

Instrument Bus Tester - Users Manual

SC2_ELE_S58E_501

Version 3.0 - tfelton 06/10/06

Overview

Scuba-2 circuit board SC2_ELE_S58E_101 is used to test the Instrument Backplane. The Instrument Bus Tester (IBT) is a printed circuit board on a standard Scuba-2 200mm x 6U form factor. The IBT is a GPIB device that is essentially a multi-channel multiplexer. By issuing GPIB commands the multiplexers on the IBT may be set to test for short circuits, open circuits and continuity on the Instrument Bus.

The board contains an Atmel 89C55 processor programmed in C which converts the GPIB ASCII commands into the correct multiplexer setup. The IBT has 2 analog front panel connector points (banana jacks) labeled Signal High and Signal Low. The GPIB commands are very specific low level commands and essential specify which of the 48 Instrument Bus signals is connected to Signal High and which is connected to Signal Low.

A National Instruments Labview program controls the testing. Labview talks to 2 GPIB devices. One device is the IBT the other is a 4 wire ohmmeter (Agilent 34401A or similar) that is connected to Signal High and Low on the IBT. A series of GPIB commands would specify the pin to connect to Signal High, specify the pin to connect to Signal Low and finally read the impedance between Signal High and Low. The Labview program accepts commands from an Excel spreadsheet which contains a list of GPIB commands. The Labview program is driven by the data in the spreadsheet. Using this method simply adding or deleting data from the spreadsheet will change the tests performed on the IBT. The impedance readings are also stored in the spreadsheet. See the section "Labview Test Program" for a more detailed explanation.

GPIB Commands and Responses

The IBT responds to a series of ASCII GPIB commands. Checking for ACK/NAK is optional. The commands are as follows:

V – Version Command

- generic command/example command: "V"
- current response is "UBC InstBusMux v1.06"

R – Reset Command

- generic command/example command: "R"

P – Set Positive Multiplexer Command

- generic command: "P" ibp-pin
- example command: "PB12"
- 'ACK' (0x06) sent when command recognized and multiplexer selected

Instrument Bus Tester - Users Manual

SC2_ELE_S58E_501

Version 3.0 - tfelton 06/10/06

N – Set Negative Multiplexer Command

- generic command: "N" ibp-pin
- example command: "NA01"
- 'ACK' (0x15) sent when command recognized and multiplexer selected

S – Slot ID Command

- generic command/example command: "S"
- response is a number ASCII number '0' to '9'

The following are some GPIB command definitions:

command-string = ibt-command end-of-line .

end-of-line = carriage-return | line-feed .

carriage-return = 0x0d .

line-feed = 0x0a .

ibt-command = pos-mux-select | neg-mux-select | reset .

reset = 'R'

version = 'V'

slot_id = 'S'

pos-mux-select = "P" ibp-pin .

neg-mux-select = "N" ibp-pin .

ibp-pin = ibp-row ibp-column .

ibp-row = "A" | "B" | "C" | "D" .

ibp-column = digit digit .

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" .

All commands are to be terminated in either a CR or LF character; they are equivalent as far as the IBT is concerned. The commands are case-sensitive. Examples of commands are:

R Perform a soft reset of the IBT.

PB13 Connect the positive probe multiplexer to pin B13.

NC22 Connect the negative probe multiplexer to pin C22.

In response to commands the IBT will issue an ASCII 'ACK' (0x06) when a multiplexer select command is recognized and executed. The Version Command and Slot ID command have defined responses. Any command that cannot be parsed will result in an ASCII 'NAK' (0x15) being returned.

Instrument Bus Tester - Users Manual

SC2_ELE_S58E_501

Version 3.0 - tfelton 06/10/06

Microcontroller Firmware Notes

The multiplexer line order had to be changed to facilitate routing. Also, it is necessary to shut off the two other multiplexers. The Atmel firmware follows the following:

```
unsigned char command[10], mux_address;
int row, column;

const unsigned char mux_lut[96] = { /* populate with pin-mapping from schematic */ };

strcpy(command, "NC09\n"); // For example.

// Look at command[0] to determine the command.
// The following assumes we've decoded the "N":

command[4] = '\0'; // Ignore anything after the first 2 digits.
column = atoi(&command[2]) - 1;
row = command[1] - 'A';

if (row >= 0) && (row <= 3) && (column >= 0) && (column <= 23)
{
    mux_address = mux_lut[ (row * 24) + column ];
}

// The positive multiplexers are 0, 1 and 2.
// The negative multiplexers are 3, 4 and 5.

// Turn on the one we want:
mux_select( 3 + (mux_address / 24), mux_address % 24 );
```

When using Labview 8 the GPIB-IB tester_ver-b.vi should be used. Labview made changes to their "Report Generation Toolkit for Microsoft Office" in version 8 such that our previous vi only worked with Labview 7. The changes were very minor with respect to our application.

Instrument Bus Tester - Users Manual

SC2_ELE_S58E_501

Version 3.0 - tfelton 06/10/06

Labview Test Program

The Labview program is able to read and write to a specified Microsoft Excel file using an input template file and create a new file from the template which contains the data that is read. The program allows some flexibility in the layout of the Excel file but this has not been tested as the default configuration matches our current requirements. This means that the currently available template files match the default setting in the Labview Program. The address of the 2 GPIB devices is programmable, but again for the same reasons, this has not been tested. Currently the InstBusMux is Device 10 (fixed in firmware) and the Multimeter is Device 2. The Excel spreadsheet files have 3 columns of commands and one column for recorded readings from the Multimeter. Each row is a sequence of commands required to get a valid reading. Columns labeled Cmd1 and Cmd2 are commands for the MUX, column Cmd3 is for the Multimeter and column "Reading" is for the results of the Cmd3 command. The first sequence number "Slot ID" can be used to determine which slot the MUX Tester is plugged into. A number 0-9 is returned where 0 is the far left slot.

The slot column in the spreadsheet is used to indicate which signals are found on that slot. If the slot column matches the slot that the GPIB card is plugged into, then the sequence of commands in that row are issued and the readings are recorded. If the slot column does not match then no readings are taken and the program checks the next row. The program steps through each row issuing commands whenever the slot number in the row matches the slot the card is plugged into. When the program gets to the bottom of the spreadsheet the user has a choice to end the program or move the tester to another slot. The end of the spreadsheet is indicated with an * in column 1 of the last row. The MUX card can be repositioned into the next slot and the program can be run again. This should be done under power using the external power input on the front panel. When all slots have been tested the program can be halted. At this point the operator specifies a file to save the data into. This method allows us to accumulate the data from one subrack into a single file.

The program is very general purpose and allows a variety of tests to be defined simply by changing the command list in the Excel spreadsheet.