert the return into a meaningful text reading:

type value	conversion
10	-1=END, others as normal
11	3 digit value (8 bit)
12	5 digit value (16 bit)
13	value * 10 (e.g. 100 -> 10.0, 152 -> 15.2)
14	as 13
15	1 digit ASCII (e.g. 0x41 -> ´A´)
16	2 digit leading zero (e.g. 5 ->05)

17	weekdays: 0=MON, 1=TUE, 2=WED, 3=THU, 4=FRI, 5=SAT, 6=SUN
18	months: 0=JAN, 1=FEB etc.
19	reserved
20	0=OFF, 1=ON
21	0=S, 1=R, 2=K, 3=J
22	0=50mV, 1=20mV, 2=8mV
23	0=°C, 1=°F, 2=°K, 3=dg, 4=mV, 5=mA, 6=V%, 7=??, 8=%, 9=ev, 10=mb, 11=cb
24	-1=END, 0=T/C, 1=LIN, 2=ATM
25	0-19=I0-19
26	-1=END, 0=CON, 1=MOD, 2=CHA
27	-1=OFF, 0-19=SP00-SP19, 20-39=IN00-IN19, 40-59=VT00-VT19
28	-1=OFF, 0-19=IN00-IN19
29	-1=OFF, 0-19=SP00-SP19
30	-1=OFF, 0-19=CH00-CH19
31	-1=END, 0=PID, 1=C-PID, 2=D-HYS, 3=ATMOS, 4=SP-DIR
32	-1=END, 0=PRZ, 1=MOT, 2=LIM, 3=PRC, 4=EVE, 5=TAB, 6=OUT, 7=CMB
33	-1=END, 0=SPACE, 1=VAL, 2=VAL-B, 3=VALx2, 4=CHAN, 5=Fxx, 6=Axx, 7=MOT, 8=STAT, 9=MISC
34	-1=END, 0=TIM/S, 1=VALUE, 2=CHAN, 3=Fxx, 4=Axx
35	-1=END, 0=TIME, 1=VAL-L, 2=---, 3=VAL-S, 4=EVENT, 5=ATMOS
36	-1=END, 0=AVG, 1=MIN, 2=MAX, 3=SEL

B.) Communications via low level data IO

Following description refers to the low level data IO option only (without bentrup DDE server).

B1.) Electrical Definitions

The signal type complies to RS485 resp. RS232. Refer to the IEEE publications for details.

B2.) Communication Settings

Check controller manual for baudrate. Commonly TC500 and TC800 series controller operate on 38400 baud, TC-S1 / M2 on 115200 (resp. 96000 on units manufactured before year 2001). Each byte consists of 1 Start, 8 Data, Parity Even and 1 Stop Bit.

B3.) Data Syntax

Each transfer cycle consist of a request command sent to the bentrup unit which answers by returning the requested data (or confirmation).

The transfer protocol for both directions complies to the following syntax:

Byte #	Description
0	receiving device ID
1	sending device ID
2	length
3	data / command type*
4, 5, ...	variable data field*
n	checksum

Note: up to 10 data / command can be processed by one request. Since each data / command has a fixed number of bytes to follow the receiver can interpret the request. In this case the bytes marked with * are resent for each data / command. Only request with valid length and checksum are processed.

receiver ID
ID of the receiving unit. Usually when data are returned, the unit simply switches sender ID and receiver ID to mark the protocol as being returned to the unit which has initiated the transfer.

sender ID
ID of the sending unit. The host PC should always use ID63. The other members of the bus can use any number as adjusted in the unit configuration.

Length
Number of bytes following (without checksum byte). See example below.

Data / Command
Refer to chapter 5.) for a listing of all available commands. If the command / data request has been processed successfully the MSB of this byte is set by the processing device.

Checksum
All bytes of the protocol are summarized as 8 bit values without carry.

Examples:
Execute Command START is being sent from the PC to unit with ID0: HEX **00 3F 01 63 A3**
Double-Read Request of analog inputs IN0 and IN1 from ID0: HEX **00 3F 04 05 00 05 01 4E**

B4.) List of commands

Command	Param. IN	OUT	Description
00	1	8 (ASCII)	0 returns manufacturers name
			1 returns model name

				2 returns version number
				3 returns controllers serial number
01	1	4 (byte/ byte/ byte/ byte)		Status of process X (a=0): **Byte 0:** Bit 7:RUN 6:HOLD 5:AUTOTUNE 4:x 3:x 2:Process terminated due to error 1:process holded 0:slave OP. **Byte 1:** res. **Byte 2**: prog no. **Byte 3**: segment no.
02	1 (dummy)	4 (double word)		Returns remaining time in seconds
04	1 (dummy)	10 (float / float / byte / byte)		Mixer status: ratio, control factor, status, error code
05	1 (x)	7 (float / byte / byte)		INxx value, unit (see configuration), error status: Bit 7:error, 6:invalid, 5:underrun, 4:overrun 1:remote controlled 0:unreliable
06				as above, but with 2nd INxx value
07				as above, but with SPxx value
08	1 (x)	2 (byte / byte)		channel xx status output % (-127=-100% to 127=100%, error status: Bit 7:Error 6:overrun 5:gradient 4:inaktiv
09	1 (x)	1 (byte)		Bit pattern of digital outputs DOx.0-x.7
0A	1 (x)	6 (float / byte / byte)		analog output xx: value, status (refer to command 05), signal type (0:OFF 1:0-20mV 2:0-50mV 3:0-5V 4:0-10V 5:0-20mA 6:4-20mA 7:0-500R/2 wire 8:0-500R/3 wire
0B	1 (x)	2 (byte / byte)		servomotor xx info: position (0-255 = 0-100%), status (-1:backward 0:stop 1:forward movement)
0C	1 (x)	7		Infobox xx value (refer to command 5)
0D				As command 09, but digital inputs
0E				As command 0A, but analog intputs
20	3 (x / y / z)	2 (word)		Parameter of programme x + 1, segmemt y and columns z
30	5 (x/y/z/ pp)	1 (byte) 0=success or error code		Write parameter pp (word) to programme x+1, segment y, column z. Note: It is mandatory to write all programme parameters of a segment one by one in sequence completing with the last column (even if this is not to be changed)
34	4 (x/y/pp)	1 (byte) 0=success or error code		Set DOx.y to ON (pp=1)/OFF (pp=0) NOTE: only on firmware >=V 7.17
38	5 (x , pppp)	1 (byte) 0=success or error code		setpoint x set to float; to cancel mode setpoint of < -10000 must be sent

40	4 (x, y, z, t)	2 (word)	Value of configuration type x, page y, row z, t:0 value, t:1 data type, t:2 lower limit t:3 lower limit
50	5 (x, y, z, pp)	1 (byte) 0=success or error code	Write parameter pp to type x, page y, row z of the configuration
5E	1 (x)	STRING	Headline of the configuration type x including structural information
5F	3 (x, y, z)	STRING	Structural and text information of the entire configuration parameter (type x, row y, extra row z)
60	1 (dummy)	1 (byte) 0=success or error code	Resets the controller (same as power off / power on)
61	1 (dummy)	1 (byte) 0=success or error code	Remote mode ON -> must be activate to enable programme read / write
62	1 (dummy)	1 (byte) 0=success or error code	Remote mode OFF
63	1 (dummy)	1 (byte) 0=success or error code	Start the actual programme
64	1 (dummy)	1 (byte) 0=success or error code	Stop the actual programme
65	1 (dummy)	1 (byte) 0=success or error code	During a programme running, skip to the next segment
66	1 (dummy)	1 (byte) 0=success or error code	Activate HOLD
67	1 (dummy)	1 (byte) 0=success or error code	Deactivate HOLD
68	1 (dummy)	1 (byte) 0=success or error code	Enter Installation mode -> must be activated to enable configuration write
69	1 (dummy)	1 (byte) 0=success or error code	Leave and save configuration
78	1 (x)	1 (byte) 0=success or error code	Load programme number (x)

We reserve the right to change specifications at anytime without prior notification.

C.) Communications via PROFIBUS

For details on this option refer to EN50170 as well as OPC server description. As well you can contact us to obtain further informations on this powerful option.

Initial Issue	Aug 22th 1998	Be
Updated	Sept 15th 2002	Be
Updated COM and baudrates	July 3rd 2004	Be
Updated write DOx.y feature	Dec 17th 2004	Be

bentrup
INDUSTRIE*STEUERUNGEN*

Zum Buchwald 4   Telefon (06404) 9144-0
D-35463 Fernwald   Telefax (06404) 9144-11