



**Group3 Control  
LabVIEW Driver for Windows  
Version 5.0  
User's Manual**

**Includes support for ISA and PCI Loop Controllers  
and Windows® 98/ME/NT/2000/XP**

Windows is a registered trade mark of Microsoft Corporation.

6 June 2007

Thank you for purchasing and using Group3 Control equipment. We hope you will join the growing number of people who are enthusiastic about the features Group3 Control has to offer.

Group3 has been designing and building specialised control systems since 1983. We are constantly upgrading and improving our products and the supporting documentation. We welcome input from our customers, so if there are aspects of the system you particularly like, or things you would like to see implemented, improved or developed in the way of hardware, software or documentation please contact your Group3 representative, or Group3 directly with your suggestions.

The Group3 website ( <http://www.group3technology.com> ) contains some pages of information on Group3 products. This site will be updated regularly, and you may choose to check it from time to time, to learn about recent developments.

Listed on the "What's New" page are the recent developments and enhancements to the product range, grouped by the year of introduction.

The page "User Technical Notes" lists current versions of software, and also details past versions, with a brief list of the features they introduced.

Group3 Technology Ltd.  
2 Charann Place, Avondale, Auckland 1026  
PO Box 71-111, Rosebank, Auckland 1348, New Zealand.  
Phone: +64 9 828 3358  
Fax: +64 9 828 3357  
Email: [info@group3technology.com](mailto:info@group3technology.com)  
Web: <http://www.group3technology.com>

# Group3 Control LabVIEW Driver User's Manual

## CONTENTS

	page
<b>1. Overview and upgrade notes</b>	<b>1-1</b>
<b>2. Loop Controllers – Hardware Description</b>	<b>2-1</b>
2.1 PCI bus Loop Controllers	2-1
2.2 ISA bus Loop Controllers	2-2
2.3 Loop Controller Board Installation	2-4
<b>3. Software Installation</b>	<b>3-1</b>
3.1 Memory Driver Installation	3-1
3.2 Group3 Control LabVIEW Driver Installation	3-1
3.3 LC Find Utility	3-2
<b>4. The Driver VIs</b>	<b>4-1</b>
<b>5. Using the Group3 Control Drivers</b>	<b>5-1</b>
<b>6. The Sample Project</b>	<b>6-1</b>
<b>7. The Configuration String</b>	<b>7-1</b>
<b>8. Setup Driver VI Descriptions</b>	<b>8-1</b>
<b>9. Miscellaneous Driver VI Descriptions</b>	<b>9-1</b>
<b>10. I/O Board Driver VI Descriptions</b>	<b>10-1</b>
<b>11. I/O Board Timeout Setup VI Descriptions</b>	<b>11-1</b>
<b>12. Upgrading Drivers in Existing Applications</b>	<b>12-1</b>



# 1 Overview and upgrade notes

The Group3 Control LabVIEW Driver consists of a set of Virtual Instruments (VIs) written in the LabVIEW graphical programming language. These VIs are intended to be used in your LabVIEW application programs to provide an interface to the Group3 Control hardware.

This manual assumes that the reader is familiar with LabVIEW, and with the Group3 Control product range. Refer to the documentation supplied with the LabVIEW package, and to the “Group3 Control User’s Manual”.

The computer communicates with the Group3 Control system hardware through one or more Group3 Loop Controller boards which are inserted in the computer’s expansion slots. There are Loop Controllers available for ISA and PCI slots in the PC platform. The product range includes controllers supporting both one and 3 communication ports for each of the slot types.

The driver is supplied on a CD-ROM and supports all LabVIEW versions from 6.0 onwards. Installation instructions for the software are given in section 3 of this manual.

## Upgrade notes

### V4.9 to V5.0

Three CINs were moved to a DLL. They were used in; Application Setup.vi, Memory Peek/Poke Byte Array.vi and Send Data.vi. This means that the Group3 driver released for LabVIEW version 6 can be used for all versions of LabVIEW: from version 6 on. In Send Data.vi, Memsetf (was a CIN, now a function in the DLL) no longer writes data to dualport if it is the same as what is already there. This was necessary as the latest computers were sufficiently fast that LabVIEW was overloading the loop controller with write operations.

### V4.8 to V4.9

Added 32-bit count capability and count zeroing to H board VI, also required changes to Application Setup.vi.

Loop controller requires software V5.1 for H board 32-bit mode.

Error Handler.vi no longer puts out error message for error code 64.

### V4.7 to V4.8

Application Setup.vi was modified to prevent accessing memory outside the mapped area. This upgrade does not require changes to application VIs.

### V4.6 to V4.7

New Mem\_32.lsb for CIN in Application Setup.vi. PCI memory driver now accessed via PCI\_lib.dll. This upgrade does not require changes to application VIs.

### V4.5 to V4.6

New Mem\_32.lsb for CIN in Application Setup.vi to accommodate version 6 of memory driver. This upgrade does not require changes to application VIs.

## **V4.4 to V4.5**

Users who have applications made with version V4.4 of the Group3 Control LabVIEW Driver set will have to make a small wiring change to their applications once they have upgraded to more recent versions. Other small differences between the versions are also described below.

To upgrade, simply copy the file G3C.llb from the appropriate LabVIEW version folder on the distribution CD to overwrite the file of the same name on your hard drive.

When you next run the application, you will find it is not executable. To remedy this, locate the VI Application setup.vi in the application's wiring diagram. The connector pane of this VI has been modified, and the wiring to the VI must be reestablished. Once this is done, the VI will again be executable, and should be saved to disk.

Four VIs in the driver set were modified in version V4.5, as follows:

### **Application setup.vi**

This VI now has an extra numerical input to allow the application to be set to simulation mode, where the application will run for test or demonstration purposes even if one or more loop controllers required for normal operation are absent from the computer. The modes allowed are:

Mode 0 - simulation not allowed; LC(s) must be present for the application to run

Mode 1 - simulation allowed and will occur only if an LC is absent

Mode 2 - simulation is forced for all LCs in the application whether or not they are present.

### **F I/O Board Serial Comms.vi**

This VI has been modified so that, when data is being sent out from the application over the serial link, if the previous message is still being handled when a new message is given to the F Board VI, then a fatal error message will not occur. Instead the new message is just ignored. The VI generates a new boolean output which when true indicates that the last message was not sent. This output remains true until a message is successfully sent. This VI does not require relinking into existing applications made using V4.4, but the user may wish modify the application in order to make use of the new output.

### **K I/O Board GPIB IEEE488.vi**

#### **Proc Board Diagnostic Function.vi**

These VIs have been modified in a similar way to the F Board VI with regard to sending data out. Similarly they do not required relinking into an existing application, and they provide the same indication that data was not sent.

---

## 2 Loop Controllers - Hardware Description

### 2.1 PCI bus Loop Controllers

The LC1-PCI and LC3-PCI are Loop Controller boards designed to operate Group3 Control modules from a PCI bus slot. The LC1 drives a single fibre-optic communication loop, while the LC3 allows three completely independent loops to be controlled from the one slot. Each loop is capable of handling 16 I/O modules. A limit of 16 loops can be controlled from any one computer - for example 5 LC3's and an LC1 give the maximum of 16 loops. LC1 and LC3 boards may be mixed in a system.

The PCI Loop Controllers are Plug and Play devices - the computer BIOS allocates the actual bus addresses on start up, and the user does not need to be involved in the process.

The LCs require a system driver to be installed on the computer. This driver software is supplied on the "Group3 Loop Controller Driver Installation CD". Instructions for installing the driver are to be found in section 3 of this manual.

The only hardware setting the user needs to be aware of is the rotary switch on the top edge of the circuit board. This is shipped from the factory set at position '0', and this setting needs to be altered only if two or more PCI Loop Controller boards are to be installed in the same computer. The switch is there to allow the user and the application software to distinguish between the different Loop Controllers in the same computer.

The switch on an LC1-PCI sets the loop number for that board.

The switch on the LC3-PCI sets the loop number (call it  $n$ ) of the top transmitter/receiver pair - the one furthest from the gold board edge fingers. The number of the middle loop is one more than the switch setting ( $n+1$ ), and the number of the lower loop is two more than the switch setting ( $n+2$ ). The allowed range for any loop number is 0 to F (hex). If the switch is set to E, the loop numbers will be E, F, and 0. If the switch is set to F, the loop numbers will be F, 0, 1.

If several Loop Controller boards are used in one computer, remember that the switch settings must be different, and that the three loops of an LC3-PCI occupy three consecutive numbers, starting with the number that the switch is set to.

For detailed specifications of the Loop Controllers, please refer to the "Group3 Control User's Manual".

## 2.2 ISA bus Loop Controllers

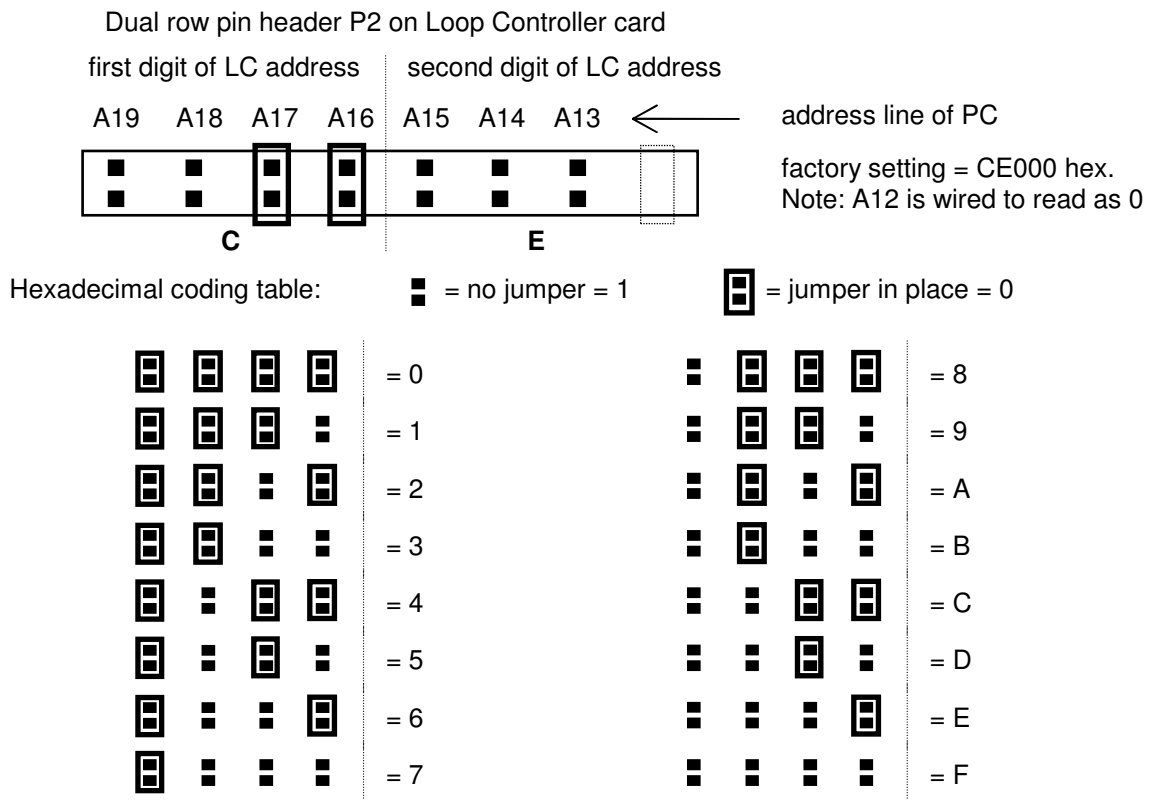
The LC1-PC and LC3-PC are Loop Controller boards designed to operate Group3 Control modules from an ISA bus slot. The LC1 drives a single fibre-optic communication loop, while the LC3 allows three completely independent loops to be controlled from the one slot. Each loop is capable of handling 16 I/O modules. The only limit to the number of boards used in a computer is the number of available ISA slots. LC1 and LC3 boards may be mixed in a system. ISA and PCI LCs may also be mixed in a system.

Communication to and from the PC processor takes place through shared memory installed on the LC boards. This shared memory is addressed directly by the control computer's address bus. Jumpers are provided on the LCs to place the shared memory at a suitable position within the computer's address range.

To operate in a 32-bit environment the boards require a system driver to be installed on the computer. This driver software is supplied on the "Group3 Control Driver Installation CD". Instructions for installing the driver are to be found in section 3 of this manual.

The shared memory of an ISA slot Loop Controller is intended to be placed in the area between 640k and 1M bytes in the computer's address space, where free address space usually is available. Conflict with other devices in the system must be avoided. The LC base address is determined by comparing PC address lines A13 to A19 with the jumper settings. 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).

The absence of a jumper on a particular address line corresponds to a 1. The presence of a jumper corresponds to a 0.





The address jumpers on an LC1-PC set the base address for that board.

The address jumpers on the LC3-PC set the base address of the top transmitter / receiver pair - the one furthest from the gold board edge fingers. The base address of the middle loop is 2048 bytes (800 hex) more than the address jumpers, and the base address of the lower loop is 4096 bytes (1000 hex) more than the address jumpers

With the jumpers set for CE, as the factory setting above, the base addresses of the three loops are CE00:0, CE80:0 and CF00:0 hex.

Note:

Care must be taken if mixing the current LC's with the earliest versions (pre 1993) which have 1Kbyte dual port RAMs (boards labelled 16000032A for LC1s, 16000048 for LC3s). These boards had an additional eighth jumper to set A12. If mixing old and new versions check that the address jumpers are set so that the memory areas used by the boards are different - it is not sufficient that the jumpers just appear to be set to physically different pins.

For detailed specifications of the Loop Controllers, please refer to the "Group3 Control User's Manual".

## 2.3 Loop Controller Board Installation

Note: Install the supplied memory drivers before installing the loop controller boards, see section 3.

A grounded static discharge strap must be worn when handling loop controllers outside of their anti-static bags, and care should be taken to avoid touching the gold edge connector contacts.

The computer must be powered off while Loop Controller boards are installed or removed.

### Loop Controller Board Identification:

The Group3 PCI and ISA loop controller boards are easily distinguishable:

PCI loop controller board (LC1-PCI and LC3-PCI):

- Each board has a rotary switch on the top edge of the circuit board.
- Each board has six circular holes on the rear metal bracket.
- The gold contacts have fine pitch spacing.

ISA loop controller board (LC1-PC and LC3-PC, aka LC1-ISA and LC3-ISA):

- Each board has a set of jumpers on the circuit board.
- Each board has one or three elliptical holes on the rear metal bracket.
- The gold contacts have coarse pitch spacing.

The Loop Controller board may be inserted in any vacant slot on the computer's mother board that is appropriate to the type of board being installed, PCI or ISA.

PCI slots have finer contact pitch spacing than ISA slots.

When planning your Loop Controller installation bear in mind that any mixture of PCI and ISA Loop Controllers, including LC1 and LC3, can be used and accessed from the LabVIEW application.

Also be aware that LC3 boards are long, so choose slots for them where they will not mechanically interfere with other components in the computer, for example the fan on the computer's processor chip.

ISA boards must have their jumpers set as described above before being fitted to the computer.

The switch settings of PCI boards can be adjusted after the board is installed.

Carefully insert the board into its slot, and secure it in place with the usual fixing screw.

---

## 3 Software Installation

Two CDs are provided for this.

### 3.1 Memory Driver Installation

The “Group3 Loop Controller Driver Installation CD” contains the software drivers required to operate Group3 Loop Controllers. You should already have this CD if you bought a Group3 Loop Controller.

- 1) Insert the CD into the computer’s CD drive. It should run automatically, but If it doesn’t, click: **Start•Run** on the computer and select “SETUP.EXE” on the CD. Follow the on-screen instructions.
- 2) Turn off the computer and install the loop controller card(s).
- 3) Run the LC\_Find utility to confirm that the LC is working properly. See section 3.3 below

### 3.2 Group3 Control LabVIEW Driver Installation

The “Group3 Control LabVIEW Driver CD” contains the Group3 LabVIEW driver plus example applications and manuals. Note: this software is for use with LabVIEW version 6 or later.

- 1) Insert the CD-ROM into the computer’s CD drive.  
There are four folders on the CD-ROM:
  - G3 Demo**
  - G3 Samples**
  - Group3 Control**
  - Group3 Manuals**
- 2) Copy the folder “Group3 Control” and paste it into the **user.lib** directory of the LabVIEW installation directory on your computer.
- 3) Copy the remaining folders anywhere you want.  
The “G3 Demo” and “G3 Samples” folders contain the examples VIs.  
The “Group3 Manuals” folder contains the manuals needed to use this driver.
- 4) Restart the LabVIEW application. LabVIEW will automatically load the new palette: “Group3 Control” for this library, in the Function palette/User Library.

#### Notes:

- The first time you open one or more of the supplied VIs, LabVIEW might ask you to locate the path of the file: G3\_Control\_LV.DLL. This is located in the folder:  
<LabVIEW installation directory>\user.lib\Group3 Control.
- Files copied from the CD-ROM might appear on the hard drive as “read-only”, i.e. with their read-only attribute set. This will prevent LabVIEW from updating these files when necessary. To avoid this problem, open Explorer (press **Windows E**), highlight the files concerned (**Ctrl A** will select all the files in a list), press **Shift F10** to open the context menu, and press **R** to open the properties window. Unselect the Read-only attribute by pressing **R** again or by clicking the check box until it is clear. Press **Enter** or click **OK** and then exit Explorer.

### 3.3 LC Find Utility

The program **LC\_Find.exe** in the \Memory Drivers folder on the “Group3 Control Driver Installation CD” will show a listing of all the Loop Controllers installed in the computer. To run the program, click **Start•Run**, and then navigate to the folder where you have copied this file. Double click **LC\_Find.exe**.

The program first gives a listing of all PCI LCs present, then after a keystroke gives a listing of all ISA LCs in the address range C0000 to FE000.

---

## 4 The Driver VIs

The G3C.IIb driver library contains the following VIs:

Group3 Control LabVIEW Driver Version History.vi

A I/O Board Fast Analog/Digital.vi  
A I/O Board Timeout Setup.vi  
Application Setup.vi  
B I/O Board 24-Digital, fast.vi  
B I/O Board 24-Digital.vi  
B I/O Board Timeout Setup.vi  
C I/O Board 8-Analog Input.vi  
CNA Analog/Digital.vi  
CNA Output Timeout Setup.vi  
Convert Offsets to Data 3-D Array to 2-D.vi  
D I/O Board 8-Analog Output.vi  
D I/O Board Timeout Setup.vi  
Display diagnostic response.vi  
DTM Error Handler.vi  
E I/O Board 4-Motor Driver.vi  
E I/O Board Timeout Setup.vi  
Error Handler.vi  
F I/O Board DTM Trigger.vi  
F I/O Board Serial Comms.vi  
G I/O Board 4-Stepper Motor Driver.vi  
G I/O Board Timeout Setup.vi  
Get Offset & Address from Arrays.vi  
H I/O Board 4-Encoder Input.vi  
I/O Board #0 Data Offset.vi  
I/O Board Setup.vi  
J I/O Board Precision 2-Analog Output.vi  
J I/O Board Timeout Setup.vi  
K I/O Board GPIB IEEE-488.vi  
LC Comms Control.vi  
LC Comms Shutdown.vi  
Loop Controller Misc Functions.vi  
Loop Controller Misc Info.vi  
Loop Controller Time Out.vi  
Memory Peek/Poke Byte Array.vi  
Misc Diagnostic Port Functions.vi  
Proc Board Diagnostic Port Function.vi  
Receive Data.vi  
Response Global.vi  
Scale Integer <-> Volt Array.vi  
Scale Integer <-> Voltage.vi  
Send Data.vi

---



## 5 Using the Group3 Control Drivers

There are many ways in which a LabVIEW program can be written using the Group3 Control driver VIs, but a good one to start with, sets up the program in a sequence structure. The first sequence frame configures the LC dual port RAM according to the Device Interfaces (DIs) which are present on the communication loop, and then sets communications running. This sequence frame runs only once. There may be one or more subsequent frames containing other system initialisation functions, but eventually execution reaches a frame in the sequence which contains the bulk of the LabVIEW application, and it runs continuously using a While Loop. A final frame in the sequence structure implements a graceful shutdown of the application when required.

Spend some time looking through the programs supplied in the “G3 Samples” and “G3 Demo” folders. They give some examples of ways you can use the driver VIs.

In particular, you may find it convenient to use the program Project.vi as the basis for starting your LabVIEW applications. This VI is discussed in detail in the next section.

Another example VI is Brief.vi in the “G3 Demo” folder. This VI runs a small Group3 Control demonstration system, and shows a typical use of the Proc Board Diagnostic Port Function.vi.

### LabVIEW Programming with the K Board

Commands can be sent to the K Board and ASCII strings can be sent and received on the bus using K I/O Board GPIB IEEE-488.vi from G3C.llb. See page 10-6 for details of the command set recognised by the K Board.

---

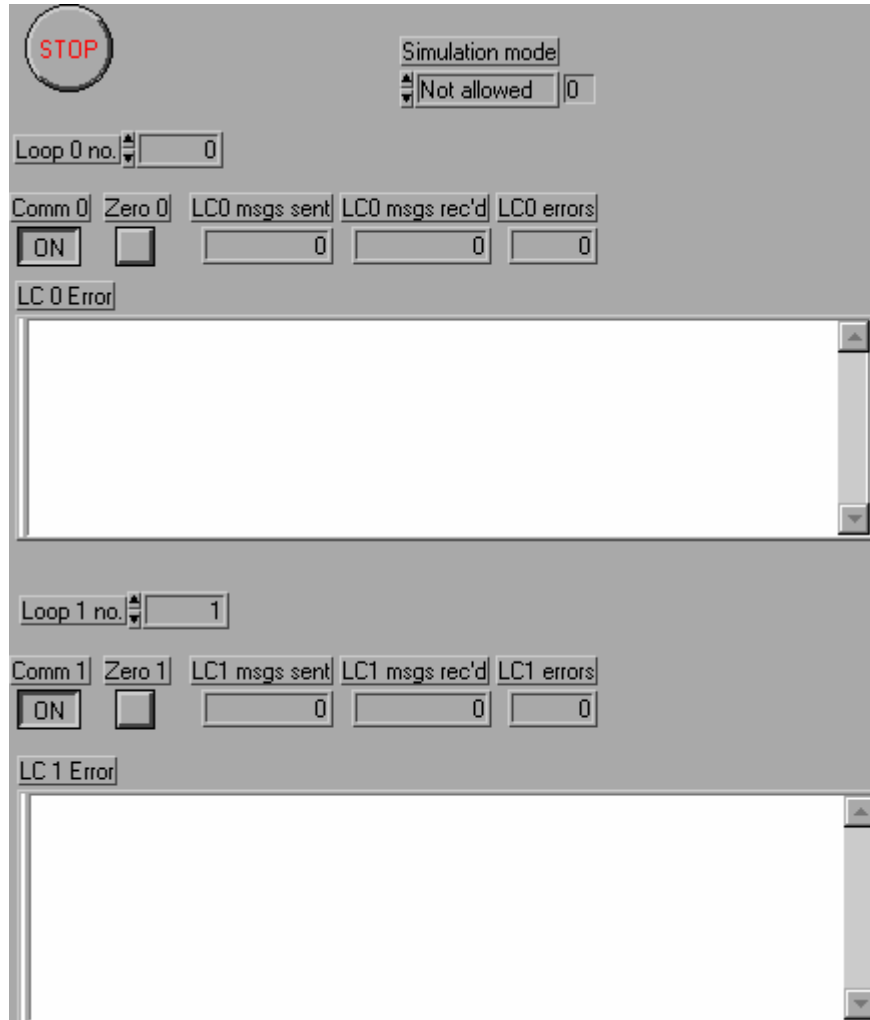




## 6 The Sample Project

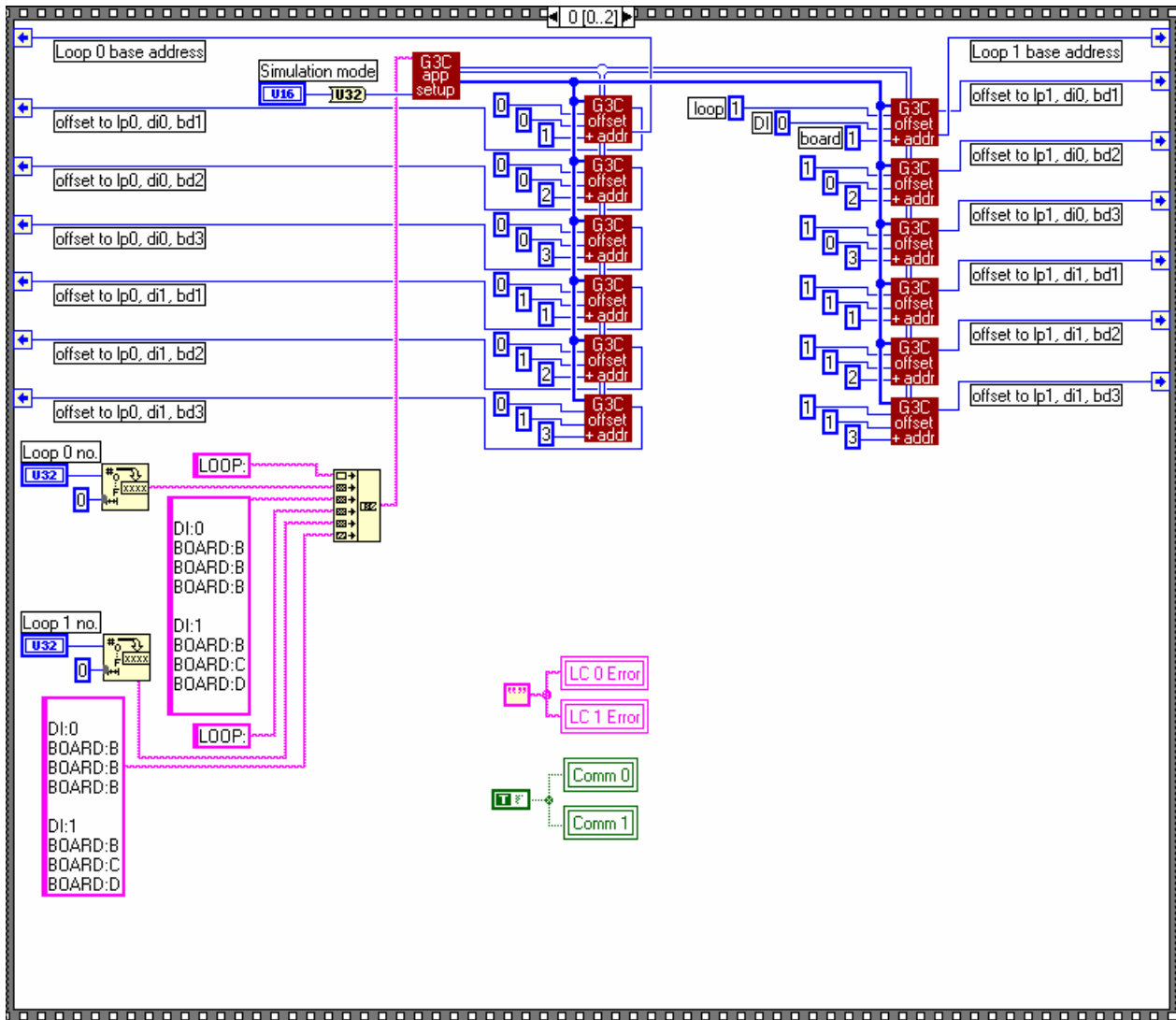
In the “G3 Samples” folder you will find a simple LabVIEW program, Project.vi, that illustrates points made in the previous section.

Project.vi embodies a Group3 Control system with two communication loops. In each loop there are two DIs, a CN3-BBB and a CN3-BCD. In the VI, as a simple example of a Group3 Control application, the read-backs from a board in one loop are wired to the controls of a board in the other loop to provide isolated bi-directional transmission of analog and digital signals between remote points.



Project.vi Front Panel

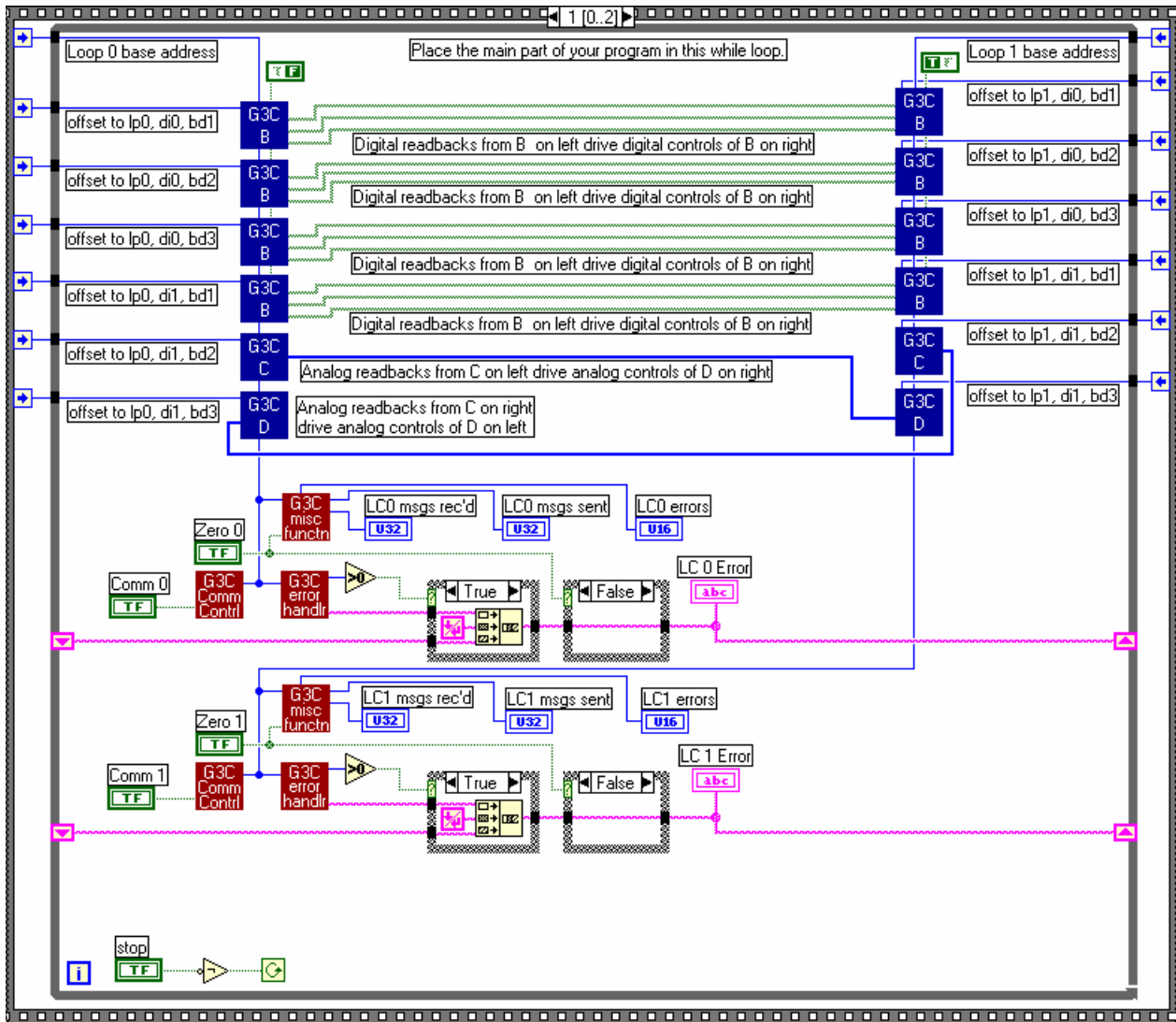
The wiring diagram of Project.vi follows on the next two pages. The VI front panel has some controls and indicators concerned with the setting up and running of the communication loops, and a stop button to allow graceful execution termination. The diagram contains a sequence structure with three frames. Frame 0 sets up the LC dual port RAM, starts communications, and generates the base addresses for each loop



Project.vi Diagram frame 0

and memory offsets for each I/O board in the system. Briefly, a text string, the setup string for this system is assembled from several sub-strings, including loop numbers from front panel controls. The setup string is decoded by **Application Setup.vi**, which generates two arrays, a 3-dimensional array containing the I/O board memory offsets by DI and loop, and a 1-dimensional array of loop base addresses. For each I/O board in the system there is a **Get Offsets & Addresses from Arrays.vi**. This vi indexes the two arrays to obtain the base address of each loop and the offsets for each board.

The base addresses and offsets are fed by sequence locals to frame 1 of the sequence, which contains the continuously operational part of the program – all the diagram is contained within a While loop, so it runs continuously until execution is halted by the operator.



Project.vi Diagram frame 1

The program inside the While loop includes sub-VIs of the various I/O boards in the system. The sub-VIs are wired up to perform the required functions, i.e. to transmit the various analog and digital values from one DI to the other.

Other sub-VIs in the diagram are arranged to report on the front panel any errors arising in the Group3 Control system, and to enable the LC timeout function. The LC timeout is set to 1.5 seconds (15 times 0.1s).

For details, refer to the "Group3 Control User's Manual".



Project.vi Diagram frame 2

## 7 The Configuration String

The configuration string specifies the Group3 Control system to be run by the LabVIEW application. In essence, it tells LabVIEW what to expect in terms of Loop Controllers in the system and their switch or jumper settings, plus the Device Interfaces associated with each Loop Controller and the I/O boards they contain. The string can be placed as a string constant within the application VI, as in the sample Project.vi described in section 6, but for larger systems it is more convenient to store the string in a disk file, called the configuration file. The file can have any name and extension you choose, though it conventionally has extension .CNF. The file is most conveniently stored in the folder containing the application VI. It can be generated using any text editor, such as Notepad or WordPad.

The configuration string is of the following form:

```
LOOP:[number of PCI LC, or address of ISA LC]
* COMM:[communication mode]
* DIAGNOSTIC_ON_DIS:[addresses of DIs to run diagnostics]
DI:0
BOARD:[board type by letter A – K or CNA] (one such line for each board in the DI)
```

### Notes:

- The configuration string may be in upper or lower case, or a mixture. Blank lines may be inserted.
- The number of an LC1-PCI is its switch setting. For an LC3-PCI the number is determined as described in section 2.1
- The address of an LC1-PC (ISA type) is its jumper setting (e.g. CE) followed by 3 zeros, (e.g. CE000). For an LC3-PC the addresses are determined as described in section 2.2. The address must always be padded out to 5 hexadecimal characters.
- The DI:n line and its following BOARD:X lines are repeated for each DI belonging to the LC specified immediately above.
- Board categories, such as the 2 in C2, are not included in the configuration string.
- The above format is repeated for each LC in the system.
- \* Lines marked with an asterisk are optional.
- COMM:FAST invokes the multiple boards per message feature. This approximately doubles the comm's performance between the LC and the DIs.
- DIAGNOSTIC\_ON\_DIS:0,1,2,etc. readies the specified DIs for diagnostics-over-the-loop feature.

An example of a complete configuration string is shown below

```
LOOP:CE000  
COMM:FAST  
DIAGNOSTIC_ON_DIS:0,1  
  
DI:0  
BOARD:B  
BOARD:C  
BOARD:D  
  
DI:1  
BOARD:F:P0:GP:P1:DTM:5  
BOARD:E  
  
LOOP:CE800  
  
DI:0  
BOARD:A  
BOARD:D  
  
LOOP:CF000  
DIAGNOSTIC_ON_DIS:0  
  
DI:0  
BOARD:G  
BOARD:J  
BOARD:H:L  
  
DI:1  
BOARD:CNA  
  
LOOP:0  
  
DI:0  
BOARD:K
```

The configuration string above specifies four communication loops, three from the ports of a 3-loop LC with address jumpers set for a base address of CE000, and the fourth from an LC1-PCI with its switch set to 0. The string has four sections, one for each loop, and each starting with the LOOP: statement. Next comes the COMM: statement which invokes the fast communication mode for the first loop. For two of the loops the diagnostics-over-the-loop function is set up using the DIAGNOSTIC\_ON\_DIS: statement.

After the initial loop definition part of the string, comes the section defining the DIs and the boards they contain. Each DI is defined first by its address, then by its I/O boards. The DI addresses must always start at 0 and then increment by 1 for each DI, up to a maximum of 15 (16 DIs total) for each loop. The addresses defined in the configuration string must correspond to the decimal equivalent of the hexadecimal address switch settings on the DIs. There can be 1, 2, or 3 BOARD: statements, depending on the number of boards in the DI. The boards in any one DI must all have their address

jumpers set uniquely in the range 1 through 3. Address 0 is reserved for the processor board, relating to the diagnostic port function.

Boards A, B, C, D, E, G and J are specified simply with their single letter.

The H board can be specified as either:

BOARD:H - use four 16-bit encoders

BOARD:H:L - use four 32-bit encoders

For 32-bit mode:

- The loop controller software must be version 5.1 or higher.
- The Group3 LabVIEW driver must be version 4.9 or higher.

Note: the selection of 16/32-bit mode doesn't affect the DI or the physical H board.

The CNA Device Interface is specified as: BOARD:CNA.

Board type F, serial communications board, takes a series of descriptors to set the functions of the two serial ports on the board, as set out below:

BOARD:F:P[port number]:[port type]:P[port number]:[port type]

Port number can be 0 or 1.

Port type can be:

- **GP** to specify a general purpose serial port.
  - **DTM** to specify that the port will be used to communicate with one or more Group3 digital teslameters. DTM is followed by :n where n is the number of teslameters connected to this port (up to 8).
-

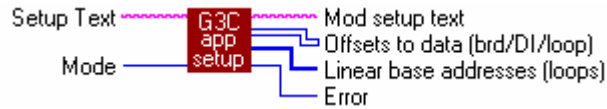




## 8 Setup Driver VI Descriptions

The VIs described in this section are used at the beginning of a LabVIEW program to allocate space in the LC dual port RAM, to configure the LC communication mode, and to set communications running.

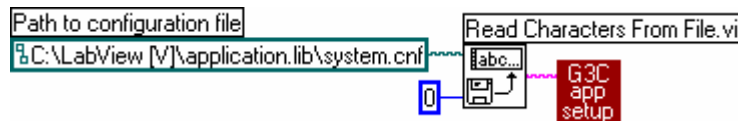
### 8.1



Application Setup.vi

**Application Setup.vi** inputs the configuration string or setup string. The string defines the entire Group3 Control system in use. From this information the VI calculates the memory address offsets from the base address of the LC dual port RAM for the data areas allocated to every DI board in the system. This information comes from the VI in the form of a 3-dimensional array which arranges the data by loop, DI, and board. A 1-dimensional array from the VI outputs the base addresses of each LC in the system. Finally, the VI sets up the communication mode of each LC, and sets communications going. Any errors encountered in deciphering the configuration string are announced in a dialog box. This VI disables the LC timeout function. The timeout function can be restarted when the main part of the program is running, using the Loop Controller Time Out.vi. See page 9-1.

The Setup Text input to Application Setup.vi can be generated by a string constant embedded in the LabVIEW diagram. Alternatively, the text can be stored in a disk file and read into the setup vi using Read Characters From File.vi, which is found in the VI library at C:\LabVIEW [V]\Vi.lib\Utility\file.lib. The function is implemented as shown below:



The MODE input to the VI allows the application to be set to simulation mode. In this mode the application can be run for test or demonstration purposes without Loop Controller(s) present in the computer. The modes allowed are:

- Mode 0 – simulation not allowed; LC(s) must be installed for the application to run
- Mode 1 – simulation allowed and will occur only if an LC is absent
- Mode 2 – simulation is forced for all LCs in the application whether or not they are present.

The mode input to the VI is numerical in the range 0 through 2, the numbers corresponding to the modes. The mode input can be conveniently provided from the application's front panel using a text ring. See Project.vi in section 6.

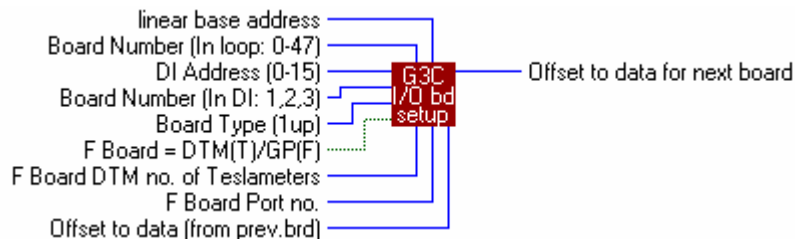
## 8.2



I/O Board #0 Data Offset.vi

**I/O Board #0 Data Offset.vi** calculates the address offset to the data area allocated to the first I/O board in the loop. This VI is called by Application Setup.vi.

## 8.3



I/O Board Setup.vi

**I/O Board Setup.vi** calculates the address offset to the data area allocated to the next board in the loop sequence. The VI allows for the space required in RAM by the current board, adds it to the current offset calculated previously, and outputs the result as the offset for the next board. This VI is called by the Application Setup.vi.

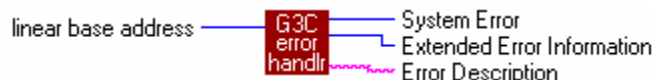
## 8.4



Get Offset & Address from Arrays.vi

**Get Offset & Address from Arrays.vi** indexes through the two arrays generated by the Application Setup.vi to output the base address of the LC and the offset of one I/O board as defined at the inputs to the VI. This VI must appear in the initialisation part of the LabVIEW application program once for every I/O board in the system. The offset and base address are used by the I/O board VI so that the latter will access the appropriate area in LC dual port RAM.

## 8.5



Error Handler.vi

**Error Handler.vi** is called by Application Setup.vi, and is used to decipher error codes during LC setting up. You may also wish to use this VI in the main body of your program to detect communication errors.

## 8.6



### Convert Offsets to Data 3-D Array to 2-D.vi

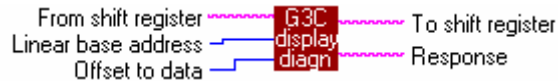
**Convert Offsets to Data 3-D Array to 2-D Array.vi** converts the 3-D array (board/DI/LC) generated by the Application Setup.vi to a 2-D array (board/LC) as used in earlier driver versions. This VI is included to allow upgrading of programs for use with this driver version.

---



## 9 Miscellaneous Driver VI Descriptions

### 9.1



Display Diagnostic Response .vi

**Display Diagnostic Response.vi** is called by the Proc Board Diagnostic Functions.vi to display diagnostic messages received from the Device Interface, using Response Global.vi.

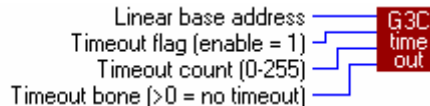
### 9.2



DTM Error Handler.vi

**DTM Error Handler.vi** converts error numbers generated by digital teslameters to ASCII error message strings.

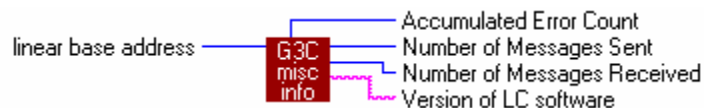
### 9.3



Loop Controller Time Out.vi

**Loop Controller Time Out.vi** is used to set the timeout count and to send a timeout bone. Setting the Time Out Flag to 1 enables the timeout function. The timeout count input is set in increments of 0.1 seconds with a range of 1 through 255 for a timeout period of 0.1 to 25.5 seconds. The timeout bone input must be loaded with a non-zero value within the timeout period to prevent a timeout. For example, if you run this VI once per second, the timeout count must be set to 11 or greater to prevent timeout occurring. If the program stops running, timeout will occur and the LC will set to zero any channels which have timeout enabled.

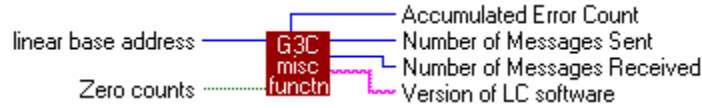
### 9.4



Loop Controller Misc Info.vi

**Loop Controller Misc Info.vi** is used to obtain message and error counts and LC software version from the LC.

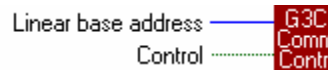
## 9.5



Loop Controller Misc Functions.vi

**Loop Controller Misc Functions.vi** performs the same functions as Loop Controller Misc Info.vi, except it has an input for zeroing the message and error counts.

## 9.6



LC Comms Control.vi

**LC Comms Control.vi** lets communications from the LC at the base address specified to be turned on and off from within the LabVIEW application. A TRUE at the Control input turns comms on, and a False turns it off.

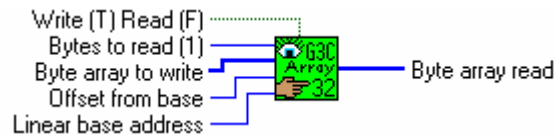
## 9.7



LC Comms Shutdown.vi

**LC Comms Shutdown.vi** stops communication from the LC at the specified base address.

## 9.8



Memory Peek/Poke Byte Array.vi

**Memory Peek/Poke Byte Array.vi** is the lowest level VI in the driver set. It reads and writes LC dual port RAM. This VI is called by most of the other VIs in the driver set. It calls a DLL to peek and poke an array of memory bytes. A constant limit of 2048 bytes from the base address has been set in the VI, and an error is returned if this limit is exceeded. The limit can be changed if necessary in the diagram of the VI. If you are using an older LC with 1k dual port RAM, the limit should be changed to 1024.

## 9.9



Receive Data.vi

**Receive Data.vi** reads data from the LC dual port RAM, making sure that multibyte reads are from only one Device Interface message. Offset to read data must be set to 5 for A and B boards, 2 for C and H boards, 12 for G, and  $10 + 16 \times (\text{no.DTMS})$  for F boards in DTM mode. Other boards do not use this VI.

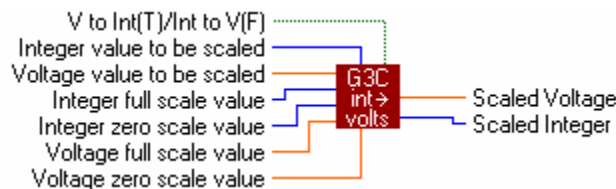
## 9.10



Response Global.vi

**Response Global.vi** is used by Display Diagnostic Response.vi.

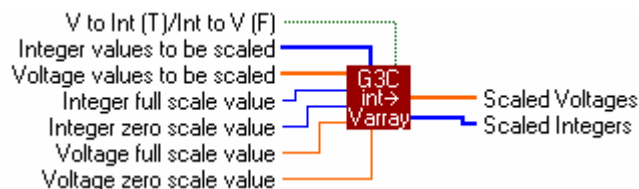
## 9.11



Scale Integer <-> Voltage.vi

**Scale Integer <-> Voltage.vi** is used to scale between integers and single precision voltage values. This VI is called by the Scale Integer <-> Volt Array.vi which uses it to scale an array of values.

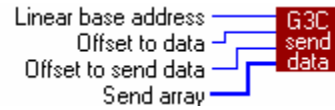
## 9.12



Scale Integer <-> Volt Array.vi

**Scale Integer <-> Volt Array.vi** is used to scale an array of values between integers and single precision voltage values. A True select will cause the VI to scale voltages to integers, and a False select will cause the VI to scale integers to voltages.

## 9.13



Send Data.vi

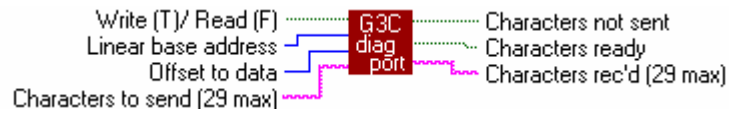
**Send Data.vi** writes data to LC dual port RAM, making sure that multibyte writes are not sent out on the loop until all bytes have been written to dual port RAM. Offset to send data is 4 for A boards, and 2 for all other output boards.

---



## 10 I/O Board Driver VI Descriptions

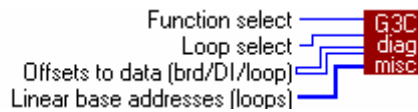
### 10.1



Proc Board Diagnostic Port Function.vi

**Proc Board Diagnostic Port Function.vi** gives access to the diagnostic port menu of a DI from within a LabVIEW program.

### 10.2



Misc Diagnostic Port Functions.vi

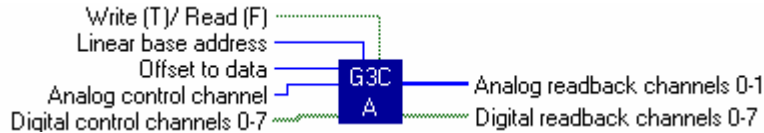
**Misc Diagnostic Port Functions.vi** is used to enable and disable the diagnostic port function and to reset all DIs that were configured for diagnostic port function in the configuration string used by Application Setup.vi.

The function select input can be set to 0, 1, or 2, for the following functions:

- 0: enable diagnostic port function
- 1: disable diagnostic port function
- 2: send reset string to DIs

The effect of sending the reset string to the DIs is to restart the DI operating software. This function is useful if you suspect that the DI software has locked up. Note that the selected function is performed on all the DIs which were configured for diagnostic port function in the loop selected by the Loop Select input.

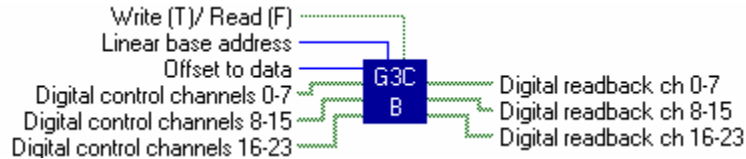
### 10.3



A I/O Board Fast Analog/Digital.vi

**A I/O Board Fast Analog/Digital.vi** is used to communicate with a Type A Fast Analog/Digital I/O Board. It handles all read and write functions of the board.

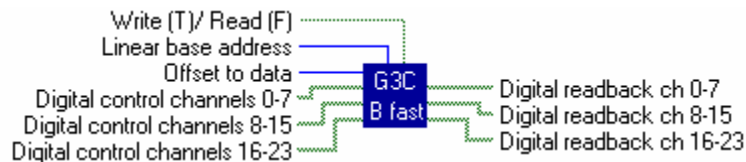
### 10.4



B I/O Board 24-Digital.vi

**B I/O Board 24-Digital.vi** is used to communicate with a Type B 24-Digital I/O Board. It handles all read and write functions of the board.

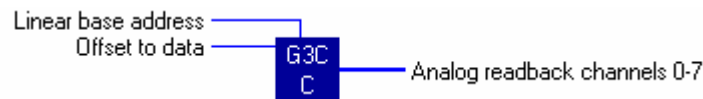
### 10.5



B I/O Board 24-Digital, fast.vi

**B I/O Board 24-Digital,fast.vi** is used to communicate with a Type B 24-Digital I/O Board. It handles all read and write functions of the board. This VI differs from the previous VI in that it is optimised for speed, but there is no provision for ensuring that all 24 bits of data are transferred to or from the board in a single message. In applications where data integrity is required, the VI in 10.4 above should be used.

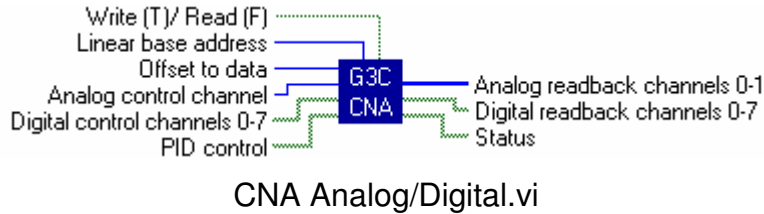
### 10.6



C I/O Board 8-Analog Input.vi

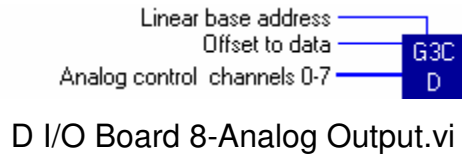
**C I/O Board 8-Analog Input.vi** is used to communicate with a Type C 8-Analog Input I/O Board. It handles the read function of the board.

## 10.7



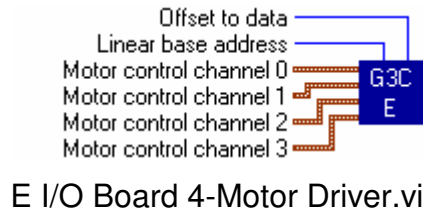
**CNA Analog/Digital.vi** is used to communicate with the CNA Device Interface. It handles all read and write functions of the CNA. For individual bit assignments of the PID Control byte and Status byte, see the front panel of this VI.

## 10.8



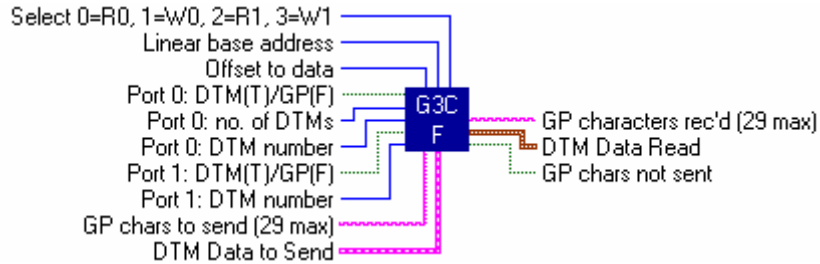
**D I/O Board 8-Analog Output.vi** is used to communicate with a Type D 8-Analog Output I/O Board. It handles the write function of the board.

## 10.9



**E I/O Board 4-Motor Driver.vi** is used to communicate with a Type E 4-Motor Driver I/O Board. It handles the write function of the board.

## 10.10



F I/O Board Serial Comms.vi

**F I/O Board Serial Comms.vi** is used to communicate with a Type F Serial Communications I/O Board. It handles all read and write functions of the board. The two serial ports on the F board can be set up individually to be either general purpose serial ports or specially configured to communicate with Group3 digital teslameters (DTMs) connected on their own serial communication loop.

The serial port parameters (number of data and stop bits, parity, and baud rate) are set up through the diagnostic port itself or through the diagnostic port VI.

The select input can be set to 0, 1, 2, or 3 for the following functions:

- 0: read from port 0
- 1: write to port 0
- 2: read from port 1
- 3: write to port 1

The DTM data to send should be assembled as a cluster of 6 elements of data:

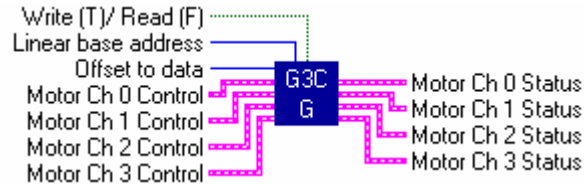
- Trigger: a 1 will cause all DTMs in trigger mode to take/store a field reading (U8).
- DTM address: the address of the DTM to receive the command in the next 4 elements (U8).
- Trigger mode: C for continuous measurement, V for triggered mode (ASCII).
- Range select: 0 for lowest, 3 for highest (U8).
- Units: G for gauss, T for tesla (ASCII).
- Zero: a 1 will set the current range to read zero (U8).

The DTM data to read is output as a cluster of 3 elements:

- Field value (U32)
- Probe temperature (U32)
- Error (U8).

The field and temperature are 4 byte representations of IEEE 82.3 format floating point numbers. The temperature is in degrees Celsius.

## 10.11



G I/O Board 4-Stepper Motor Driver.vi

**G I/O Board 4-Stepper Motor Driver.vi** is used to communicate with a Type G 4-Stepper Motor I/O Board. It handles all read and write functions of the board.

Motor control is a cluster containing the following elements:

Mode: bit 0: run mode: 0 = position control, 1 = continuous run  
bit 1: stop mode: 0 = rotor locked (energised), 1 = rotor free (windings off)  
bit 2: step increment: 0 = full step, 1 = half step

Control: bits 0 & 1: 00 = stop free (all windings off)  
01 = reverse (position count decrementing)  
10 = forward (position count incrementing)  
11 = stop locked (motor energised)

Bits 0 and 1 are relevant only in continuous run mode.

bit 2: 0 = actual position count accumulates  
1 = count set to zero.

Step count: the desired step position to which the motor is to run (I32)

Step rate: the maximum step rate or speed of the motor (U16)

Acceleration: the rate at which the motor speeds up and slows down (U8)

Note that many setup functions are available through the diagnostic port.

Motor status is a cluster containing the following elements:

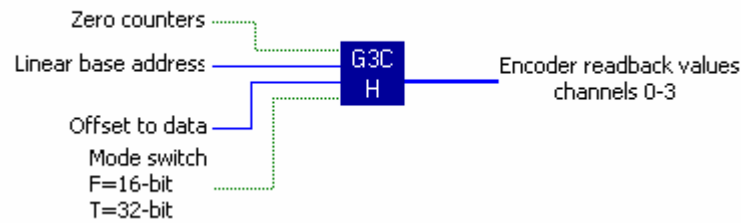
Actual step count: the current motor position (I32)

Analog input: the value of the analog input (U8)

Lower limit: True if the lower limit digital input is active (T/F)

Upper limit: True if the upper limit digital input is active (T/F)

## 10.12



H I/O Board 4-Encoder Input.vi

**H I/O Board 4-Encoder Input.vi** is used to communicate with a Type H 4-Encoder Input I/O Board. It handles the read function of the board.

The H board can now be configured for either 16-bit or 32-bit encoders (previously only 16-bit). The only conditions on using 32-bit mode are:

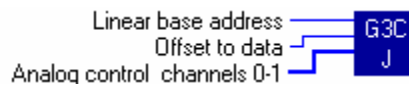
- The loop controller software must be version 5.1 or higher.
- The Group3 LabVIEW driver must be version 4.9 or higher.

If the Group3 LabVIEW driver is updated and the 16/32-bit H board VI is installed into an old application, then by default, the H board VI will operate in 16-bit mode, unless changes are made. The extra “mode” terminal will default to False (16-bit).

To use the new 32-bit encoder feature, the following must be done, in addition to the normal/previous connections and setup:

1. Set the “Mode” terminal to TRUE.
2. In the config’ script, refer to the H board as “BOARD:H:L” (“:L” to use 32-bit).

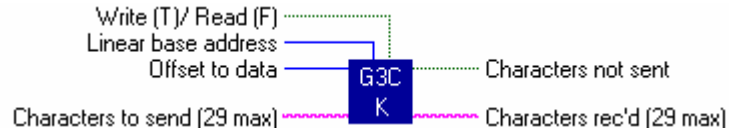
## 10.13



J I/O Board Precision 2-Analog Output.vi

**J I/O Board Precision 2-Analog Output.vi** is used to communicate with a Type J Precision 2-Analog Output I/O Board. It handles the write function of the board.

## 10.14



K I/O Board GPIB IEEE-488.vi

**K I/O Board GPIB IEEE-488.vi** is used to communicate with a Type K GPIB Controller Board. The Type K board recognises the commands specified below. Using these commands, the board can handle up to eight devices connected to the bus.

Commands and data are passed to the K board as strings input to the K Board VI. Then the system passes the strings up to the DI containing 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, a command has to be written out to that device (a command that the instrument understands), asking for the value of the measurement, then perform a read operation to the instrument to bring back the data, specifying the number of bytes required. These commands can be concatenated and sent as a single string.

Responses coming back from the K Board can be read out from the K Board VI. Responses consist of acknowledgement characters from the K Board, together with any response from the device(s) on the bus.

GPIB Commands for the K board follow on the next pages.

## GPIB Commands for the K board

Throughout this list the following conventions are observed:-

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

Descriptions in < > 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.

\n = line feed, new line, <LF>, hex 0A.

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

Upper or lower case alpha characters may be used.

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

### **RD <number of bytes to read> [,<device address>] \r**

READ command, to address and/or read data from a device on the GPIB.

If <number of bytes to read> is 0, then this command prepares the device for future reads; it leaves the device addressed for reads.

Example: RD0,DEV1 \r addresses device one for future reads.

If no <device address> is given, this command reads from the device previously addressed for a read.

Example: RD15\r reads 15 bytes from the previously addressed device

Response: \r <DATA> \n e.g. \r -32.5C \n

### **WRT [<device address1>[,<device address2>]] \r <DATA> \r**

WRITE command, sends <DATA> to the addressed devices.

Example: WRT DEV1 \r T \r will write the character 'T' to device 1

A number of device addresses can be given, <DATA> will be written to all. If no addresses are given, will write to previously addressed devices.

Response: \r <count of bytes received (in hex)> \n

The example above produces \r 1 \n, or \015 1 \012 if using octal notation

### **EOS <eos character [+ offset]> [,<device address (dev0..dev7)>] \r**

Changes or Disables the EOS (End of String) response, [ending on reads]

The <eos character> should be in hex representation. eg 0d for <CR>

add 0x400 to it for "terminate read on EOS" action

add 0x800 to it for "set EOI on EOS" action

add 0x1000 to it to select 8-bit compare of EOS

e.g. EOS 0a \r changes the EOS character to Line Feed

Response: \r <old eos in hex> \n

Example EOS 0a \r produces \r D \n if old EOS was a <CR> (0D in hex).



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

Enables or disables the issuing of an EOI on a Write.

The EOI can be automatically added on the end of a Write.

Response: \r <old EOI setting> \n

Example: EOT 1 \r enables EOI, returns \r 0 \n if originally disabled.

**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-line or off-line. eg. ONL 0 \r takes the controller off-line.

Resets all if placed on-line.

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 IEEE-488 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:-

EOI,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 board's 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

Sets 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 unassert 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

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. e.g. 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 in use.

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).

---

## 11 I/O Board Timeout Setup VI Descriptions

Timeout setup VIs are provided for the output I/O boards, types A, B, D, E, G, and J, and for the CNA Device Interface.

The timeout feature of the LC is normally enabled, but can be disabled globally using Loop Controller Timeout.vi. See section 9.3.

When the timeout feature is enabled, the I/O board timeout setup VIs allow timeout to be disabled on individual channels.

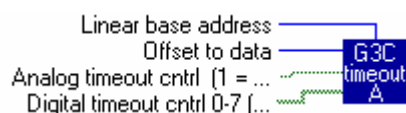
If timeout is active and enabled for any channel, and if the LabVIEW application stops accessing the Loop Controller, then after the timeout period as set by the Loop Controller Timeout.vi the LC will set the channel to zero for analogs or off for digitals.

If timeout is disabled by Loop Controller Timeout.vi, or if a channel's timeout is disabled by a timeout setup VI, then the channel output will be maintained by the LC even if the computer stops accessing the LC.

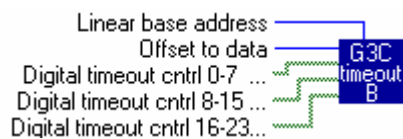
(Note that the LC timeout is quite separate and distinct from the DI timeout feature. The latter, when activated in the DI via the diagnostic port, shuts down output channels when communication between the LC and the DI is interrupted, as would occur if a fibre-optic cable became unplugged, or if LC communications stopped either intentionally or because of a computer fault. LC timeout occurs when the computer stops accessing the LC, but the LC is still active and communicating with the DIs.)

To disable timeout of a channel, include the timeout setup VI relevant to the I/O board concerned in your diagram, wire the appropriate loop base address and offset-to-data to it, and place a TRUE on the control input corresponding to the desired channel. Channels which are to shut down on timeout should have a FALSE placed on their timeout control inputs.

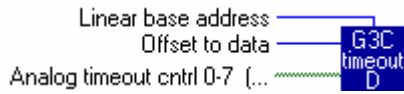
The timeout setup VI icons are shown below.



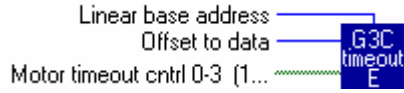
A I/O Board Timeout Setup.vi



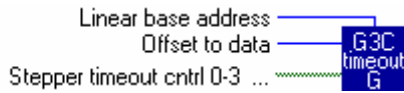
B I/O Board Timeout Setup.vi



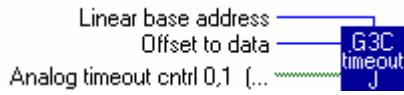
D I/O Board Timeout Setup.vi



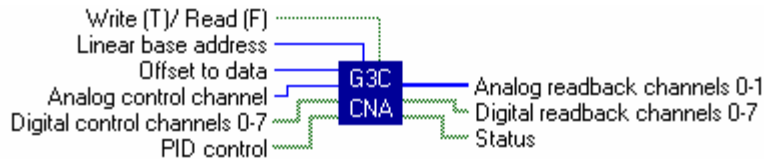
E I/O Board Timeout Setup.vi



G I/O Board Timeout Setup.vi



J I/O Board Timeout Setup.vi



CNA Output Timeout Setup.vi

## 12 Upgrading Drivers in Existing Applications




If you are upgrading from version 4.4 of the Group3 Control LabVIEW Driver, see the upgrade notes in section 1 of this manual.

This version 5.0 of the Group3 Control LabVIEW Driver is similar in structure to earlier versions 4.X which were supplied as VI library Cnet.llb. The major changes are:

- Call Library Functions (accessing a Group3 DLL) are called from some VIs for faster access to LC shared memory, using the new memory drivers supplied with the LabVIEW driver. The new memory drivers allow the use of both ISA and PCI Loop Controllers in systems running Windows 98, ME, NT and 2000.
- Two of the VIs have been renamed –
  - Set Up ControlNet Application.vi *is renamed to* Application Setup.vi
  - ControlNet Error Handler.vi *is renamed to* Error Handler.viThese VIs will need to be reinserted into the application, as described below.
- Some of the VIs have had their connector panes changed, and will need to be rewired into the application, as described below.
- The following VIs have been omitted from this LabVIEW driver version as they are no longer required –
  - ControlNet System Setup 9th-bit.vi
  - ControlNet System Setup SDLC.vi
  - platglob.vi

The following notes are offered as a guide to replacing the old driver version with this new version in an existing application –

1. Install the memory driver software that is appropriate to your computer operating system and the type(s) of Loop Controller boards that you are using, as described in section 3.1.
2. Install the LabVIEW driver, as described in section 3.2.
3. Your old version of the LabVIEW driver will be in VI library Cnet.llb, located in folder Cnet.lib off the LabVIEW folder. Preferably make a backup of Cnet.llb on a removable medium, and then delete it from your hard drive. Alternatively, rename Cnet.llb as Cnet.sav, for example. However, be aware that LabVIEW is very clever at finding files it needs, when loading an application, and we definitely do not want it to find Cnet.llb, whatever its name is, from now on.
4. Run LabVIEW, and open your application. When the application tries to load the old VIs from Cnet.llb, it will not be able to find them, and will ask you to help. For most of the driver VIs, navigate to the VI library G3C.llb in the “Group3 Control” folder, open up G3C.llb and select the appropriate VI. It should be necessary to navigate to G3C.llb only once; LabVIEW will be able to find most of the remaining driver VIs in the new VI library. However, for Setup ControlNet Application.vi and ControlNet Error Handler.vi do **NOT** navigate but simply click the Cancel button to skip these VIs and allow the loading to continue.

5. When loading is complete, there will be a number of errors in the application VI, as indicated by the Run arrow being broken and the Error List appearing.
  6. Repeat this process for each of the errors –
    - Click on each error in turn, and click the Find button. LabVIEW will take you to the appropriate place in the block diagram. In most cases the VI will be “washed out”, i.e. in a pale color. In two cases, however, the VI will be replaced by a box containing a large question mark .
    - Right-click the pale VI or , and click Replace.
    - Click Choose a VI, and navigate to G3C.llb, located in the “Group3 Controls” folder.
    - Open up G3C.llb, and highlight the appropriate replacement VI in the list. In the case of , choose either Application Setup.vi or Error Handler.vi, according to the context in the application. Click OK.
    - When the replacement VI is in position in full color, right-click it and select Show Terminals. Make sure the wiring connects to the correct terminals on the VI.
    - Where you have embedded Group3 Control driver VIs in sub-VIs of the main diagram, these sub-VIs will have to be opened so that the new driver VIs can be installed.
    - After each VI has been replaced, click the Run button again to refresh the Error List.
    - Note that the new driver VI icons have G3C in place of CNet.
    - Miscellaneous errors such as Wire Type Conflict are easily resolved on a case-by-case basis.
  7. When all the errors have been corrected, the application should run, and it can now be saved.
-



# 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.pmpower.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 (Rosaling)  
email: lqrosaling@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@vicontrols.com website: www.vicontrols.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