



## **Group3 DDE Server**

### **User's Manual**

For use with the Group3 Control System

**G3DDE32.EXE V4.5b**

8 June 07



# CONTENTS

	<b>page</b>
<b>1. Installation</b>	
1.1 Loop Controller Installation	1-1
1.2 DDE Server Installation	1-2
1.3 The Configuration File, LINK.TAB	1-3
<b>2. Using G3DDE32</b>	
2.1 Tag Definitions - Item Names	2-1
2.2 CNA module - Control & Status flag bytes	2-4
2.3 Handling ASCII strings through a type F I/O board using DDE	2-5
2.4 GPIB commands - Using the Type K board with G3DDE	2-6
<b>3. Running Group3 Control using InTouch</b>	
3.1 Defining a Tagname – step-by-step procedure	3-1
Raw Data Ranges	3-3
Deadband	3-4
Tag/Item definitions for Stepper Motor - Type G board	3-5
Tag/Item definitions for the CNA module	3-6
3.2 Sending and Receiving ASCII Strings with INTOUCH	3-7
3.3 Using the GPIB (Type K) Board with InTouch	3-9



# 1 INSTALLATION

The basic requirements for running Group3 Control with DDE are:

- 1 Loop Controller card installed in the computer and memory drivers installed.
- 2 DDE server files copied to the hard disk, with program item properties set up.
- 3 A correctly defined Configuration file.

## 1.1 Loop Controller Installation

Install the memory access drivers from the “Loop Controller Driver Installation” CD provided: The drivers are required by G3DDE32.EXE (the Group3 DDE server), which in turn can be used by any DDE-aware application, including WonderWare InTouch and Microsoft Excel. The driver installation disk has an automated installer program: SETUP.EXE

The Loop Controller card may be plugged into any vacant slot in the computer.

The base address jumpers on an ISA card are set to CE00:0 at the factory.

The switch at the top of a PCI card is set to 0 at the factory.

If these settings need to be changed, refer to section 4 of the Group3 Control User's Manual.

## 1.2 DDE Server Installation

### 32 bit application, for Windows 98, ME, NT, 2000, and XP

- 1) Using File Manager or DOS Prompt, copy the file G3DDE32.EXE into a suitable directory (e.g.C:\Group3) of the computer's hard disk.
- 2) Copy the file WWCOMDLG.DLL from the CD to the same directory.
- 3) Set up an icon on the Desktop and alter its properties as follows:

Using File Explorer, right click the file G3DDE32.EXE, select “create shortcut”, and drag the highlighted file to the desktop.

Move to the desktop and right click the new icon, select “properties”, then “shortcut” and check them as follows:

Target:            *{full path to the executable file}*  
                    leave this at the default offered.

Start in:           *{defaults to the same as the Target}*  
                    however it should be changed to the full path of where the file LINK.TAB is/will be created.  
                    It is suggested that this should be the InTouch application directory.

Shortcut key: (none,  
                  or a special combination of Ctrl-Alt-<letter key> to start the server)

Check the `Run Minimised' box.

Back on the desktop, edit the text under the icon to indicate which InTouch application (and LINK.TAB) it is going to use.

- 4) Check that there is a file **CTL3D32.DLL** in the \windows\System(32) directory on the hard disk.  
If there isn't, copy the CTL3D32.DLL from the \DLL directory on the installation CD to the \windows\System(32) directory.

**IMPORTANT:** If your hard disk already has a copy of CTL3D32.DLL, and it has a more recent date than the copy supplied by Group3, then do not to overwrite it.

### 1.3 The Configuration File, LINK.TAB

The purpose of the LINK.TAB configuration file is to inform the DDE server about the Device Interfaces on the loop and the I/O boards they contain.

The DDE server sets up the Loop Controller to communicate on the fibre-optic loop.

Each Device Interface has its own unique address as set by its address switch. To satisfy the DDE server, the Device Interface addresses for each loop must start at 0 and be a contiguous set, 0, 1, 2, etc with a maximum possible value of 15 (F in hex, as set by the DI address switch).

Each Device Interface contains 1, 2, or 3 input/output (I/O) boards, and each is given its own unique address within the Device Interface by means of movable jumpers on the board. To satisfy the DDE server, the board addresses for each loop must start at 1 and be a contiguous set, 1, 2, 3 (as set by the board jumpers). The maximum possible number of boards in a DI is three, although a DI may contain fewer boards than this.

The configuration file, LINK.TAB, can be generated using any general purpose text editor which saves text files in straight ASCII code.

The file should be in the following format, with further explanation on the following pages. The column on the right represents the contents of LINK.TAB.

---

	Example:
<b>LOOP:</b> base address of Loop	LOOP: 0
<b>MODE:</b> SDLC or FAST SDLC	MODE: SDLC
<b>BOX:</b> name of Device Interface 0.	BOX: Ion Source
<b>CARD:</b> I/O board number 1	CARD: B
<b>CARD:</b> I/O board number 2	CARD: C
<b>CARD:</b> I/O board number 3	CARD: D

---

**LOOP:**

The value entered here should be the base address of the loop, as determined by the set of jumpers on the ISA loop controller card, or a number as determined by the switch on the PCI card.

For the PCI card the loop number should be specified as a number in the range 0 to 15, corresponding to the 0 to F setting on the switch on the top edge of the board.

With the LC3 three loop card, the loop number of the controller handling the port at the top of the board (furthest from the edge finger connector) is as set on the switch. The next controller, which handles the center port, has loop number one higher, and the third controller, whose port is nearest the edge connector, has a loop number 2 higher than the switch setting.

For the ISA card the base address should be specified as a four digit number, preceded by 0x to indicate hexadecimal. The LC determines the base address by comparing PC address lines A13 to A19 with the jumper settings. Thus the base address of the shared memory can be placed anywhere between 0000:0 hex and FE00:0 hex in increments of 0200:0 hex (8092 bytes).

With the LC3 three loop card, the base address of the controller handling the port at the top of the board (furthest from the edge finger connector) is as set on the jumpers. The next controller, which handles the center port, has base address 2048 bytes higher, and the third controller, whose port is nearest the edge connector, has base address another 2048 bytes higher. With the jumpers set for CE, as above, the base addresses of the three loops are CE00, CE80, and CF00.

**MODE:**

Mode should be followed by either SDLC or FAST SDLC.

If absent, the loop will default to SDLC.

The fibre-optic loop carries message packets to the DIs. The protocol is always SDLC running at 1.15Mbaud. Initially each message packet contained information to or from a single I/O board. With the introduction of DI and LC software version 4.3, there is an option to select a mode where a message packet contains information for all boards in a DI, not just a single I/O board. Not all boards can be combined, but in practice the loop speeds up by a factor of two. This mode should only be used if all DIs and the LC are running version 4.3 or greater.



**BOX:**

The name of the Device Interface following **BOX:** is arbitrary, but should be unique for each Device Interface on the loop.

**CARD:**

The type of the I/O card in the DI.

Card types are as follows (use the board type letter, or the name in the second column, not both):

<b>A</b>	<b>FAST_ANALOG</b>	type A board, Analog & Digital combined
<b>B</b>	<b>DIGITAL</b>	type B board, 24 Digital channels
<b>C</b>	<b>8_INPUT</b>	type C board, 8 Analog Inputs
<b>D</b>	<b>8_OUTPUT</b>	type D board, 8 Analog Outputs
<b>E</b>	<b>MOTOR</b>	type E board, 4 DC motor drivers
<b>F:g:g</b>	<b>SERIAL:g:g</b>	type F board, both serial ports in general mode
<b>F:M:n:g</b>	<b>SERIAL:M:n:g</b>	type F board, port 0 in Group3 teslameter mode, (n = number of teslameters), Port 1 in general communications mode.
<b>F:M:n:M:x</b>	<b>SERIAL:M:n:M:x</b>	type F board, both ports in teslameter mode, (n and x = number of teslameters),
<b>G</b>	<b>STEPPER</b>	type G board, 4 Stepper motor drivers
<b>H</b>	<b>ENCODER</b>	type H board, 4 Quadrature encoders
<b>J</b>	<b>2_OUTPUT</b>	type J board, 2 precision analog outputs
<b>K</b>	<b>GPIB</b>	type K board, IEEE488 / GPIB controller
<b>CNA</b>	<b>DI_A</b>	type CNA Device Interface, Analog & Digital

Some older LINK.TAB files may have FO\_BOARD as an alternative to SERIAL .

The sequence starting with BOX: is repeated as necessary in order of Device Interface address switch settings to describe all Device Interfaces in the system.

Comments beginning with a semi-colon (;) may be inserted on lines of their own or after file entries.

If the system contains more than one loop, the above sequence is repeated with the LOOP: entry for the next loop, with its base address, followed by the box and card specifications for this loop. Different loops can be set to other communications modes.

Blank lines and indentation with spaces and tabs are acceptable.  
Upper or lower case may be used anywhere.

A sample LINK.TAB file follows:

---

; Ion Source Control System

LOOP: 0 ; PCI loop controller for primary machine – loop 0  
MODE: FAST SDLC

BOX:GROUND  
CARD:B  
CARD:C  
CARD:D

BOX:HIGH\_VOLTAGE  
CARD:DIGITAL  
CARD:8\_INPUT  
CARD:SERIAL:G:M:2

BOX:MAGNET  
CARD:CNA

---

If the system contains more than one loop, the above sequence is repeated with the LOOP: entry for the next loop, then the box and card specification for this loop.

The process is repeated for all loops in the system.

## 2 Using G3DDE32

When using a DDE based application to bring in or send out information to Group3 Control modules each I/O channel or tag in the system is referenced via DDE by strings of the form:-  
application name / topic name / item name

The application name is **G3DDE32** (32 bit systems).

The DDE has previously been called CNETDDE and G3DDE16 (16-bit versions) and G3DDENT (32 bit version). If upgrading an existing system, these old names must be replaced with G3DDE32.

The topic name is **Group3Control**.

Pre 1999 versions used the topic name "ControlNet" - we have had to phase out the use of this name. The DDE server will however still support the use of "ControlNet" for older systems. New systems should use "Group3Control".

The Item name for I/O points is defined in the following section.

### 2.1 Tag Definitions / Item names

The format of the Item name is:

**Ln:a:b:t:c:d:p**

where

**Ln** = loop number, if more than one

**a** = Device Interface address (0-15)

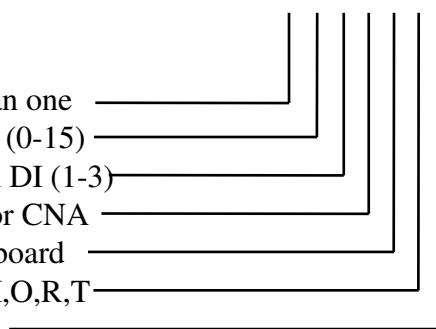
**b** = I/O board number within DI (1-3)

**t** = I/O board type, A to K, or CNA

**c** = channel number on that board

**d** = tag definition indicator, I,O,R,T

**p** = polarity / DTM address



**Ln** Loop number is denoted by **Ln** in the tag definition, where n = the number of the loop, starting with 0, in the order in which they appear in the configuration file. If the system contains only one loop, the L0: may be omitted from the tag definition.

**a** Device Interface Address, 0 to 15, as per rotary switch on front of DI.

**b** I/O board number, 1, 2, or 3.

Each Device Interface contains 1, 2, or 3 input/output (I/O) boards, and each is given its own unique address within the Device Interface by means of movable jumpers on the board.

The CNA is always addressed as board number 1

**t** I/O board type is the alpha character denoting board type.  
 Categories of boards within a type are not specified here, e.g. type C1 and C2 boards are both denoted as C.  
 A CNA module has **CNA** in this position.

**c** Channel number is the number of the channel within the I/O board, starting at 0.  
 If there is only one channel of that type, then a 0 should be put in this position.

**d** The value taken by this position varies according to the type of I/O board.

Analog channels (A,C,D,J boards & CNA)	<b>I</b> = analog input <b>O</b> = analog output
Digital channels (A & B boards, & CNA)	<b>R</b> = digital input <b>T</b> = digital output
DC motor driver (Type E board) use	<b>C</b> = dc motor control <b>S</b> = dc motor speed <b>A</b> = dc motor acceleration
Serial Communications (Type F board, General serial comm's: port type 0)	<b>I</b> = serial input string <b>O</b> = serial output string
Serial Communications (Type F board, Group 3 teslameter mode: port type 1)	<b>R</b> = range <b>U</b> = units <b>Z</b> = zero <b>F</b> = field reading <b>T</b> = temperature reading
Stepper Motor Controller (Type G board)	<b>M</b> = mode control <b>C</b> = control byte <b>O</b> = desired absolute step position <b>R</b> = maximum stepping rate <b>A</b> = stepping rate acceleration <b>I</b> = actual absolute step position <b>P</b> = analog input <b>D</b> = digital inputs
Encoder (Type H board)	<b>I</b> = input count
IEEE488 / GPIB controller (Type K board)	<b>I</b> = serial input string <b>O</b> = serial output string
CNA module	<b>C</b> = PID control flags <b>S</b> = Module Status flags

**p** This part of the Item Address is only used for some board types:

Analog boards (Types A,C,D,J & CNA) used for polarity indicator	<b>B</b> = bipolar <b>U</b> = unipolar.
Serial (Type F board, in teslameter mode)	used for the Teslameter address.

### Some Examples of Item Names:

- L2:0:1:B:22:R** defines the tag as belonging to the third loop, Device Interface address 0, board number 1, a Type B I/O board, channel 22, digital input.
- 1:2:F:1:I** defines the tag as belonging to Device Interface address 1, board number 2, a Type F I/O board, channel 1, input string. Loop0 assumed.
- 2:3:F:0:F:3** the tag belongs to Device Interface 2, board number 3, a Type F I/O board, channel 0, teslameter address 3, and will input the field reading.

Further examples of Item names:

A stepping motor on channel 0 of a type G board which is board number 1 in a Device Interface set to address 3 could have the following tags assigned to it.

- 3:1:G:0:M** mode control
- 3:1:G:0:C** if M is set to 1, this tag controls the motor
- 3:1:G:0:O** if M is set to 0, this tag sets the desired absolute motor position
- 3:1:G:0:R** this tag sets the maximum stepping rate
- 3:1:G:0:A** this tag sets the stepping rate acceleration
- 3:1:G:0:I** this tag reads in the current absolute motor position
- 3:1:G:0:P** this tag reads in the current value of the analog input on this channel
- 3:1:G:0:D** tag reads in the states of the digital channels as an 8-bit binary number

## 2.2 CNA module Control and Status flag bytes

The control and status flags are organised as a byte for each. The individual flags are bits within the byte, defined as follows:-

**Control Byte** flags - sent to the CNA module.

**Bit0** PID Reset - used to zero the analog output and clear all PID calculations.

If bit 0 is set to 1 then all of the following occur:

- the controller output is set to zero.
- the integral accumulation is cleared to zero.
- all PID calculations are stopped.

**Bit1** PID hold - used to hold the output and PID calculations.

If bit 1 is set to 1, then all of the following occur:

- the controller output is held at its last value.
- the integral accumulation is halted.

**Status Byte** flags - read from the CNA module.

All bits read as zero if all is running normally - this allows an application to test the whole byte, and take no further action if it is zero.

**Bit0** Analog Output - DAC over or under flow

Read as 1 if a number outside the range -32,000 to +32,000 (bipolar) or 0 to 64000 (unipolar) has been sent up to the module. The module adjusts the number to the most +ve or -ve value possible, as appropriate, and sets this overflow/underflow status flag.

**Bit1** Analog Input0 - overflow/underflow of ADC

**Bit2** Analog Input1 - overflow/underflow of ADC

Read as 1 if a number outside the range -32,000 to +32,000 (bipolar) or 0 to 64000 (unipolar) has been generated by the analog input circuitry.

**Bit3** PID calculation overflow.

Read as 1 if, in calculating the PID output, a number was generated that exceeded the limits as specified through the diagnostic port. The integral is clamped to either the upper or lower limit, as appropriate.

**Bit4** Simulations active.

Read as 1 if one or more of the module's channels is simulated.

**Bit5** Board needs calibrating.

Read as 1 if the calibration data has been cleared within the module.

## 2.3 Handling ASCII strings through a type F I/O board using DDE

With the Group3 DDE Server, G3DDE, ASCII character strings can be passed between your program and a Type F Serial Communications board. Strings can be sent from your program to the fibre-optic transmitter on a Type F board in a Device Interface. Similarly, strings received at the Type F board fibre-optic receiver are returned to your program.

Strings are transmitted between the Loop Controller and Device Interfaces in bursts of up to 29 characters in length. However, as far as the user is concerned, strings of unlimited length may be received by the fibre-optic port and sent back to you. Transmission over the loop from the Device Interface is initiated when either 29 characters have been received, when a string terminator (as defined in the Device Interface through the diagnostic port) has been received, or a two-character-width timeout has occurred (as enabled in the Device Interface through the diagnostic port). Strings sent out by you to the Device Interface may also be of unlimited length.

Full XON/XOFF flow control is implemented at the Type F board fibre-optic port if enabled through the device interface diagnostic port. When the port is transmitting a string to an external device, the latter may send an XOFF character (octal 023) to the receive port in order to halt data flow. An XON (octal 021) will restart data flow. If the fibre-optic port is receiving a string from an external device and its 1024 character buffer reaches the two-thirds full point, an XOFF will be sent from the fibre-optic transmitter for each character received thereafter, until the buffer has been emptied to the one-half full point. Then an XON character is sent.

All the printing ASCII characters (octal 40 to 177) are handled by the DDE server in literal form as text, except for the backslash character, \ . Because the backslash character is used to indicate a special character (control, hex or octal), to actually send a ' \' character requires special techniques. There are three possible strings:- either send \\ (two backslashes), or \134 (an octal string), or \x5C (a hex. string).

Any character can be handled in terms of its octal code in the form \O , where O is a string of three octal characters. For example, 'ACK' would be sent as \006 , and 'ACK' + 'return' as \006\015 . Control characters received at the F board fibre-optic port are returned from the DDE server in this form.

Characters can also be handled in hexadecimal code in the form \xH or \XH , where H is a string of two hexadecimal characters.

The octal and hexadecimal forms are generally useful for sending the non-printing or control characters \000 through \037 , or \x00 through \x1F . The following control characters can be sent in abbreviated form, as defined below:

<u>string</u>	<u>character</u>	<u>string</u>	<u>character</u>	<u>string</u>	<u>character</u>
\a	bell	\n	linefeed	\v	vertical tab
\b	backspace	\r	carriage return	\t	horizontal tab
\f	formfeed			\\	backslash

## 2.4 GPIB commands - Using the Type K board with G3DDE

Commands and data are passed to the K board in a manner similar to that used to operate the Serial communications (Type F) board. Command strings are assembled as a DDE message, and then the Group3 system passes the strings up to the DI with the Type K board. The DI interprets the commands and puts the correct signals onto the GPIB lines.

The commands that can be sent to the GPIB, and the responses are detailed below.

Typically, to get a measurement from an instrument on the GPIB, one has to write out a command to that device (a command that the instrument understands), asking for the value of the measurement, then perform a read operation to that instrument to bring back the data, specifying the number of bytes required. These commands can be concatenated and sent as a single string if required.

### GPIB Commands for the K board

Throughout this list the following conventions are followed:-

The standard abbreviation for the GPIB command is shown in **BOLD** type.

Descriptions in *<italics>* should be replaced with the appropriate number/name/data.

*<device address>* is either 4 hex digits, or "dev0"..."dev7"

Any parameters enclosed in square brackets, [ ], are optional - not always necessary. The brackets are not to be put in the command string.

\r = carriage return, <CR>, hex 0D, shown in input string as octal \015

\n = line feed, new line, <LF>, hex 0A, shown in input string as octal \012

Command strings are shown with spaces for clarity - the spaces are not necessary for correct responses from the K board.

Frequently the responses from the K board have leading zeros suppressed - thus if changing the EOS character from <CR> (0d hex) the response is just \rd\n not \r0d\n





**TMO** *<new timeout value>* [, *<device address>* ] \r

Changes the default timeout value for the controller, if no address given.  
if device address is given, changes timeout used for that device.

*<new timeout value>* is a decimal number in the range 0 to 15.

A value of 0 disables the timer, 1-15 produces a timeout time of:

value:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
time:	35us	70us	278us	556us	2ms	8.7ms	35ms	72ms	284ms	569ms	2.2s	8.7s	37s	74s	290s

Response: \r *<old tmo value in hex>* \n

**ONL** *<either of: "on" or "1", or "off" or "0">* \r

Takes controller on or off line. eg. **ONL 0** \r takes the controller offline.  
Resets all if placed online.

Response: \r \n

**CLR** [ *<device address1>* [, *<device address2>* [...]] \r

Clears the specified devices.

If no device addresses given, then sends a DCL.

Response: \r \n

**CMD** *<count of number of bytes>* \r *<data>* \r

Send Command messages, sends IEEE488 commands onto the GPIB.

Example: **CMD 1** \r ? \r sends the UNLISTEN command (ASCII 3F)

Response: \r *<count of bytes received, in hex>* \n

**LINES** \r

Reports the status of the GPIB control lines.

Response: \r *<the lines>* \n

where *<the lines>* is a hex word made up as:-

E0I,ATN,SRQ,REN,IFC,NRFD,NDAC,DAV forming the upper byte,

and a mask of which of these are valid in the lower byte.

(DAV is not valid if one of many listeners, and,

if not CIC, only get to see this boards SRQ)

**LN** *<device address1>* [, *<device address2>* [, ...]] \r

Find the listening devices on the GPIB

Response: \r *<l1>* [, *<l2>* [,...]] \n

where *<l1>*, *<l2>* .. are hex addresses of the listeners found.

**LOC** [ *<device address1>* [, *<device address2>* [,...]] \r

set devices to local mode,

addresses devices to listen, then sends GTL (Go To Local) command.

If no device addresses are given, and:

the K board is a controller, then deassert REN

the K board is not a controller, try to set it to local.

Response: \r \n

**TRG** [ *<device address1>* [, *<device address2>* [...]] \r  
Triggers the listed devices, or those already addressed - issues a GET.  
Response: \r \n

**RSP** *<device address1>* [, *<device address2>* [...]] \r  
Request Serial Poll  
Response: \r *<p1>* [, *<p2>* ,[.]] \n  
where *<p1>*, *<p2>* are two digit hex responses from the devices polled.

**RSV** *<status byte>* \r  
Change the serial poll status byte.  
Response: \r *<old status byte>* \n

**IST** *<either of: "on" or "1", or "off" or "0">* \r  
Change the parallel poll response bit.  
Response: \r *<old IST>* \n

**PPC** *<pp bit>*, *<pp sense>* [, *<device address>* ] \r  
Parallel Poll Configure. Configure board or device.  
*<pp bit>* is which of the 8 DIO lines this device will respond on.  
*<pp sense>* is the sense of the signal put on the line selected by *<pp bit>*  
if = 1, then will respond with a "1" on the selected line.  
Response: \r \n

**PPU** [ *<device address1>* [, *<device address2>* [...]] \r  
Parallel Poll Unconfigure.  
If no addresses given and:-  
the K board is the controller then send PPU.  
the K board is not a controller, unconfigure it.  
Response: \r \n

**RPP** \r  
Request a Parallel Poll.  
Response: \r *<hex byte from poll>* \n

**CAC** *<either of: "on" or "1", or "off" or "0">* \r  
Become Active Controller.  
Response: \r \n

**GTS** *<either of: "on" or "1", or "off" or "0">* \r  
Go To Standby controller state.  
Response: \r \n

**SIC** [ *<time>* ] \r  
Send Interface Clear. for 1..2ms, also makes K board system controller.  
Response: \r \n

**SRE** <either of: "on" or "1", or "off" or "0"> \r  
Set/reset the Remote ENable (REN) line on the bus  
Response: \r 1 \n if REN was set  
          \r 0 \n if REN was not set

**RWLS** <device address1> [, <device address2> [,...]] \r  
Go to Remote, with lockout command to the addressed devices.  
Response: \r \n

**REM** [ <device address1> [, <device address2> [,...]]] \r  
Go to Remote command to the addressed devices.  
If no devices specified, then REN bus line is asserted.  
Response: \r \n

**ECHO** <either of: "on" or "1", or "off" or "0"> \r  
Turn echo and prompt on or off. eg. **ECHO 1** \r echoes commands.  
Response: \r \n

**STAT** \r  
Returns current status of the interface, composed of the status word (ibsta) and the error variable (iberr).  
Response: \r <hex word of (IBSTA)> , <hexword of (IBERR)> \n

**ADDR** <device address (dev0 ..dev7)> , <new address (hex word)> \r  
ADDRESS change of device dev0....dev7.  
Changes address held in memory.  
No change if SAD (secondary address) = 0.  
No change if PAD (primary address) = -1.  
Response: \r <old address as a hex word> \n

**CADDR** <address (in hex)> \r  
Controller ADDRESS change. Changes address of controller  
Response: \r <old address in hex> \n

### **General comments on the use of the type K board.**

At the start of a measurement cycle the devices on the GPIB to be used should be put in a known state. This is most easily done by issuing a **DCL** (Device CLear) to the devices to be used.

Before sending data and instructions to devices on the GPIB it is advisable to send an **UNL** (UNListen) command to inhibit all current listeners on the bus. ( **CMD 1 \r ? \r** )

Also, it is highly recommended that a sequence that causes an instrument to be addressed as a talker should be terminated with an Untalk command once the data has been collected. ( **CMD 1 \r \_ \r** ).

## 3 Running Group3 Control using InTouch

This section contains instructions specific to defining tags for the graphics software package called InTouch, by WonderWare Corporation.

It is generally easier and more efficient to define all the I/O points or tags at the start, before starting to draw control screen graphics etc.

You will need the list of all I/O points, along with polarities and ranges that you defined when initially planning the system. This has probably evolved a bit from the list you ordered the hardware from, so maybe it would be best to use the channel by channel set up record you generated when first configuring the DI.

### 3.1 Defining a Tagname - step-by-step procedure

Define application directory

Start Windows.

Double click on the WonderWare icon, to select the application directory.

Create a directory for your application. If a directory for the application already exists, confirm that this directory is highlighted as the current application directory. If necessary, select from the directories listed in the directories window.

Select Run WindowMaker. Click on OK. WindowMaker now starts.

At the opening screen of WindowMaker, click on Special on the menu bar.

Select Tag Name Dictionary to bring up the Tag Name Definition screen.

Click on New to define a new tag.

Enter a name for the new tag.

You may have settled on names for each I/O point, but there is a balance between making the name self explanatory but long, or cryptic and short. Some packages have restrictions on name length, and/or on the length of algebraic expressions written using the tagnames. Use the shortest name possible without getting too cryptic. Remember that every tag in the system will need a unique name.

Enter the Type of tag. For external tags (tags to or from Group3 Control module) choose from:

DDE discrete (for digital channels)

DDE integer (for analog channels)

DDE real (for Group3 teslameter readings via Type F I/O board)

DDE message (for general ASCII strings via Type F or K boards)

The DDE access name is initially unassigned. Click on its button, and select New.

The DDE Access Name Definition window appears.

The DDE Access Name Definition screen should be completed as follows:

The DDE Access Name is Group3Control

The DDE Application/Server Name is G3DDE32

The DDE Topic Name is Group3Control

The above DDE access, application/server, and topic names now apply to all tags.

Check "Request Initial Data" & "Advise Only Active Items" at the bottom of the window.

Click on OK, and return to the Tag Definition screen.

In the Item field, enter the Item Address Definition, which is in the form: **a:b:t:c:d:p** as defined earlier in this document.

Fill out the rest of the Tag Definition screen as appropriate. Press F1 for context sensitive help from InTouch.

If Tags are to be for values output from Group3 Control modules, then they will need to have the button labelled **Read/Write** checked. If the tag is to be for readings of values measured by Group3 Control modules then they can be made Read Only .

Avoid checking the buttons **Log Data**, **Log Events** or **Retentive** unless it is really necessary for the application - Each of these causes disk access and extra processing overhead which can slow a system down if many tags change at once.

Make sure you define each I/O point with the same polarity and range settings as you recorded from the diagnostic port. Remember that the raw number for analog channels, as found in the loop controller, can have different values depending on the polarity setting of that channel.

The data scaling that takes place within InTouch is determined by the min/max EU and min/max Raw figures.

The EU figures are the range of values that will be used and displayed within InTouch. The Raw figures are the range of values measured/generated by Group3 modules, and are detailed on next page for each I/O board type.

As an example of the automatic scaling that InTouch does to tag values take the case of a 0 to 10 volt signal being measured by a type C board input set on Unipolar range of 0 to 10 volts.

The **min EU** would be set to 0, the **Max EU** set to 10, with the corresponding **Min Raw** set to 0, and the **Max Raw** set to 64000. InTouch will then automatically scale the 0 to 64000 raw input values to be in the 0 to 10 range of the EU values.

Note that Group3 Control uses a range of 64,000 for 16 bit values, not the possible maximum of 65,536 allowed by 16 bits. The Group3 modules use the extra "headroom" between 64,000 and 65,536 for internal calibration calculations. The values sent over the loop should remain within the 64,000 range.

## Raw Data Ranges

		<b>Range</b>
<b>A board</b>		
Analog inputs 16 bit	unipolar	0 to 64000
	bipolar	-32000 to +32000
Analog output 14 bit	unipolar	0 to 16000
	bipolar	-8000 to +8000
<b>B board</b>		
all digital - no data ranges.		
<b>C board</b>		
Analog inputs 16 bit	unipolar	0 to 64000
	bipolar	-32000 to +32000
<b>D board</b>		
Analog outputs 14 bit	unipolar	0 to 16000
	bipolar	-8000 to +8000
<b>E board</b>		
DC motor speed 8 bit		0 to 255
DC motor acceleration		0 to 255
DC motor control		0 to 3
<b>F board</b>		
General Serial - DDE Message - max. length 29 bytes		
Teslameter mode:		
Range = DDE Integer		0 to 3
Field reading = DDE Real		-1000000 to +1000000
Temperature reading = DDE Real		0 to 200
Units = DDE Message - max length 29 bytes, takes	G (gauss) or T (tesla)	
Zero = DDE Discrete, a value of 1 will zero the teslameter		
<b>G board</b>		
Analog inputs 8 bit	unipolar	0 to 255
Digital inputs - 8 bits in one		0 to 255
Positions 32 bit		-2147483648 to +2147483647
Acceleration		0 to 255
Speed (stepping rate)		0 to 5000
<b>H board</b>		
Encoder counts		-32768 to +32767
<b>J board</b>		
Analog outputs 16 bit	unipolar	0 to 64000
	bipolar	-32000 to +32000
<b>K board</b>		
GPIB message = DDE Message - max. length 29 bytes		
<b>CNA module</b>		
Analog inputs 16 bit	bipolar	-32000 to +32000
	unipolar	0 to +64000
Analog output 16 bit	bipolar	-32000 to +32000
	unipolar	0 to +64000
PID Control and Module Status bytes:-		0 to 255, (bit-wise access required)

## **Deadband**

The Windows DDE interface to InTouch can get temporarily overloaded if a lot of channels all change at once. Unless great care is taken in installation the 16 bit analog inputs are bound to have some noise and flutter on them. This can unnecessarily overload the system, particularly if they are measuring slow changing parameters such as temperatures or magnet currents. For these channels, and in general unless the full speed accuracy is required, use the filtering within the DI, and use the deadband system within InTouch.

Set the deadband to a reasonable figure unless very high accuracy is required. A deadband value of 100 on a C board input still allows readings accurate to 0.2%, and most process variables are not measured to that accuracy.

Do not check the "use tagname as Item Name" box.

Click on Save, then Done when all tags are completed.





## Tag / Item definition for the CNA module

The format of the Item name is: **Ln:a:b:t:c:d:p** , as described before.

**a** = module address

**b** = board number, is always **1** for a CNA

**t** = board type, is always **CNA**

**c** = channel number, and should be **0** if there is only one channel of that type.

**d** = varies according to the type of channel

**I** for analog input

**O** for analog output

**R** for digital input

**T** for digital output

**C** for the PID control flags byte (read/write)

**S** for the module status flags byte (read only)

**p** = polarity of analog, which is **B** (bipolar) or **U** (unipolar)

To access the states of the individual flags within the flag bytes it is necessary to use the InTouch bit manipulation instructions. Thus if some item on the display screen was meant to change colour according to the state of bit 4 of the status byte (indicating that a simulation was active within the CNA module), then the colour of that item should be controlled by the expression *<Tagname used for status byte>.4*

## 3.2 Sending and Receiving ASCII Strings with INTOUCH

ASCII character strings can be sent both ways between InTouch and Group3 DIs . Strings are uplifted from the InTouch database and sent from the fibre-optic transmitter on a Type F Serial Interface board in a Device Interface. Similarly, strings received at the Type F board fibre-optic receiver are returned to the InTouch database.

Strings are transmitted between the Loop Controller and Device Interfaces in bursts of up to 29 characters in length. However, as far as the user is concerned, strings of unlimited length may be received by the fibre-optic port and sent back to InTouch. Transmission over the loop from the Device Interface is initiated when either 29 characters have been received, when a string terminator (as defined in the Device Interface through the diagnostic port) has been received, or a two-character-width timeout has occurred (as enabled in the Device Interface through the diagnostic port). Strings sent out by you to the Device Interface may also be of unlimited length.

Full XON/XOFF flow control is implemented at the Type F board fibre-optic port. When the port is transmitting a string to an external device, the latter may send an XOFF character (octal 023) to the receive port in order to halt data flow. An XON (octal 021) will restart data flow. If the fibre-optic port is receiving a string from an external device and its 1024 character buffer reaches the two-thirds full point, an XOFF will be sent from the fibre-optic transmitter for each character received thereafter, until the buffer has been emptied to the one-half full point. Then an XON character is sent.

All the printing ASCII characters (octal 40 to 177) are handled by InTouch and the DDE server in literal form as text, except for the backslash character, \ . Because the backslash character is used to indicate a special character (control, hex or octal), to actually send a '\ ' character requires special techniques. There are three possible strings:- either send \\ (two backslashes), or \134 (an octal string), or \x5C (a hex. string).

Any character can be handled in terms of its octal code in the form \O , where O is a string of three octal characters. For example, ACK would be sent as \006 , and ACK + return as \006\015 . Control characters received at the F board fibre-optic port are returned to InTouch in the same form.

Characters can also be handled in hexadecimal code in the form \xH or \XH , where H is a string of two hexadecimal characters.

The octal and hexadecimal forms are generally useful for sending the non-printing or control characters \000 through \037 , or \x00 through \xx1F . The following control characters can be sent in abbreviated form, as defined below:

<u>string</u>	<u>character</u>	<u>string</u>	<u>character</u>	<u>string</u>	<u>character</u>
\a	bell	\n	linefeed	\v	vertical tab
\b	backspace	\r	carriage return	\t	horizontal tab
\f	formfeed			\\	backslash

The string to be sent from InTouch can be a text string, or be made up from a number of commands and/or tag values which have been joined together in an InTouch script.

To convert a numerical tag value to a string value, use the TEXT function available in the script editors. The format to produce a string to represent a tag *analog\_value* is:

TEXT( *analog\_value* ,"#")

This string can be joined to device address and other commands/values within an InTouch script. For example, to send a string to program an ASCII controlled device, a sequence such as the following may be required:

<address> <return> <command> <analog value> <return>  
A2        \r        W        *analog\_value*        \r

The analog value can be generated from an InTouch screen by creating a variable output slider to change the tag named *analog\_value* . A Data Change script should be created, activated by a change in *analog\_value* , that assembles *analog\_value* into a string to be stored in another tag, *outstring* , which is the final string to be sent to the controlled device. *outstring* should be defined as a DDE message, with item address defined as the Type F board's InTouch address.

The script would need to have the following form:-

*outstring* = A2\rW + TEXT( *analog\_value* ,"#") + \r

The ASCII sequence *outstring* will then appear at the DI Type F board fibre-optic transmitter with a suitably updated *analog\_value* embedded in it, each time *analog\_value* is changed.

Depending on the number of tags available, the process of generating strings can be made even more general, with commands derived from memory message tags that can be changed on-screen. If these tags are defined as memory messages, then they can just be added together in the script. The TEXT function is only necessary to convert numerical values of tags to ASCII strings.

Note that there is a feature of InTouch that is designed to reduce "unnecessary " DDE traffic that can cause a bit of a problem while sending strings. InTouch will not resend a DDE message string if the message has not changed - effectively it says "that message is the same as the last one, so I'll save communications overhead and not repeat it". In theory one ought to be able to repeatedly send the same command string to get a series of readings, but InTouch will only send the string once.

So to get repeated readings of the same parameter from the same instrument one must ensure that the DDE message changes. There are several ways to do this.

1) Split a single line command up into two strings,

If these two strings are sent consecutively to the DDE message out tag, then the value of the message tag changes between successive reads.

2) Send a null string out before or after the desired string to ensure the value of the message tag changes between successive reads.

3) Make sure that you read or write to another instrument, or to a different section of the same instrument before requesting another read of the parameter. This ensures that the value of the tag DDE\_OUT (or whatever name you choose for the F board tags) changes between reads, and so the updated message string will always be sent out.

### 3.3 Using the GPIB (Type K) Board with InTouch (IEEE488/GPIB Controller)

Commands and data are passed to the K board in a manner similar to that used to operate the Serial communications (Type F) board. Command strings are assembled in InTouch as a DDE message, then Group3 Control passes the strings up to the DI with the Type K board. The DI interprets the commands and data, and puts the correct signals onto the GPIB lines.

Typically, to get a measurement from an instrument on the GPIB, one has to write out a command to that device (a command that the instrument understands), asking for the value of the measurement, then perform a read operation to that instrument to bring back the data, specifying the number of bytes required. These commands can be concatenated and sent as a single string if required.

The commands that the Type K board responds to are detailed in section 2.

When using InTouch the different commands (RDDEV1, WRTDEV2\rT\n etc.) can be assigned to different memory tags, and the outgoing DDE message tag made up by joining the appropriate memory tags together in a script.

An entire write / read data sequence can be sent as one string.

For example **WRTDEV1\rT\rRD9,DEV1\r**.

sends the character T to device1 (requesting a Temperature reading, say) then addresses device1 requesting 9 bytes of data.

Note that there is a feature of InTouch that is designed to reduce "unnecessary " DDE traffic that can cause a bit of a problem while sending strings. InTouch will not resend a DDE message string if the message has not changed - effectively it says "that message is the same as the last one, so I'll save communications overhead and not repeat it". In theory one ought to be able to repeatedly send the above command string to get a series of temperature readings, but InTouch will only send the string once.

So, to get repeated readings of the same parameter from the same instrument one must ensure that the DDE message changes. There are several ways to do this.

1) Split the single line command up into two strings,  
WRTDEV1\rT\r and RD9,DEV1\r.

If these two strings are sent consecutively to the DDE message out tag, then the value of the message tag changes between successive reads.

A button or timer script could be written to send out the messages as:-

```
DDE_OUT = "WRTDEV1\rT\r";  
DDE_OUT = "RD9,DEV1\r";
```

2) Send a null string out before or after the desired string to ensure the value of the message tag changes between successive reads.

A button or timer script could be written to send out the messages as:-

```
DDE_OUT = "WRTDEV1\rT\rRD9,DEV1\r";  
DDE_OUT = "";
```

3) Make sure that you read or write to another instrument, or to a different section of the same instrument before requesting another read of the parameter. This ensures that the value of the tag DDE\_OUT (or whatever name you choose for the K board tags) changes between reads, and so the updated message string will always be sent out.

# GROUP3 TECHNOLOGY LTD

## LIMITED WARRANTY

Group3 Technology Ltd. (hereinafter called the Company) warrants instruments and other products of its manufacture to be free from defects in materials and workmanship that adversely affect the product's normal functioning under normal use and service for a period of one year from the date of shipment to the purchaser.

The obligation of this warranty shall be limited to repairing or replacing, at the discretion of the Company and without charge, any equipment which the Company agrees is defective as set out above within its warranty period. The Company will reimburse lowest freight rate two-way charges on any item returned to the Company's factory or any authorised distributor or service centre, provided that prior written authorisation for such return has been given by the Company.

This warranty shall not apply to any equipment which the Company determines to have become defective owing to mishandling, improper installation, alteration, negligence, inadequate maintenance, incorrect use, exposure to environmental conditions exceeding specifications, or any other circumstance not generally acceptable for equipment of a similar type.

The Company reserves the right to make changes in design without incurring any obligation to modify previously manufactured units.

No other warranties are expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company is not liable for consequential damages.

# Group3 Technology Ltd - DISTRIBUTORS & REPRESENTATIVES

## European Region

### United Kingdom

#### **Pulse Power & Measurement Ltd**

65 Shrivenham Hundred Business Park  
Watchfield, Swindon, Wiltshire, SN6 8TY, UK  
Tel: +44 (0)1793 784389 Fax: +44 (0)1793 784391  
email: sales@ppm.co.uk website: www.pppower.co.uk

### **Denmark, Sweden, Norway, Finland, Iceland, Belgium, Holland, Italy, Turkey, Russia, India** **Danfysik A/S**

Møllehaven 31, P.O. Box 29, DK-4040 Jyllinge, Denmark.  
Tel. +45 4679 0000 Fax +45 4679 0001 Contact: Erik Steinmann  
email: es@danfysik.dk website: www.danfysik.com

### **Germany, Poland, Czech & Slovak Republics, Ukraine**

#### **Schaefer Technologie GmbH**

Mörfelder Landstrasse 33, D-63225 Langen, Germany.  
Tel. +49 6103 30098-0 Fax +49 6103 30098-29 Contact: Martin Schaefer  
email: info@schaefer-tec.com website: www.schaefer-tec.com

### **Switzerland, Austria**

#### **Schaefer-Tec AG**

Badimatte 21, Postfach 431, CH-3422 Kirchberg, Switzerland  
Tel. +41 34 423 70 70 Fax +41 34 423 70 75 Contact: Martin Bossard  
email: ch@schaefer-tec.com website: www.schaefer-tec.com

### **France, Spain, Portugal**

#### **Schaefer-Techniques Sarl**

1, Rue du Ruisseau Blanc, F-91620 Nozay, France  
Tel. +33 1 6449 6350 Fax +33 1 6901 1205 Contact: Christophe Dubegny  
email: info@schaefer-tech.com website: www.schaefer-tech.com

### **Italy**

#### **Schaefer Italia SRL**

Via Minzoni, 57, I-45100 Rovigo, Italy  
Tel. +39 0425 460 218 Fax +39 0425 462 064 Contact: Paulo Bariani  
email: italia@schaefer-tec.com website: www.schaefer-tec.com

## **China**

#### **MT Electronic Co. Ltd.**

Room 503, No.24 Building Jing Tong Yuan, Sunny Uptown International Department,  
Chao Yang District, Beijing, China 100024  
Tel./Fax +86 10 6570 0095, mobile: +86 130 0116 1549, Contact: Liang Qing (Rosalind)  
email: lqrosalind@yahoo.com.cn website: www.mt-elec.com

## **India**

#### **Transact India Corporation**

5/1A, Grants Building, Arthur Bunder Road, Colaba, Mumbai 400 005, India  
Tel. +91 22 2285 5261, or 2283 4962 extn 22, or 2202 8735 Fax +91 22 2285 2326  
email: sales@transact.co.in Contact: Arish Patel arish@transact.co.in direct dial +91 22 563 64866

## **Israel**

#### **Scientific Products & Technology 3000 Ltd.**

P.O. Box 1425, Rosh Ha'Ayin 40850, Israel  
Tel. +972 3 901 4479 Fax +972 3 901 4481 Contact: Rafael Thaler  
email: info\_spt@netvision.net.il website: www.spt.co.il

## **Japan**

#### **Hakuto Company Ltd., Scientific Equipment Department,**

1 - 13, Shinjuku 1-chome, Shinjuku-ku, Tokyo 160-8910, Japan  
PO Box 25 Tokyo Central 100-8691  
Tel. +81 3 3225 8051 Fax +81 3 3225 9011 website: www.hakuto.co.jp  
Contact: Mr Tsugio Saitoh email: saito-tsugio@hakuto.co.jp  
Contact: Mr Shunsuke Takahashi email: takahashi-shunsuke@hakuto.co.jp

## **United States & Canada**

#### **GMW Associates - magnets, magnetic instrumentation, control systems**

955 Industrial Road, San Carlos, CA 94070.  
P.O. Box 2578, Redwood City, CA 94064, U.S.A.  
Tel. +1 650 802 8292 Fax +1 650 802 8298 Contact: Brian Richter  
email: brian@gmw.com website: www.gmw.com

#### **VI Control Systems - LabVIEW programming, control systems**

2173 Deer Trail, Los Alamos, NM 87544.  
Tel. (505) 662 1461 Fax (866) 422 2931 Contact: Neal Pederson  
email: np@viconcontrols.com website: www.viconcontrols.com

## **Manufacturer**

#### **Group3 Technology Ltd.,**

2 Charann Place, Avondale, Auckland 1026, New Zealand.  
P.O. Box 71-111, Rosebank, Auckland 1348, New Zealand.  
Tel. +64 9 828 3358 Fax +64 9 828 3357  
email: info@group3technology.com website: www.group3technology.com