USER MANUAL

ProDAQ Data Acquisition Function Cards

ProDAQ 3610 48-Channel, DIO Function Card

PUBLICATION NUMBER: 3610-XX-UM-0101

Bustec Production, Ltd.
Bustec House, Shannon Business Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

PROPRIETARY NOTICE

This document and the technical data herein disclosed, are proprietary to Bustec Production Ltd., and shall not, without express written permission of Bustec Production Ltd, be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than Bustec Production Ltd. The information herein has been developed at private expense, and may only be used for operation and maintenance reference purposes or for purposes of engineering evaluation and incorporation into technical specifications and other documents, which specify procurement of products from Bustec Production Ltd. This document is subject to change without further notification. Bustec Production Ltd. reserves the right to change both the hardware and software described herein.

Table of Contents

Table of figures

Reference Documents

Title	Number
ProDAQ 3120 Hardware Manual	3120-XX-HM
ProDAQ 3150 Hardware Manual	3150-XX-HM

Glossary

1. Introduction

The ProDAQ 3610 48-Ch. Digital-I/O function Card provides 48 channels of TTL compatible, programmable digital I/O. The channels are organized in groups of eight, but can also be accessed in groups of 16 or 32 channels at a time.

Each group can be individually programmed to function as input or output. Due to the double buffering scheme implemented on the card, the input and output can be performed asynchronously or strobed using one of the trigger signals.

Using the on-board FIFO memory, digital pattern can be generated or acquired. Each group can be individually programmed whether to take part in the generation or acquisition.

The ProDAQ 3610 function card is one of a range of function cards designed to provide full functionality when installed in one of the range of ProDAQ motherboard modules such as the ProDAQ 3120.

2. Installation

2.1. Unpacking and Inspection

The ProDAQ module is shipped in an antistatic package to prevent any damage from electrostatic discharge (ESD). Proper ESD handling procedures must always be used when packing, unpacking or installing any ProDAQ module, ProDAQ plug-in module or ProDAQ function card:

- Ground yourself via a grounding strap or similar, e.g. by holding to a grounded object.
- Discharge the package by touching it to a grounded object, e.g. a metal part of your VXIbus chassis, before removing the module from the package.
- Remove the ProDAQ module from its carton, preserving the factory packaging as much as possible.
- Inspect the ProDAQ module for any defect or damage. Immediately notify the carrier if any damage is apparent.

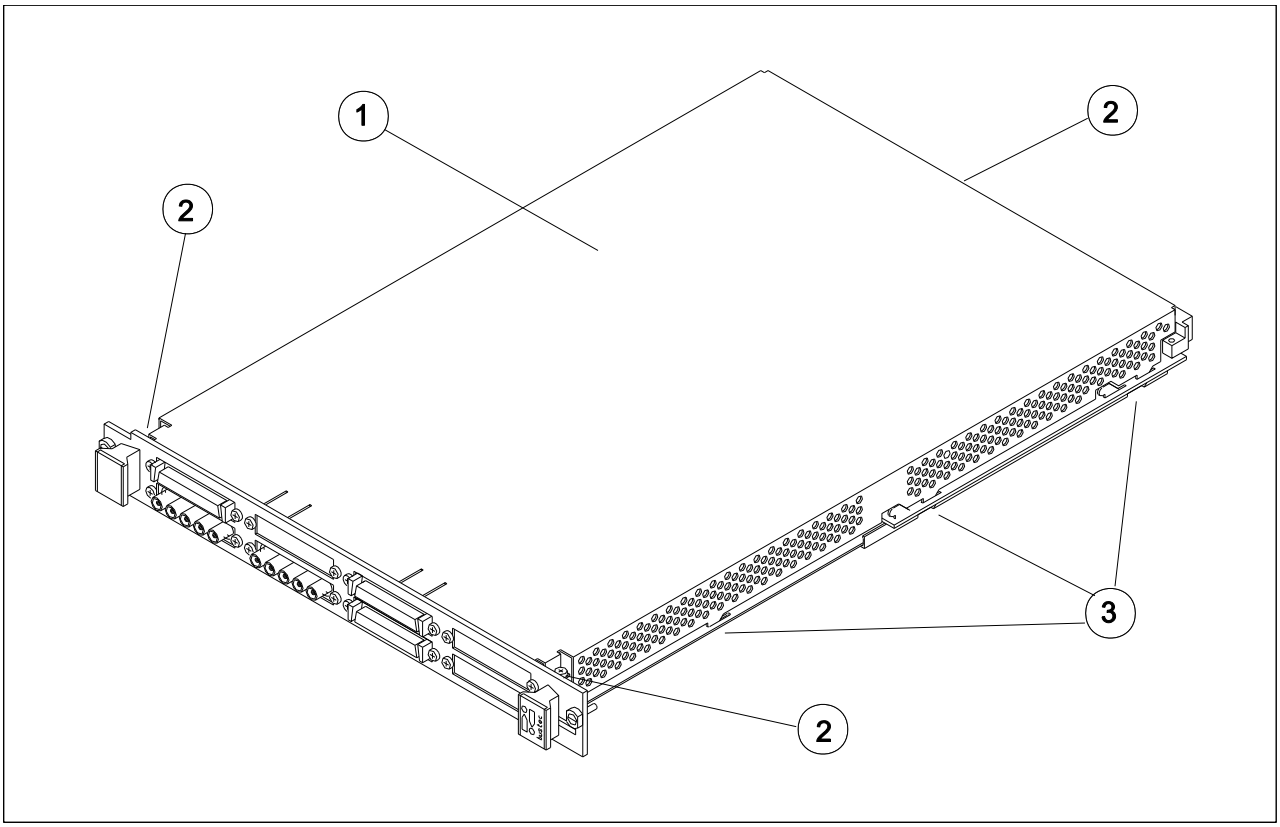
2.2. Reshipment Instructions

Use the original packing material when returning a ProDAQ module to Bustec Production Ltd. or calibration or servicing. The original shipping carton and the instrument's plastic foam will provide the necessary support for safe reshipment.

If the original anti-static packing material is unavailable, wrap the ProDAQ module in anti-static plastic sheeting and use plastic spray foam to surround and protect the instrument. Reship in either the original or a new shipping carton.

2.3. Preparing the ProDAQ Module

To install a ProDAQ Function Card into one of the ProDAQ Motherboards, you need to remove the modules top cover:



1 - Module Cover 2 - Cover Screws 3 - Cover Hooks

Figure 1 - Removing the ProDAQ Module Cover

To remove the top cover, remove the one countersunk screw in the back and the two panhead screws towards the front panel (②), that hold the cover in place. Remove the cover by sliding it out of its position towards the VXIbus connectors and up. Take special care about the hooks (③) holding it into place. Try not to lift the cover straight up. See Figure 1 for the location of the screws.

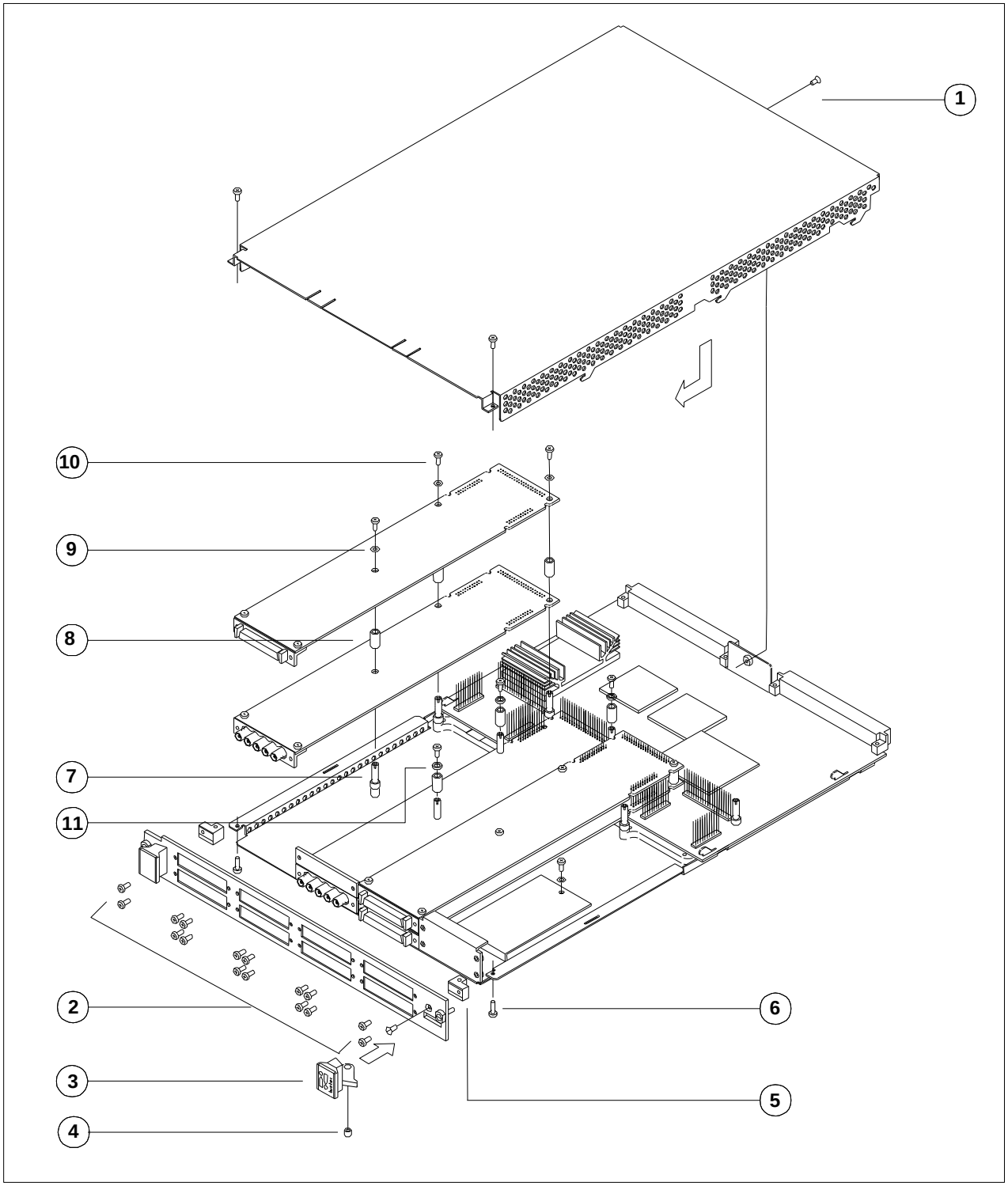
To re-install the cover, slide it back into its position by placing the small hooks over their holes and moving the cover down and forward. Secure the top cover using the two panhead screws and one countersunk screw (②).

2.4. Installing a ProDAQ Function Card

The ProDAQ Function Cards are arranged inside the ProDAQ Module in four stacks of two cards each. The function cards are mounted face down, e.g. the front-panel connectors as well as the motherboard connectors are underneath the PCB.

To install a ProDAQ Function Card in any of the possible positions, use the following procedure (See figure 2 for reference):

- Remove the top cover of the module as described earlier in this chapter (Fig. 2, Pos. 1).
- Remove all screws on the front-panel holding installed function cards or double filler panels in place (Fig. 2, Pos. 2). Screws holding single filler panels don't need to be removed.
- Remove the two panhead screws that mount the front panel to the modules bottom cover (Fig. 2, Pos. 6).
- Please take special care of the module handles and the rings (Fig. 2, Pos. 3 and 4), which are also fixed by those screws. The mounting angle (Fig. 2, Pos. 5) stays fixed to the front panel.
- Remove the front panel by moving it forward carefully so as to avoid bending the installed function cards.
- Choose the stack and position (lower or upper) where you want to mount the function card. If the stack, in which the function card should be installed, is covered by a double filler panel, you have to remove it before installing the function card.
- Remove the three 2.5mm panhead screws and the crinkle washers from the stack's standoffs (Fig. 2, Pos. 9 and 10 for example).
- If you want to install a function card in the upper position of a stack without having a function card in the lower position, you need to mount both spacers (Fig. 3, Pos. 11) on each standoff. If the stack is already populated with a function card in the lower position, mount only the bigger spacer (Fig. 2, Pos. 8) onto each standoff.
- Place a bayonet (supplied) on each standoff. Align the function card over these and slide carefully down. The function card should be held parallel to the modules bottom cover all the time during its way down.
- Fix the function card by mounting the three 2.5mm panhead screws and the crinkle washers onto each standoff. If you install a function card in the lower position of a stack, you need first to mount both spacers (Fig. 2, Pos. 11) onto each standoff.
- Re-mount the modules front-panel. If there is only one function card mounted in a stack, cover the remaining opening in the front panel by a single filler panel.
- Re-mount the modules top cover.



1 - 2.5mm Panhead Screws
2 - 2.5mm Panhead Screws
3 - Module Handle
4 - Ring
5 - Mounting Angle
6 - 2.5mm Panhead Screws
7 - Standoff
8 - Spacer
9 - Crinkle Washer
10 - 2.5mm Panhead Screw
11 - 2mm Spacer

Figure 2 - The ProDAQ Module Assembly

2.5. Removing a ProDAQ Function Card

Removing a ProDAQ Function Card is exactly the reverse operation then installing it. After removing the top cover and the front panel as described previously, remove the three roundhead screws that fix the function card(s) on the standoffs.

Take special care when removing the function card(s) not to bend the motherboard connectors.

After removing the function card(s), install the correct combination of spacers on the standoffs. If a stack is populated with only one function card, each of the standoffs needs to be mounted with both spacers to cover the distance between the cards as well as the PCB thickness of the missing card. If a stack is populated with two function cards, only the bigger spacer must be mounted.

Fix any remaining function card again by mounting the three panhead screws on the standoffs, re-mount the front panel and the modules cover.

3. Theory of Operation

3.1. General Description

The ProDAQ 3610 function card houses 48 channel of TTL compatible digital I/O. The channels are organized in groups of 8 channels, where each group can be programmed individually to function as input or output. Each group consists of two input and two output stages, so that both input and output signals can be read or set either asynchronously under program control or synchronously by a strobe signal (see Figure 3 - ProDAQ 3610 Block Diagram).

3.2. Direct Read/Write Operations

The I/O groups on the 3610 are accesses using group read/write registers, each 8-bit wide. The internal data bus on the ProDAQ 3610 is 16-bit wide, so that groups 1&2, 3&4 or 5&6 can be accessed simultaneously.

3.2.1. Asynchronous Read/Write

When the ProDAQ 3610 is configured for asynchronous I/O, each read or write access to one of the group read/write registers will activate both stages in the input or output path. If a group is configured for output, values written to the registers will immediately set the output levels of the respective channels. Read accesses will return the last value written to the group register. If a group is configured as input, values read from the group register will show the current level of the channels. Write accesses will have no effect.

3.2.2. Synchronous Read/Write

When the card is configured for synchronous I/O, read and write accesses to one of the group registers will access only the second stage in the input/output path. If a group is configured for output, write accesses to the group register will cause the written values to be stored in the second stage buffer. A strobe signal can then synchronously for all channels latch the values from the second stage into the final output stage. For groups configured as input, the strobe signal will cause the synchronous latching of the current input level into the first input stage, where it is stored and can be asynchronously read out by read accesses to the group registers.

3.3. Pattern Acquisition and Generation

The on-board FIFO memory can be used to generate or acquire digital pattern. Each group can be individually programmed whether to take part in the generation or acquisition. The FIFO can only be used to either generate OR acquire a digital pattern. The groups that take part in the pattern acquisition or generation need to be properly configured as input (acquisition) or output (generation) groups. During pattern acquisition or generation no direct read/write operation can be performed.

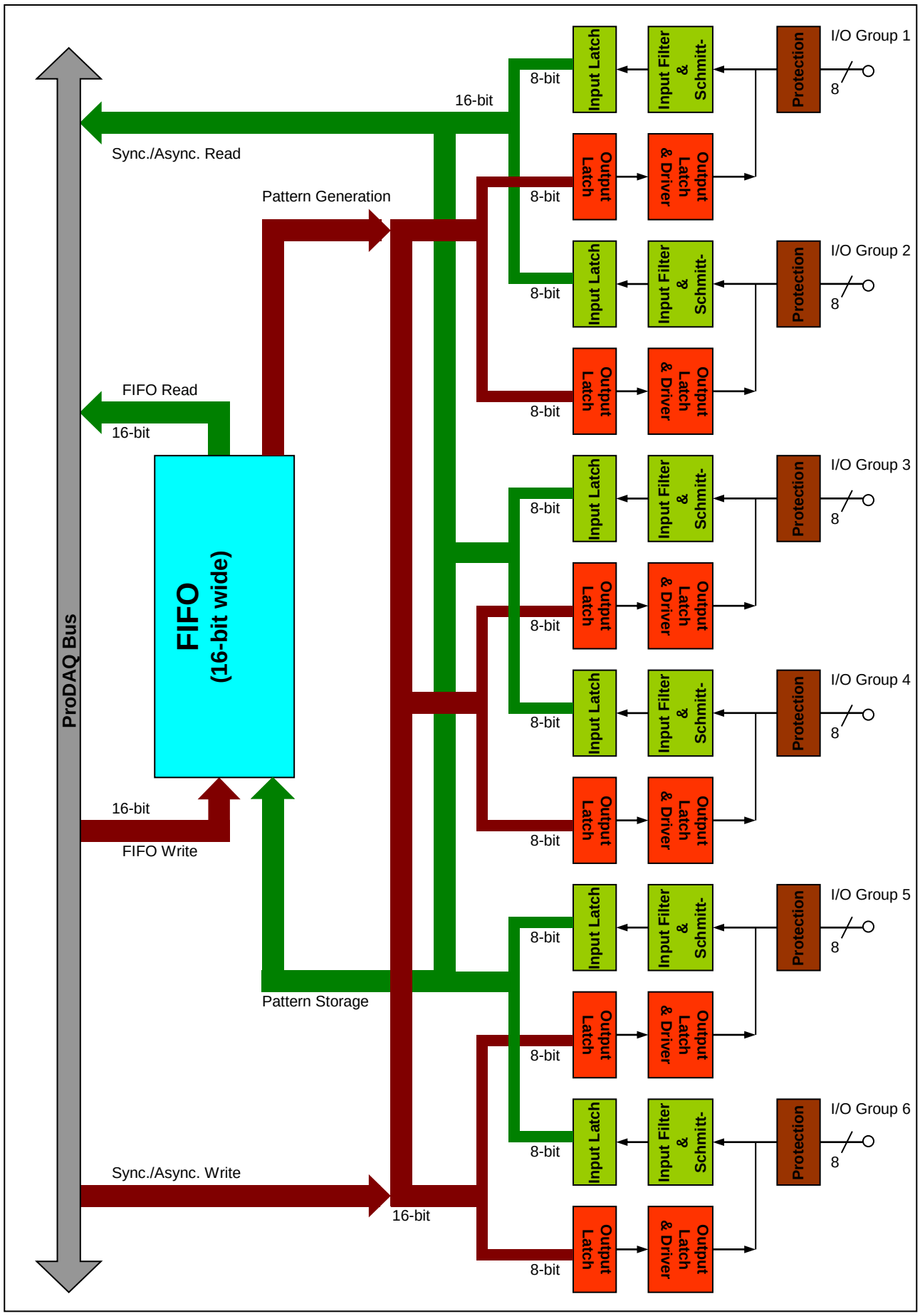


Figure 3 - ProDAQ 3610 Block Diagram

Which group takes part in the pattern acquisition or generation can be selected via the pattern register. However, because of the 16-bit width of the FIFO and internal bus on the 3610, the arrangement of the data in the FIFO is related to the three 16-bit pairs of groups G1&G2, G3&G4 and G5&G6.

FIFO location 1	Data Group 2 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 1 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 2	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 3	Data Group 6 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 5 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 4	Data Group 2 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 1 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 5	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location ...	...	...

Figure 4 - FIFO Data Layout (all groups selected)

If all groups are selected, the first FIFO word is used to store or update the 16-bit register for groups 1 and 2; the second FIFO word to store or update the 16-bit register for groups 3 and 4; the third FIFO word to store or update the 16-bit register for groups 5 and 6; the fourth FIFO word to store or update the 16-bit register for groups 1 and 2 again and so on.

FIFO location 1	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 2	Data Group 6 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 5 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 3	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 4	Data Group 6 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 5 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 5	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location ...	...	...

Figure 5 - FIFO Data Layout (groups 3,4,5,6 selected)

If groups 1 and 2 do not take part in the capture or generation, the first FIFO word will be used to store or update the 16-bit register for groups 3 and 4; the second FIFO word to store or update the 16-bit register for groups 5 and 6; the third FIFO word to store or update the 16-bit register for groups 3 and 4 again and so on.

FIFO location 1	---	Data Group 1 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 2	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 3	Data Group 6 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 5 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 4	---	Data Group 1 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location 5	Data Group 4 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	Data Group 3 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
FIFO location ...	...	...

Figure 6 - FIFO Data Layout (groups 1,3,4,5,6 selected)

If only group 2 is selected not to take part in the pattern acquisition or generation, the whole first FIFO word is still used to store or update the 8-bit register for group 1, not using the upper 8 bit of the FIFO word. If group 1 would not take part in the pattern acquisition or generation, the lower 8 bit of the FIFO word would not be used. In general each combination of selection in a pair of groups will cause the usage of a whole FIFO word to store the related data.

3.4. Trigger

The ProDAQ 3610 can receive trigger signals via the function card trigger line from the motherboard or by using channels 1 to 8 as front-panel trigger inputs. The trigger signal can be used either to gate or clock the pattern acquisition or generation. If the trigger signal is used to gate the pattern acquisition or generation, scans are performed during the time the gate is active. The update/sample rate is defined by the number of group pairs involved in the process. In the clocked mode, either the internal clock source or an external trigger is used to clock the scans.

The internal can generate periods between 1.2µs and 102.4µs (in steps of 400ns), resulting in sample/update frequencies between 9.765 kHz and 833.334 kHz. If an external clock is used, the maximum sample/update frequency depends on the number of groups involved in the capture or generation.

4. The VXI*plug&play* Driver

4.1. Installation

The ProDAQ 3610 48-Ch. Digital I/O function card is supplied with a VXI*plug&play* driver. To install the driver, run the “Setup.exe” application coming with it and follow the instructions presented. Make sure that no other ProDAQ software is running when you start the set up.

The installation program will by default perform a complete installation. It will install the driver files in the directory tree defined by the %VXIPNPPATH% environment variable and shortcuts into the VXIPNP program group of the start menu. To choose a different path and/or custom installation options is not recommended and may result in malfunctioning of the soft front panel and any application trying to use the driver.

4.2. The Soft Front Panel

The purpose of soft front panel application is to demonstrate the instrument’s abilities. After the start of the soft front panel application, the user will be presented with a dialog box showing all available ProDAQ 3610 instruments in a system, allowing the selection of one instrument to connect to (see Figure 7 - Function Card Selection). The soft front panel is not designed to handle more then one instrument at a time. If there is only one instrument available, the dialog box will not appear and the soft front panel application will automatically establish the communication to this instrument.

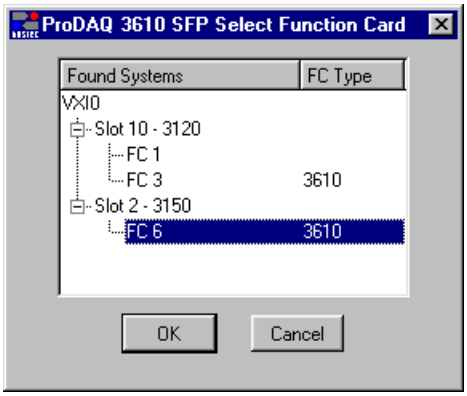


Figure 7 - Function Card Selection

If no ProDAQ 3610 is available in your system, the soft front panel application can be run in demo mode, allowing to operate all controls as if connected to a 3610.

After initializing the ProDAQ 3610 function card, during which a splash screen is displayed, the soft front panel window will appear (see Figure 8). Using the four tab panels displayed the user can select graphical interfaces representing the four top-level functions implemented by the VXIplug&play driver:

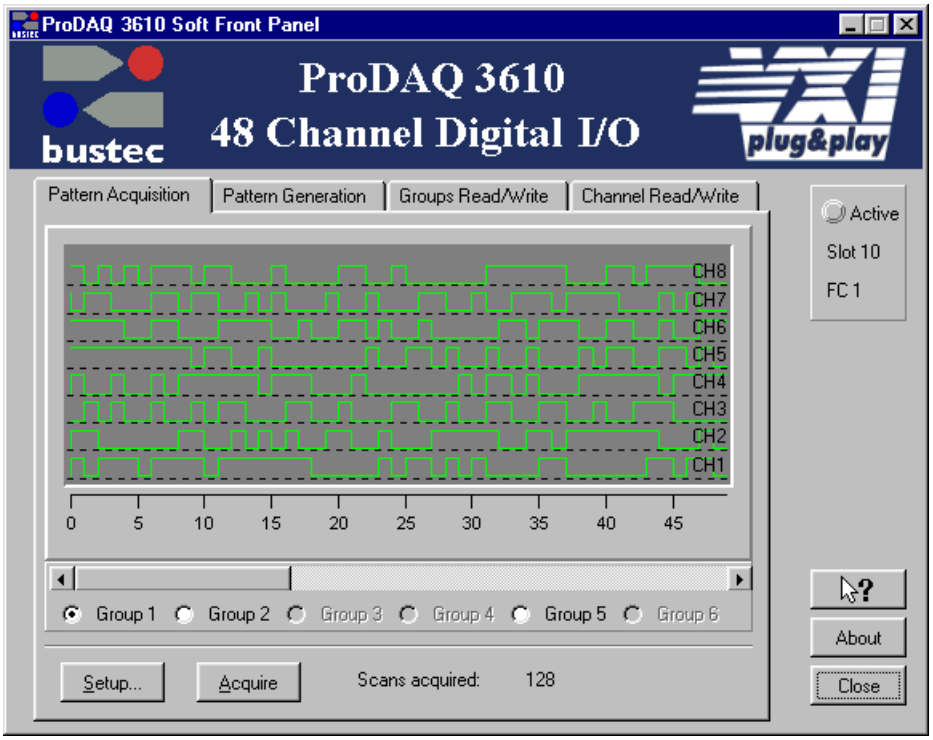


Figure 8 - ProDAQ 3610 Soft Front Panel User Interface

4.2.1. Pattern Acquisition

The controls on the “Pattern Acquisition” page allow to acquire and display a digital pattern. The functionality is equivalent to the function **`bu3610_acquirePattern()`** as implemented by the driver (see 5.4: Pattern Acquisition). The parameter for the pattern acquisition can be changed in the setup dialog displayed using the “Setup...” button in the lower left corner.

Selecting the “Acquire” button will start the pattern acquisition. During the acquisition the number of acquired scans is displayed. When the acquisition is finished, the acquired pattern is displayed in the graph control. Use the radio buttons to select which group is displayed.

4.2.2. Pattern Generation

The controls on the “Pattern Generation” page allow to generate a digital pattern as implemented by the driver function **`bu3610_generatePattern()`** (see 5.5: Pattern Generation). The data used to generate the pattern can be defined using the data editor dialog displayed by the button “FIFO contents”. The arrangement of data in the buffer depends on the I/O groups selected to take part in the generation (See 3.3, Pattern Acquisition and Generation). The data for pattern generation needs always to be arranged by scans.

The I/O groups participating in the generation and all other parameters can be specified directly on the “Pattern Generation” page and the generation can be started by the “Generate” button in the lower left corner. During the generation, the number of scans performed to update the I/O groups is displayed.

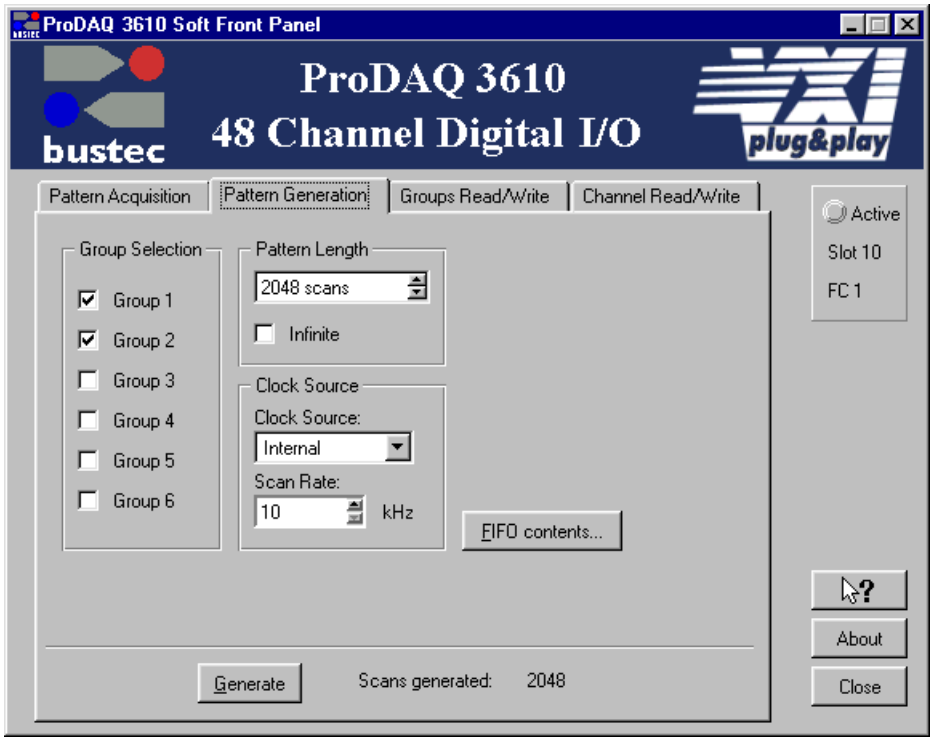


Figure 9 - Pattern Generation Page

4.2.3. Direct Group Read/Write

The third page allows to direct read or write data from/to any or all of the six I/O groups. To include a group in the direct read/write, enable the respective check box on the left, than choose a direction and enter the data in case you want to write to that group. Select one of the two buttons labeled “Read” or “Write” to perform a read or write operation on the I/O groups enabled.

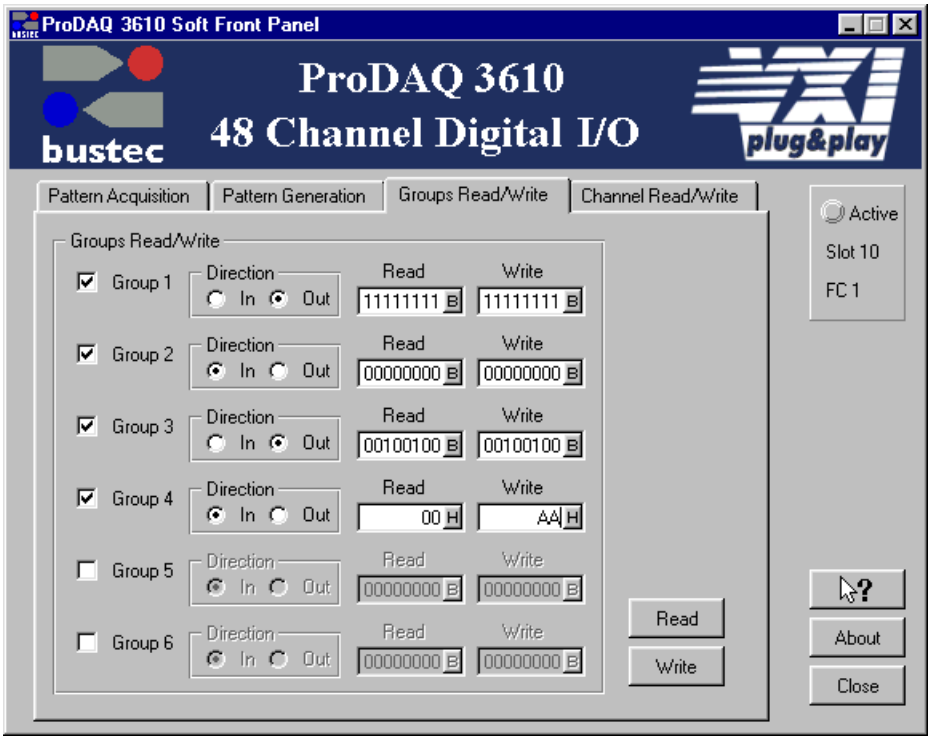


Figure 10 - Direct Group Read/Write

When performing a read operation, groups selected for output will return the last value written to them. Write operations on groups selected for input are ignored.

The functionality implemented by this panel is equivalent to using the driver functions **`bu3610_configureGroups ()`**, **`bu3610_writeGroups()`** and **`bu3610_readGroups()`** as discussed in chapter 5.2 and 5.3.

4.3. Direct Channel Read/Write

The “Channel Read/Write” page allows you to directly perform an read or write operation on a single channel similar to the driver functions bu3610_readChannel() and bu3610_writeChannel().

Figure 11 - Direct Channel Read/Write

Caution: Selecting a channel and using the “Read” or “Write” button to perform the operation may also cause the I/O group containing the channel to change its I/O direction, effecting the other channel in that group.

5. Programming the ProDAQ 3610

This chapter shows how to program the ProDAQ 3610 Digital-I/O function card using the VXI*plug&play* driver. Complete examples can be found in the "Examples" subdirectory of the driver. All functions are explained in detail in the help file coming with the driver.

5.1. Connecting to the Function Card

To initialize the driver and connect to the ProDAQ motherboard, the standard VXI*plug&play* initialization function **`bu3610_init()`** is used (see Figure 12, ①). (Please refer to the VXI*plug&play* standard VPP-4.3, section 4.3 for a detailed description of the address string used.)

After initializing the driver and connecting to the motherboard, the driver must be told which one of the eight possible function cards on a ProDAQ motherboard to work with. This is done by the function **`bu3610_fcSelect()`**. It takes as an argument the session established via the function **`bu3610_init()`**, the function card number and a boolean value specifying whether to reset the selected function card (see Figure 12,②).

```
#include <visa..h>
#include <bu3610.h>

main (int argc, char **argv)
{
     ViStatus status;
     ViSession session;
     ViChar descr[256];


①    if ((status = bu3610_init("VXI0::2::INSTR", VI_TRUE, VI_TRUE, &session)) != VI_SUCCESS)
     {
          viStatusDesc (rm_session, status, descr);
          printf ("Error: bu3610_init() failed due to %s\n", descr);


          return -1;
     }

②    if ((status = bu3610_fcSelect(session, 1, VI_TRUE)) != VI_SUCCESS)
     {
          viStatusDesc (instr_session, status, descr);
          printf ("Error: bu3610_fcSelect failed due to %s\n", descr);

          return -1;
     }

     /* OR: */

     if ((status = bu3610_paramInit("VXI0::2::INSTR", 1, VI_TRUE, VI_TRUE, &session)) != VI_SUCCESS)
③    {
          viStatusDesc (rm_session, status, descr);
          printf ("Error: bu3610_paramInit() failed due to %s\n", descr);


          return -1;
     }

     /* ... */
```

Figure 12 - Opening a Session

For your convenience, the driver contains a new function called **`bu3610_paramInit()`**, which combines the functionality of the **`bu3610_init()`** and **`bu3610_fcSelect()`** functions by extending the argument list of the standard initialization function with a parameter specifying the function card number (see Figure 12,③).

For the driver functions to work properly, you will either have to use the function **bu3610_paramInit()** to open a session with the device, or you will have to call the function **bu3610_fcSelect()** after calling the function **bu3610_init()** and before any other driver function is called.

To close a session with the ProDAQ 3610 48-Ch. Digital-I/O function card, the standard VXIplug&play function **bu3610_close()** must be used.

5.2. Configuring the I/O Groups

After the initialization or any reset all groups on the ProDAQ 3610 function card are configured as input. To configure any of the groups as output, the function **bu3610_configureGroups()** need to be used. It takes as arguments the session to the instrument and six flags, one for each group. Passing a zero (“0”) in any place will configure the respective group as input, a one (“1”) will configure the group as output (see Figure 13, ①).

5.3. Using Direct Read/Write

The VXIplug&play driver for the ProDAQ 3610 48-Ch. Digital-I/O function card provides several functions for the direct read/write access to the channels and I/O groups. For accessing single channels, the two functions **bu3610_readChannel()** and **bu3610_writeChannel()** can be used. They combine the necessary boolean operations to operate on a single bit in an I/O group with the direct read or write access (see Figure 13, ② and ③).

```
    /* ... */

    /* configure groups 1,2,4,6 as input; 3,5 as output */
①   if ((status = bu3610_configureGroups (session, 0, 0, 1, 0, 1, 0)) < VI_SUCCESS)
    {
        viStatusDesc (rm_session, status, descr);
        printf ("Error: bu3610_ configureGroups () failed due to %s\n", descr);

        return -1;
    }

    /* read channel 1 (bit 0 in group 1) */
②   if ((status = bu3610_readChannel (session, 1, &val)) != VI_SUCCESS)
    {
        viStatusDesc (instr_session, status, descr);
        printf ("Error: bu3610_readChannel failed due to %s\n", descr);

        return -1;
    }

    /* set channel 20 (bit 5 in group 3) to 1 (high) */
    if ((status = bu3610_writeChannel (session, 20, 1)) != VI_SUCCESS)
③   {
        viStatusDesc (rm_session, status, descr);
        printf ("Error: bu3610_ writeChannel () failed due to %s\n", descr);

        return -1;
    }


    /* ... */
```

Figure 13 - Using Direct Read/Write Access

Operations on I/O groups instead of single channels are performed passing 16-bit values, where the bits 7-0 represent the values for the single channels in the group. The high-level functions **`bu3610_readGroups()`** and **`bu3610_writeGroups()`** use an array with six elements of such values and a channel mask, where the lowest 6 bit indicate which group shall take part in the read or write operation (see Figure 14, ① and ②).

```
    {
        ViSession session;
        ViInt16  mask, data[6];

        /* .... */

        /* read out groups 2 and 4 (mask bits 1 and 3) */
        mask = 0x0A;
①       if ((status = bu3610_readGroups (session, mask, data)) < VI_SUCCESS)
        {
            viStatusDesc (rm_session, status, descr);
            printf ("Error: bu3610_readGroups () failed due to %s\n", descr);

            return -1;
        }

        /* write to groups 3 and 5 (mask bits 2 and 4) */
        mask = 0x14;
        data[2] = 0x5A;
②       data[4] = 0x11;
        if ((status = bu3610_writeGroups (session, mask, data)) != VI_SUCCESS)
        {
            viStatusDesc (instr_session, status, descr);
            printf ("Error: bu3610_writeGroups failed due to %s\n", descr);

            return -1;
        }

        /* ... */
    }
```

Figure 14 – Direct I/O Group Access

The value for each I/O group has a fixed position in the array, independent whether the respective group is selected for the access by the group mask. Performing a read operation on a group configured for output will return the last value written to the group, but performing a write operation on a group configured for input will return an error.

5.4. Pattern Acquisition

To acquire a digital pattern, the I/O groups selected to take part in the acquisition are scanned with a fixed clock rate and the values are moved to the on-board FIFO memory. The FIFO memory stores the data until the host computer is ready to read out the data. The timing for this asynchronous read-out depends on the amount of data in the FIFO.

The driver function bu3610_acquirePattern() can be used to acquire a pattern of up to 65535 scans length. It configures the function card for pattern acquisition using the parameters *groupMask*, *patternLengthScans*, *clockSource* and *scanRateHz*. As in the other driver functions operating on multiple I/O groups, the parameter *groupMask* defines which I/O group shall take part in the acquisition. The parameter *patternLengthScans* defines the number of scans to be performed, and the parameter *clockSource* and *scanRateHz* define the scan rate. If the constant bu3610_CLK_INTERNAL is chosen as a clock source, *scanRateHz* defines the scan rate in hertz. All other values configure the board to use an external clock, in which case the parameter *scanRateHz* is ignored.


```
{
     ViSession session;
     ViInt16  mask, data[512];

     /* .... */

     /*
      * Acquire a pattern using I/O group 2 and 4
      * The pattern will have a length of 100 scans
      * and will be sampled with 10 kHz.
      */


     mask = 0x0A;
     if ((status = bu3610_acquirePattern (session, mask, 100,
                          bu3610_CLK_INTERNAL, 10000.0, 1, data)) < VI_SUCCESS)
     {
          viStatusDesc (rm_session, status, descr);
          printf ("Error: bu3610_acquirePattern () failed due to %s\n", descr);

          return -1;
     }


     /* ... */
}
```

Figure 15 - Pattern Acquisition

The data return can be arranged by scans or by I/O groups, i.e. if the data is arranged by scans, first all I/O group data for scan one is stored in the buffer, then all I/O group data for scan two is stored and so on. If the data is arranged by I/O groups, first all scans for the first I/O group selected to take part in the acquisition is stored in the buffer, then all data for the second I/O group selected to take part in the acquisition is stored and so on.

5.5. Pattern Generation

To generate a digital pattern using the ProDAQ 3610 48-Ch. Digital I/O function card, first of all a buffer with the pattern to be generated must be prepared. The arrangement of data in the buffer depends on the I/O groups selected to take part in the generation (See 3.3, Pattern Acquisition and Generation). The data for pattern generation needs always to be arranged by scans.

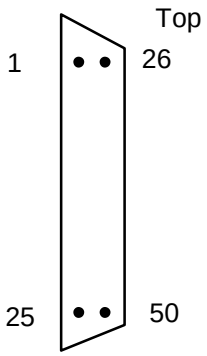
The driver function **`bu3610_generatePattern()`** uses nearly the same parameter as the function **`bu3610_acquirePattern()`**. As mentioned above, the parameter *groupMask* specifies which I/O group will take part in the generation; the parameter *patternLengthScan* specifies the number of scans that will be performed, and the parameter *clockSource* and *scanRateHz* specify the clock used for the sampling (see Figure 16). An additional parameter *patternRepetition* can be used to specify how often the pattern provided should be generated. Using the onboard DSP of the ProDAQ 3150 High-performance Motherboard, the pattern provided can be generated for an infinite amount of time, but in that case the whole pattern needs to be smaller then half the FIFO size of the 3610 and needs to have an even number of scans.


```
{
     ViSession session;
     ViInt16  mask, data[512];

     /* .... */

     /*
      * Generate a pattern using I/O group 2 and 3
      * The pattern will have a length of 100 scans
      * and will be sampled with 10 kHz.
      */

     data[0] = 0x5A00;     /* upper 8-bit contain data for group 2, first scan */
     data[1] = 0x0011;     /* lower 8-bit contain data for group 3, first scan */
     data[0] = 0xA500;     /* upper 8-bit contain data for group 2, second scan */
     data[1] = 0x0022;     /* lower 8-bit contain data for group 3, second scan */
     data[0] = 0x5A00;     /* upper 8-bit contain data for group 2, third scan */
     data[1] = 0x0044;     /* lower 8-bit contain data for group 3, third scan */
     /* ... */

     mask = 0x06;
     if ((status = bu3610_generatePattern (session, mask, 100, 1,
                               bu3610_CLK_INTERNAL, 10000.0, data)) < VI_SUCCESS)
     {
          viStatusDesc (rm_session, status, descr);
          printf ("Error: bu3610_generatePattern () failed due to %s\n", descr);

          return -1;
     }


     /* ... */
}
```

Figure 16 - Pattern Generation

Appendix A: Front Panel Connector

The front panel connector used on the ProDAQ 3610 is a 50-pin female SCSI with the following pin-out:



Signal	Channel	Pin #		Channel	Signal
G1-1	1	1	26	17	G3-1
G1-2	2	2	27	18	G3-2
G1-3	3	3	28	19	G3-3
G1-4	4	4	29	20	G3-4
G1-5	5	5	30	21	G3-5
G1-6	6	6	31	22	G3-6
G1-7	7	7	32	23	G3-7
G1-8	8	8	33	24	G3-8
G2-1	9	9	34	25	G4-1
G2-2	10	10	35	26	G4-2
G2-3	11	11	36	27	G4-3
G2-4	12	12	37	28	G4-4
G2-5	13	13	38	29	G4-5
G2-6	14	14	39	30	G4-6
G2-7	15	15	40	31	G4-7
G2-8	16	16	41	32	G4-8
G5-1	33	17	42	41	G6-1
G5-2	34	18	43	42	G6-2
G5-3	35	19	44	43	G6-3
G5-4	36	20	45	44	G6-4
G5-5	37	21	46	45	G6-5
G5-6	38	22	47	46	G6-6
G5-7	39	23	48	47	G6-7
G5-8	40	24	49	48	G6-8
GND	--	25	50	--	GND

Appendix B: Register Description

A.1 Address Map

All addresses are given in a hexadecimal notation. “FC Address” specifies the address in the internal function card address space of the motherboard. “VXI Offset” is the offset to be used when accessing this register directly via the function card space mapped into the VXI memory space of the motherboard.

FC Address	VXI Offset	Register Name	Access	Function
Control Register				
0	0	FCID_REG	RO	ID register for automatic board identification
1	4	---	-	Not used
2	8	GCSR	RWC	General control and status register
3	C	FCLEN	RO	Size of installed FIFO (2048 or 16384 kWords)
4	10	---	-	Not used
5	14	OTRI	RW	Output trigger control
6	18	ITRI	RW	Input trigger control
7	1C	DIVCLK	RW	Clock divider for use in the stand-alone mode
8	20	MODE	RW	Operation mode register
9	24	REPCOUNT	RW	Repetition Counter
A	28	---	RW	Not used
B	2C	PATTERN	RW	Selects groups for pattern capture/generation
C	30	OENA	RW	Output enable register
D	34	---		Not used
E	38	DG0R	RW	Group 1 trigger status
Group Register				
80	200	G1	RW	Group 1 direct access register
81	204	G2	RW	Group 2 direct access register
82	208	G3	RW	Group 3 direct access register
83	20C	G4	RW	Group 4 direct access register
84	210	G5	RW	Group 5 direct access register
85	214	G6	RW	Group 6 direct access register
86	218	G1_2	RW	Group 1&2 direct access register
87	21C	G3_4	RW	Group 3&4 direct access register
88	220	G5_6	RW	Group 5&6 direct access register
Memory Space				
4000-FFFF	20000-3FFFC	FIFO	RW	FIFO access area

All registers appear are 16-bit wide.

A.2 Detailed Register Description

A.1.1 FCID Register

The function card ID register contains a unique number used to identify the card.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Initial	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0
Content	Function Card ID (0x7070)															

A.1.2 GCSR Register

General control and status register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RC	RC	RC	RC	RO	RO	RO	RC	--	--	RWC	RW	RWC	RW	RWC	WC
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	FIFO_EM	TRIG_I	REPZERO	TRIG_O	FIFO_AE	FIFO_HF	FIFO_AF	FIFO_FF	n/a	n/a	SINGLE	TRG_START	ENA	START	STOP	RESET

RESET Resets the function card.

STOP Writing a one (“1”) to this bit stops the pattern capture/generation. The bit is automatically cleared when the capture/generation has stopped.

START Writing a one (“1”) to this bit starts the pattern capture/generation. The bit is automatically cleared when the capture/generation is finished or has been stopped.

ENA Enables the pattern capture/generation.

TRG_START Enables the pattern capture/generation to react on the trigger selected by the ITRI register.

SINGLE Writing a one (“1”) to this bit will start the pattern capture/generation for one cycle only.

FIFO_FF A one (“1”) indicates that the FIFO memory is full.

FIFO_AF A one (“1”) indicates that the FIFO memory is almost full (less than 128 words free).

FIFO_HF A one (“1”) indicates that the FIFO memory is half full.

FIFO_AE A one (“1”) indicates that the FIFO memory is almost empty (less than 128 words in FIFO).

TRIG_O A one (“1”) indicates that an output trigger as selected by the OTRI register was generated. This bit is cleared by reading, but also by starting the pattern capture/generation or reset.

REPZERO A one (“1”) in this bit indicates that the internal repetition counter is zero. This bit is cleared by reading, but also by starting the pattern capture/generation or reset.

TRIG_I A one (“1”) indicates that an input trigger as selected by the OTRI register was detected. This bit is cleared by reading, but also by starting the pattern capture/generation or reset.

FIFO_EM A one (“1”) indicates that the FIFO memory is empty.

A.1.3 FCLEN Register

Specifies the FIFO size in words.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RC	RC	RC	RC	RO	RO	RO	RC	--	--	RW	RW	RWC	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	FIFO size 3610-AA: 0x0800 (2048 words) 3610-AB: 0x4000 (16384 words)															

A.1.4 OTRI Register

Register to configure the output trigger.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RW	RW	RW	RW	--	RW	RW	RW	RW	RW	RW	RW	--	--	--
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	STAT	INT_TRG			POL	n/u	OMP	OMB	SWT	MBT	DIVC	ISEL	ITRIS	not used		

ITRIS A one “1” in this bit enables the input trigger from channel 1-8 as selected in the ITRI register as a source for the output trigger.

ISEL Writing a one (“1”) to this bit enables the internal trigger selector.

DIVC Writing a one (“1”) to this bit enables the sample clock as specified in the DIVCLK register as a source for the output trigger.

MBT Enables the input trigger from the motherboard as a source for the output trigger.

SWT Writing a one (“1”) to this bit activates the output trigger (Software Trigger).

OMB Writing a one (“1”) to this bit routes the output trigger to the motherboard.

OMP Writing a one (“1”) to this bit routes the output trigger to the motherboard. The trigger is generated as a pulse with a width of 125 ns.

POL Sets the polarity of the output. Writing a one “1” to this bits selects the output trigger to be generated as a “active high” signal. A zero (“0”) selects the output trigger to be generated as a “active low” signal.

INT_TRG Selects one of the internal sources as the source for the output trigger:

Value	Source enabled
0	Pattern capture/generation finished
1	Repeat counter zero
2	FIFO full
3	FIFO almost full
4	FIFO almost empty
5	FIFO half full
6	FIFO empty
7	Pattern capture/gen. scan finished

STAT Shows the status of the output trigger. Reading a zero (“0”) indicates that the trigger is not active, reading a one (“1”) indicates that it is active.

A.1.5 ITRI Register

Configures the input trigger of the function card.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RO	RW	RW	RW	RW	--	--	--	RW	RW	RW	RW	--	--	--	--
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	STAT	INSELECT			POL	not used			SWT	MBT	DIVC	ISEL	not used			

ISEL Writing a one (“1”) to this bit enables the internal trigger selector.

DIVC Writing a one (“1”) to this bit enables the sample clock as specified in the DIVCLK register as a source for the input trigger.

MBT Enables the input trigger from the motherboard as a source for the input trigger.

SWT Writing a one (“1”) to this bit activates the input trigger (Software Trigger).

POL Sets the polarity of the output.

INSELECT Selects one of the channel in group 1 as the source for the input trigger:

Value	Source enabled
0	Channel 1
1	Channel 2
2	Channel 3
3	Channel 4
4	Channel 5
5	Channel 6
6	Channel 7
7	Channel 8

STAT Shows the status of the input trigger. Reading a zero (“0”) indicates that the trigger is not active, reading a one (“1”) indicates that it is active.

A.1.6 DIVCLK Register

This register holds the divider value for deriving the sample clock from the internal clock source.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0
Content	not used								divider value							

The base clock for the divider has a period of 400ns. Valid values to be set are 2 to 255, resulting in divider values of 3 to 256, or sample clock periods of 1.2µs to 102.4µs.

A.1.7 MODE Register

The function card ID register contains a unique number used to identify the card.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	--	RW	--	--	RW	RW	RW	RW	RW	RWC	RW	RW	RW
Initial	0	0	0	X	0	X	X	0	0	0	0	0	0	0	0	0
Content	FIDC	FIDD	ADAI	n/u	ATC	n/u	n/u	SWT	SRT	EOE	FIRS	FIEN	OCW	IOG	IOR	FIWB

FIWB Enables/disables FIFO loopback mode.

ICR Enables/disables asynchronous reading.

IOG Enables/disables asynchronous reading for register DG0R.

OCW Enables/disables asynchronous writing.

FIEN Enables the FIFO for pattern capture/generation.

FIRS Resets the FIFO.

SRT

SWT

EOE General output enable.

ATC Enables the pattern capture/generation in continuous mode.

ADAI Enables the pattern capture/generation to run infinitive. Otherwise it will stop after the number of scans programmed in the REPCOUNT register.

FIDD Disable the FIFO drop mode (enabled by default).

FIDC enable FIFO clock.

A.1.8 REPCOUNT Register

This register holds the repetition counter for the pattern capture/generation.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Content	Repetition Counter															

The repetition counter is a 16-bit down counter used to specify how many words of a pattern shall be captured or generated. The number of scans performs is register value plus one.

A.1.9 PATTERN Register

Defines which channel groups take part in the pattern capture/generation and whether data shall be captured or generated.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0
Content	not used									DIR	G6	G5	G4	G3	G2	G1

G1 – G6 A one (“1”) in one of the bits enables the respective group to take part in the pattern capture/generation.

DIR A one (“1”) in this location specifies a pattern to be generated. A zero (“0”) specifies a pattern to be captured.

A.1.10 OENA Register

Defines the input/output mode for the groups for direct read/write.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0
Content	not used										G6	G5	G4	G3	G2	G1

G1 – G6 A zero (“0”) in one of the bits selects the respective group to be an input group; a one (“1”) selects it to be an output group. For any output to take place, the general output enable bit in the MODE register has to be enabled.

A.1.11 DG0R Register

Shows the status of group 1 when used as direct trigger input lines.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	--	--	--	--	--	--	--	--
Content	not used								CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

CH8 – CH1 Current level of the input lines of group 1.

A.1.12 G1 Register

Read/write register for direct accesses to group 1.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	--	--	--	--	--	--	--	--
Content	NOT USED								CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

A.1.13 G2 Register

Read/write register for direct accesses to group 2.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	--	--	--	--	--	--	--	--
Initial	--	--	--	--	--	--	--	--	X	X	X	X	X	X	X	X
Content	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	NOT USED							

A.1.14 G3 Register

Read/write register for direct accesses to group 3.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	--	--	--	--	--	--	--	--
Content	NOT USED								CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17

A.1.15 G4 Register

Read/write register for direct accesses to group 4.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	--	--	--	--	--	--	--	--
Initial	--	--	--	--	--	--	--	--	X	X	X	X	X	X	X	X
Content	CH32	CH31	CH30	CH29	CH28	CH27	CH26	CH25	NOT USED							

A.1.16 G5 Register

Read/write register for direct accesses to group 5.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	--	--	--	--	--	--	--	--	RW	RW	RW	RW	RW	RW	RW	RW
Initial	X	X	X	X	X	X	X	X	--	--	--	--	--	--	--	--
Content	NOT USED								CH40	CH39	CH38	CH37	CH36	CH35	CH34	CH33

A.1.17 G6 Register

Read/write register for direct accesses to group 6.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	--	--	--	--	--	--	--	--
Initial	--	--	--	--	--	--	--	--	X	X	X	X	X	X	X	X
Content	CH48	CH47	CH46	CH45	CH44	CH43	CH42	CH41	NOT USED							

A.1.18 G1G2 Register

Read/write register for direct accesses to groups 1 and 2.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Content	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1

A.1.19 G3G4 Register

Read/write register for direct accesses to groups 3 and 4.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Content	CH32	CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17

A.1.20 G5G6 Register

Read/write register for direct accesses to groups 5 and 6.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operation	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Content	CH48	CH47	CH46	CH45	CH44	CH43	CH42	CH41	CH40	CH39	CH38	CH37	CH36	CH35	CH34	CH33

Appendix C: Specifications

<table>
<tr><td colspan="2">Number of input/output channels</td><td>48</td></tr>
<tr><td rowspan="5">Inputs</td><td>Level</td><td>high: > 2.0V
low: < 0.8V</td></tr>
<tr><td>Current</td><td>±1µA</td></tr>
<tr><td>Filter</td><td>RC Filter (510 Ω/100pF)</td></tr>
<tr><td>Detection</td><td>Schmitt-Trigger (0.8V Hysteresis)</td></tr>
<tr><td>Overvoltage Protection</td><td>6.5 Volts</td></tr>
<tr><td rowspan="2">Outputs</td><td>Steady State Low</td><td>max. 0.6V @ 0.8mA</td></tr>
<tr><td>Steady State High</td><td>min. 2.4V @ -8mA</td></tr>
<tr><td colspan="2">Data Throughput</td><td>Max. 6 MByte/sec:
<table><tr><th># of Channels</th><th>Max. I/O Frequency</th></tr><tr><td>48</td><td>1 MHz</td></tr><tr><td>32</td><td>1.5 MHz</td></tr><tr><td>16</td><td>3 MHz</td></tr></table></td></tr>
<tr><td colspan="2">FIFO</td><td>2 kWord (-AA) or 16 kWord (-AB)</td></tr>
<tr><td colspan="2">Trigger Inputs</td><td>Motherboard or Front Panel (Channel 1 to 8)</td></tr>
<tr><td colspan="2">Current Consumption</td><td>120mA @ 5V</td></tr>
<tr><td colspan="2">Power Consumption</td><td>< 0.6 Watt</td></tr>
<tr><td colspan="2">Connector</td><td>50-pin SCSI</td></tr>
<tr><td colspan="2">Dimensions</td><td>230mm x 53mm (9.1inch x 2.1inch)</td></tr>
<tr><td colspan="2">Weight</td><td>< 100 g.</td></tr>
<tr><td colspan="2">Operating Temperature</td><td>0°C to 50°C</td></tr>
<tr><td colspan="2">Storage Temperature</td><td>-40°C to 70°C</td></tr>
<tr><td colspan="2">Warm-up Time</td><td>< 30 min.</td></tr>
<tr><td colspan="2">MTBF</td><td>93520 Hrs.</td></tr>
<tr><td colspan="2">Software Support</td><td>VXIplug&play Driver</td></tr>
</table>

Bustec Production, Ltd.
World Aviation Park, Shannon, Co. Clare, Ireland
Tel: +353 (0) 61 707100, FAX: +353 (0) 61 707106

Bustec, Inc.
E346, 34428 Yucaipa Blvd., Yucaipa, CA 92399, U.S.A
Tel. +1 (909) 797 0484, Fax: +1 (760) 751 1284