TEXIO
Test and Measurement Solutions

INSTRUCTION MANUAL

GP-IB INTERFACE UNIT
PU SERIES

B71-0134-00

■ About Brands and Trademarks

"TEXIO" is the product brand name of our industrial electronic devices.

All company names and product names mentioned in this manual are the trademark or the registered trademark of each company or group in each country and region.

■ About the Instruction Manual

CONTENTS

USING THE PRODUCT SAFELY

■ Preface

To use the product safely, read instruction manual to the end.
Before using this product, understand how to correctly use it.
If you read the manuals but you do not understand how to use it, ask us or your local dealer.
After you read the manuals, save it so that you can read it anytime as required.

■ Pictorial indication

The manuals and product show the warning and caution items required to safely use the product.
The following pictorial indication and warning character indication are provided.

<Pictorial indication>	
	Some part of this product or the manuals may show this pictorial indication. In this case, if the product is incorrectly used in that part, a serious danger may be brought about on the user's body or the product. To use the part with this pictorial indication, be sure to refer to the manuals.
	If you use the product, ignoring this indication, you may get killed or seriously injured. This indication shows that the warning item to avoid the danger is provided.
	If you incorrectly use the product, ignoring this indication, you may get slightly injured or the product may be damaged. This indication shows that the caution item to avoid the danger is provided.

Please be informed that we are not responsible for any damages to the user or to the third person, arising from malfunctions or other failures due to wrong use of the product or incorrect operation, except such responsibility for damages as required by law.

USING THE PRODUCT SAFELY

■ Do not remove the product's covers and panels

Never remove the product's covers and panels for any purpose.
Otherwise, the user's electric shock or fire may be incurred.

■ Warning on using the product

Warning items given below are to avoid danger to user's body and life and avoid the damage or deterioration of the product. Use the product, observing the following warning and caution items.

■ Warning item on abnormality while in use

If smoke or fire is generated from the product while in use, stop using the product, turn off the switch, and remove the power cord plug from the outlet. After confirming that no other devices catch fire, ask us or your local dealer.

■ Calibration

Although the performance and specifications of the product are checked under strict quality control during shipment from the factory, they may be deviated more or less by deterioration of parts due to their aging or others.
It is recommended to periodically calibrate the product so that it is used with its performance and specifications stable.
For consultation about the product calibration, ask us or your local dealer.

■ Daily Maintenance

When you clean off the dirt of the product covers, panels, and knobs, avoid solvents such as thinner and benzene. Otherwise, the paint may peel off or resin surface may be affected. To wipe off the covers, panels, and knobs, use a soft cloth with neutral detergent in it.
During cleaning, be careful that water, detergents, or other foreign matters do not get into the product.
If a liquid or metal gets into the product, an electric shock and fire are caused.
During cleaning, remove the power cord plug from the outlet.

Use the product correctly and safely, observing the above warning and caution items.
Because the instruction manual indicates caution items even in individual items, observe those caution items to correctly use the product.

If you have questions or comments about the manuals, ask us or E-Mail us.

1. THE DIGITAL (GP-IB-488.2 SCPI) PROGRAMMING OPTION

1-1. Introduction

The internal factory installed GP-IB interface allows to operate the PU Power supply from a computer via GP-IB-488 communication bus.

The GP-IB interface allows the user complete remote control of the power supply, including output voltage and current limit programming, setting the Over Voltage Protection, Under Voltage Limit and Foldback protection. The Output Voltage and Output Current can be measured and the power supply status can be monitored.

Commands that are standard with digital programming include:

· Program Voltage	· Program Current
· Measure Voltage	· Measure Current
· Over-Voltage Shutdown	· Current Fold Back Shutdown
· GP-IB-488.2 Compliant	· SCPI Compliant

1-2. Scope of manual

This manual contains the information needed to operate the optional embedded digital interface used in the Power Supply. This manual does not include specifications for digital accuracy and response rate. These values are only valid for the power supply in which the interface is installed, so the specifications are given in the User Manual for the power supply.

1-3. Using digital programming

1-3-1. The GP-IB-488.2 interface

The GP-IB-488 digital programming interface (also called the GP-IB interface) is a popular way to connect instruments to a computer. It uses a specialized 24-pin cable with connector that allows cables to be 'stacked' together. There are eight data wires, eight control wires and eight ground wires. If the system runs from a personal computer, there are numerous vendors of GP-IB controller cards and software.

The GP-IB-488 standard has gone through several upgrades. The GP-IB-488.1 focused on the handshaking of the eight control lines. The GP-IB-488.2 added status registers inside each instrument and it added common commands to make programming groups of instrument easier. The latest specification, SCPI, adds guidelines for the command syntax so one vendor's power supply will use the same commands as another's. The Interface follows all of these standards.

Because many instruments may be connected and independently controlled by a single GP-IB controller, each instrument must have a unique address. The GP-IB controller automatically sets its address equal to the power supply address.

1-4. Configuring the GP-IB interface

1-4-1. Setting the GP-IB select switch

The interface contains a two position DIP switch that is accessible from the rear of the Power Supply and located next to the GP-IB cable connector.　Switch 2, located to the right of Switch1, is not used.　Refer to Fig 1-1 for location of the GP-IB connector and the GP-IB select switch at the rear panel of the supply.

Fig 1-1: GP-IB connector and GP-IB select switch location

Placing Switch 1 in the ON Position, Up, will activate this interface and deactivate the Power Supply’s Serial I/O capability.　Placing Switch 1 in the OFF Position, Down, will deactivate this interface and activate the Power Supply’s Serial I/O capability.

The position of Switch 1 may be changed at any time; but, if a new setting is to take effect, the Power Supply input power must be power cycled.

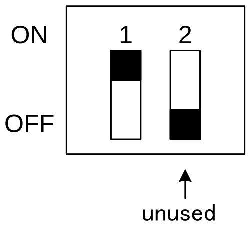


Fig 1-2: GP-IB select switch

1-4-2. Figuring the GP-IB controller

A typical GP-IB controller is a personal computer with an GP-IB interface card.　Each card vendor supplies its own configuration instructions and interface software.

Each time the software is executed, the controller must be configured as follows:

- Controller Address = 0. This is factory default for all controllers.
- EOI Flag = TRUE.　The “End or Identify” is a control line in the GP-IB cable that is asserted when the last character of a message string is sent.　It is required for this interface.
- EOS Flag = FALSE: The “End of String”, used in some instruments to indicate the last character of a message, is not supported by this interface.

1-5. The interface input buffer

The Interface contains a 208 byte input buffer to save commands as they are received from the GP-IB-488.2 bus. The buffer is divided into sixteen 13-byte fields. Thus the command:
SOURCE: VOLTAGE: AMPLITUDE 123.45 will consume 4 fields and the command:
: VOLTAGE 123.45 will consume 2 fields.

The user may enter more than 1 concatenated command, separated by semicolons, to be executed. The interface will process all commands before returning any data/status to the GP-IB bus or accepting any new commands to execute. Messages returned to the user will be the result of the last command executed.

If a command error exists, or an error is reported by the Power Supply, all subsequent commands in the buffer will be terminated and the status returned to the user.

If the user enters more that 16 fields an error will be generated. Also, if the user enters more than 13 bytes in any field, an error will be generated.

1-6. Getting started with the software

A computer can use a variety of controllers, programs, and programming languages for the GP-IB bus.
Here is an example showing a minimal program to set the voltage, set the current and measure the voltage from a power supply.

1-6-1. Example session using the 'IBIC' console

A popular console program is National Instruments "Win32 Interactive Control" (file: ibic.exe). As the operator types each command on the computer, at the colon prompt, it is immediately sent to the power supply. This example works only for computers with National Instruments and compatible GP-IB controller cards.

```
Win32 Interactive Control
Copyright 1996 National Instruments Corporation
All rights reserved.

Type 'help' for help or 'q' to quit.

 : ibdev
   enter board index: 0                  <----  Controller address
   enter primary address: 6              <----  Supply address
   enter secondary address: 0
   enter timeout: 12
   enter 'EOI on last byte' flag: 1
   enter end-of-string mode/byte: 10

 ud0: ibwrt "sour: volt 100"             <----  Program supply to
 [0100] (cmpl)                                  100 volts output
 count: 12

 ud0: ibwrt "sour: curr 5"               <----  Program supply to
 [0100] (cmpl)                                  5 amps output
 count: 11

 ud0: ibwrt "meas: volt?"                <----  Query: "What is output voltage?"
 [0100] (cmpl)
 count: 10

 ud0: ibrd 50                            <----  Read response
 [2100] (end cmpl)                         Supply reported output voltage
 count: 11

 31 30 30 2e 30 38 0a      100.08.       <----
```

1-6-2. Example program written in visual basic

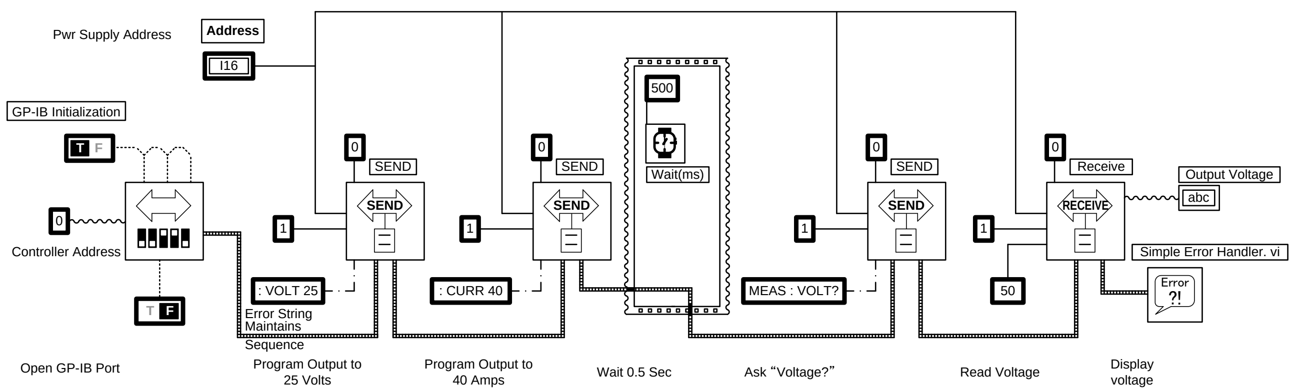
Microsoft's Visual Basic is a windows programming language that may be used to create "virtual instruments" and automation programs. Here is a simple program, which sends commands to a power supply to set the voltage, set the current and measure the voltage. The program's window only contains two items: a "Start" button and a text box to show the measured voltage. The syntax of the CALLed functions is correct only for National Instruments and compatible GP-IB controllers. Don't forget to add the forms "Ni-global.bas" and "Vbib-32.bas" to your project.

Example Program Written in Visual Basic

```
Option Explicit
Dim SupplyUD As Integer              'supply device descriptor
Dim strMeasVolt As String * 50       'buffer for reading input message

  Private Declare Function GetTickCount Lib "kernel32" () As Long

  Private Sub cmdStart_Click()
  'start program here after "Start" button clicked
        'open GP-IB port, get "User Device Description" = SupplyUD
      'assume power supply address is set to "6" on DIP switch
    Call ibdev (0, 6, 0, T3s, 1, 10, SupplyUD)

    Call ibwrt (SupplyUD, ":volt 100")    'program output to 100 volts
    Call ibwrt (SupplyUD, ": curr 2")     'program output to 2 amps
    Wait 500          'wait 0.5 sec to settle
    Call ibwrt (SupplyUD, "meas: volt?")    'ask "What is output voltage?"
    Call ibrd (SupplyUD, strMeasVolt)     'read back output voltage
    txtOutVolt.Text = strMeasVolt      'display output voltage on window
  End Sub

  Private Sub Wait (mSecWait As Long)
  'subroutine to wait "mSecWait" milliseconds
    Dim StartTime As Long
    StartTime = GetTickCount
    Do
    Loop While (GetTickCount - StartTime < mSecWait)
  End Sub
```

1-6-3. Example program written in labVIEW

The National Instruments LabVIEW programming language is a popular language, which is optimized for instrument control and data analysis. It is a graphical language where functions are shown as icons with connection points and data flows along drawn lines. Here is a simple program, which sends commands to a power supply to set the voltage, set the current and measure the voltage. The program's window only contains two items: a numeric control for the supply GP-IB address and a text indicator to show the measured voltage.



2. PROGRAMMING COMMANDS

2-1. Command notes

Expressions enclosed in square brackets, [ ], are optional and entered without the [or].

Expressions enclosed in greater than/less than, < >, are programming values and entered without the < or >.

The expression <SP> represents a one character ASCII Space.

In all commands upper case characters can be interchanged with lower case characters.

WORD	CAN BE REPLACED WITH
: AMPLITUDE	: AMPL
: CONDITION	: COND
: CURRENT	: CURR
: ENABLE	: ENAB
: ERROR?	: ERR?
: EVENT	: EVEN
: IMMEDIATE	: IMM
: LEVEL	: LEV
: LIMIT	: LIM
MEASURE	MEAS
MODE?	MOD?
: OPERATION	: OPER
OUTPUT	OUTP
: PRESET	: PRES
: PROTECTION	: PROT
: QUESTIONABLE	: QUES
SOURCE	SOUR
: STATE	: STAT
STATUS	STAT
SYSTEM	SYST
: TRIPPED	: TRIP
: VERSION	: VERS
: VOLTAGE	: VOLT

2-2. Programming and measurement commands

2-2-1. Program output voltage

The output voltage can be programmed by sending the command:

[SOURce]: VOLTage [:IMMediate][:LEVel][:AMPLitude]<SP><value>

Where <value> is any valid voltage with or without a decimal place.

Alternate Format:
SOURCE: VOLTAGE<SP><value>
: VOLTAGE: AMPLITUDE<SP><value>
: VOLTAGE<SP><value>
Examples:
SOURCE: VOLTAGE: AMPLITUDE 15.77
: VOLTAGE 3.25

2-2-2. Read programmed voltage command

To read what voltage the supply was programmed to, regardless of the actual voltage, send the following command and read the response message.

[SOURce]: VOLTage [: AMPLitude]?

Alternate Format:
SOURCE: VOLTAGE?
: VOLTAGE: AMPLITUDE?
: VOLTAGE?
Examples:
SOURCE: VOLTAGE: AMPLITUDE?
: VOLTAGE?

2-2-3. Measure voltage command

The output voltage can be measured by sending the command:

MEASure: VOLTage?

When the controller does the next GP-IB Read, the interface will send the measured voltage to it.

Example:
MEASURE: VOLTAGE?

2-2-4. Program output current command

The output current is programmed by sending the command:

[SOURce]: CURRent [:IMMediate][: LEVel][: AMPLitude]<SP><value>

Where < value > is any valid current with or without a decimal place.

Alternate Formats:
SOURCE: CURRENT<SP><value>
: CURRENT: AMPLITUDE<SP><value>
: CURRENT<SP><value>
Examples:
SOURCE: CURRENT: AMPLITUDE 15.77
: CURRENT 3.25

2-2-5. Read programmed current command

To read what current the supply was programmed to, regardless of the actual current, send the following command and read the response message.

[SOURce]: CURRent [: AMPLitude]?

Alternate Format:
SOURCE: CURRENT?
: CURRENT: AMPLITUDE?
: CURRENT?
Examples:
SOURCE: CURRENT: AMPLITUDE?
: CURRENT?

2-2-6. Measure current command

The output current can be measured by sending the SCPI command:

MEASure: CURRent?

When the controller does the next GP-IB Read, the supply will return the amperes of current being produced
Example:
MEASURE: CURRENT?

2-2-7. Enable the supply output command

The power supply output can be turned on by sending this command:

OUTPut: STATe<SP>1

The output will immediately jump to the last programmed voltage and current.
Example:
OUTPUT: STATE 1
Notes:
1 can be replaced with ON

2-2-8. Disable the supply output command

The power supply output can be shut off by sending this command:

OUTPut: STATe<SP>0

This command is equivalent to programming the output to zero volts.
Example:
OUTPUT: STATE 0
Notes:
0 can be replaced with OFF

2-2-9. Read output enable command

Reads the Power Supply output enable. Places a 1 in the output queue if the supply is enabled and a 0 if the supply is disabled.

Syntax:
OUTPut: STATe?
Example:
OUTPUT: STATE?

2-2-10. Go to local mode command

Places the supply under control of the Front Panel Controls.

Syntax:
SYSTem: SET<SP><0>
Example:
SYSTEM: SET 0
Notes:
0 can be replaced with LOC

2-2-11. Go to remote mode command

Places the supply under control of the GP-IB Interface.

Syntax:
SYSTem: SET<SP><1>
Example:
SYSTEM: SET 1
Notes:
1 can be replaced with REM

2-2-12. Go to remote with local lock out command

Places the supply under control of the GP-IB Interface and disables the Front Panel go to Local Button.

Syntax:
SYSTem: SET<SP><2>
Example:
SYSTEM: SET 2
Notes:
2 can be replaced with LLO

2-2-13. Read programming mode command

Reads the mode of the Power Supply. Place a 0 in the output queue if the supply is in Local Mode, a 1 if the supply is in Remote Mode and a 2 if the supply is in Remote Mode with Local Lock Out.

Syntax:
SYSTem: SET?
Example:
SYSTEM: SET?

2-3. Output protection commands

2-3-1. Over voltage protection

2-3-1-1. Set the over voltage protection level command

Set the over voltage protection level of the Power Supply to <value>, where <value> is a number between zero and the maximum supply output voltage. A decimal point is optional.

Syntax:
[SOURce]: VOLTage: PROTection: LEVel<SP><value>

Alternate Format:
: VOLTAGE: PROTECTION: LEVEL<SP><value>
Examples:
SOURCE: VOLTAGE: PROTECTION: LEVEL 25.00
SOURCE: VOLTAGE: PROTECTION: LEVEL MAX
Notes:
If <value> equals MAX, the supply will set its over voltage to its maximum level.

2-3-1-2. Read the over voltage protection level command

Read the over voltage protection level of the Power Supply.

Syntax:
[SOURce]: VOLTage: PROTection: LEVel?

Alternate Format
: VOLTAGE: PROTECTION: LEVEL?
Example:
SOURCE: VOLTAGE: PROTECTION: LEVEL?

2-3-1-3. Read over voltage tripped state command

Read if the Power Supply over voltage has tripped.

Syntax:
[SOURce]: VOLTage: PROTection: TRIPped?

Alternate Format:
: VOLTAGE: PROTECTION: TRIPPED?
Example:
SOURCE: VOLTAGE: PROTECTION: TRIPPED?

Read the response number. The normal response is "0" (zero). If a "1"(one) is returned, an over voltage has occurred and the output is shut down.

2-3-1-4. Set the under voltage protection level command

Set the under voltage protection level of the Power Supply.

Syntax:
[SOURce]: VOLTage: LIMit: LOW<SP><value>

Alternate Format:
: VOLTAGE: LIMIT: LOW<SP><value>
Example:
SOURCE: VOLTAGE: LIMIT: LOW 25.00

2-3-1-5. Read the under voltage protection level command

Read the under voltage protection level of the Power Supply.

Syntax:
[SOURce]: VOLTage: Limit: LOW?

Alternate Format
: VOLTAGE: LIMIT: LOW?
Example:
SOURCE: VOLTAGE: LIMIT: LOW?

2-3-2. Current fold back protection

2-3-2-1. Set current fold back protection command

Turn on the current fold back protection of the Power Supply.

Syntax:
[SOURce]: CURRent: PROTection: STATe<SP><1>

Alternate Format:
: CURRENT: PROTECTION: STATE<SP>1
Example:
SOURCE: CURRENT: PROTECTION: STATE 1

2-3-2-2. Clear current fold back protection command

Turn off the current fold back protection of the Power Supply.

Syntax:
[SOURce]: CURRent: PROTection: STATe<SP><0>

Alternate Format:
: CURRENT: PROTECTION: STATE<SP>0
Example:
SOURCE: CURRENT: PROTECTION: STATE 0

2-3-2-3. Read current fold back state command

Read if the Fold back is enabled or disabled. The interface will return an ON if Fold Back Protection is set or an OFF if not set.

Syntax:
[SOURce]: CURRent: PROTection: STATe?

Alternate Format:
: CURRENT: PROTECTION: STATE?
Example:
SOURCE: CURRENT: PROTECTION: STATE?

2-3-2-4. Read fold back tripped state command

Read if the Power Supply Current Fold Back has tripped. The interface will return a 1 if Fold Back Protection has tripped or a 0 if not tripped.

Syntax:
[SOURce]: CURRent: PROTection: TRIPped?

Alternate Format:
: CURRENT: PROTECTION: TRIPPED?
Example:
SOURCE: CURRENT: PROTECTION: TRIPPED?

2-4. Operating condition commands

2-4-1. System error enable command

Clears the Error Queue and enables all error messages to be placed in the Error Queue.

Syntax:
SYSTem: ERRor: ENABle

Action:
Direct the interface to save error messages.
Example:
SYSTEM: ERROR: ENABLE

2-4-2. Read system fault command

The oldest error message is removed from the Error Queue and placed in the Output Queue. If the Error Queue was empty, a 0 is placed in the Output Queue.

Syntax:
SYSTem: ERRor?
Example:
SYSTEM: ERROR?

2-4-3. Read supply output mode command

Read if the Power Supply is in the Constant Voltage, Constant Current or Output Off Mode. The interface will return a CV if the supply is in Constant Voltage Mode, a CC if the supply is in Constant Current Mode or an OFF if the supply output is off.

Syntax
SOURce: MODe?:
Example:
SOURCE: MODE?:

2-4-4. Set power supply power-up mode command

Set the Power Supply for Auto-Restart or Safe-Start operation upon power up.

Syntax:
OUTPut: PON<SP><value>
Where value = 0 for Safe-Start or value = 1 for Auto-Restart
Examples:
OUTPUT: PON 0,
OUTPUT: PON 1
Notes:
This command is in addition to the SCPI compliance requirements.

2-4-5. Report power supply power mode command

Report the Power Supply Auto-Restart or Safe-Start operation upon power up mode. The interface will return an ON if the supply is Auto-restart operation or an OFF if the supply is in Safe-Start operation

Syntax:
OUTPut: PON?
Example:
OUTPUT: PON?
Notes:
This command is in addition to the SCPI compliance requirements.

2-4-6. Read SCPI version command

Read the SCPI Compliance year that this interface adheres to.

Syntax:
SYSTem: VERSion?
Example:
SYSTEM: VERSION?

2-5. Common commands

2-5-1. Clears status command

Clears all event registers and stored error messages. Relays the command to the Power Supply

Syntax:
*CLS
Example:
*CLS

2-5-2. Set service request enable command

Set the Service Request Enable Register.

Syntax:
*SRE<SP><value>
Example:
*SRE 140
Notes:
<value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 255.

2-5-3. Read service request enable command

Read the value of the Service Request Enable Register.

Syntax:
*SRE?
Example:
*SRE?
Notes:
The returned <value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 255.

2-5-4. Read status byte command

Read the value of the Status Register. The Status Byte Register contains eight bits, which are set to show that some other register has recorded an event or an error. See Figure 3-1.
The response to this query will be a binary weighted number from 0 to 255.

Syntax:
*STB?
Example:
*STB?
Notes:
The returned <value> is a decimal number representing the sum of all the bits.
The range of <value> is 0 to 255.

2-5-5. Set standard event status ‘ENABLE’ register command

Set Event Status Enable Register to a value. See Figure 3-1.

Syntax:
*ESE<SP><value>
Example:
*ESE 74
Notes:
<value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 255.

2-5-6. Read standard event status ‘ENABLE’ register command

Read the value of the Event Status Enable Register. See Figure 3-1.

Syntax:
*ESE?
Example:
*ESE?
Notes:
The returned <value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 16,767.

2-5-7. Read standard event status ‘EVENT’ register command

Read the value of the Event Status Register. See Figure 3-1.

Syntax:
*ESR?
Example:
*ESR?
Notes:
The returned <value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 16,767.

2-5-8. Read identity command

Read Company Logo, Power Supply ranges, Serial Number and Revision of the Power Supply and Version of this GP-IB Interface. When the controller reads the output from the power supply, a single-line identity string will be returned. A typical identity string format is:
Manufacturer/Model <max volt>-<max curr>, S/N <supply serial>,
REV <power supply revision-GP-IB interface version>

Syntax:
*IDN?
Example:
*IDN?

2-5-9. Reset command

Resets the Power Supply

Syntax:
*RST
Example:
*RST

2-5-10. Self test query command

Test that the Interface and the Power Supply are operational. This will be accomplished by sending a measure voltage command to the power supply. The result of the measure voltage command is ignored. The result will be tested for completion and/or error.

Syntax:
*TST?
Example:
*TST?

2-5-11. Operation complete command

Set the Operation Complete Bit in the Standard Event Status Register when all operations have finished

Syntax:
*OPC
Example:
*OPC

2-5-12. Read operation complete command

Place a 1 in the Output Queue when all operations have finished.

Syntax:
*OPC?
Example:
*OPC?

2-5-13. Save power supply setting command

Sends a command to the Power Supply causing it to save its operating settings: Programmed voltage, Current, Over Voltage, Under Voltage, Remote/Local Mode, Auto/Safe Restart, Current Fold Back, etc. can be stored in Electrically Erasable Programmable read-only Memory (EEPROM). These values are the powerup default settings.
To change one or more settings, enter the one or more commands with new settings and then enter this command.

Syntax:
*SAV<SP><0>
Example:
*SAV 0

2-5-14. Recall power supply settings command

Sends a command to the Power Supply causing it to recall its operating settings:
Programmed voltage, Current, Over Voltage, Under Voltage, Remote/Local Mode, Auto/Safe Restart and Current Fold Back

Syntax:
*RCL<SP><0>
Example:
*RCL 0

2-6. Status commands

2-6-1. Read operational condition ‘EVENT’ register command

Reads the Operational Condition Event Register and puts the result in the Output Queue.

Syntax:
STATus: OPERation [: EVENt]?

Alternate Format:
STATUS: OPERATION?
Example:
STATUS: OPERATION: EVENT?
Notes:
The returned <value> is a decimal number representing the sum of all the event bits.
The range of <value> is 0 to 16,767.

2-6-2. Read operational condition ‘CONDITION’ register command

Reads the Operational Condition Register and puts the result in the Output Queue.

Syntax:
STATus: OPERation: CONDition?
Example:
STATUS: OPERATION: CONDITION?
Notes:
The returned <value> is a decimal number representing the sum of all the condition bits.
The range of <value> is 0 to 16,767.

2-6-3. Sset operational condition ‘ENABLE’ register command

Sets the Operational Condition Enable Register.

Syntax:
STATus: OPERation: ENABle<SP><value>
Example:
STATUS: OPERATION: ENABLE 53
Notes:
The <value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 16,767.

2-6-4. Read operational condition ‘ENABLE’ register command

Reads the Operational Condition Enable Register.

Syntax:
STATus: OPERation: ENABle?
Example:
STATUS: OPERATION: ENABLE?
Notes:
The <value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 16,767.

2-6-5. Read questionable condition ‘EVENT’ register command

Reads the Questionable Condition Event Register and puts the result in the Output Queue.

Syntax:
STATus: QUEStionable [: EVENt]?

Alternate Format:
STATUS: QUESTIONABLE?
Example:
STATUS: QUESTIONABLE: EVENT?
Notes:

The returned <value> is a decimal number representing the sum of all the event bits.
The range of <value> is 0 to 16,767.

2-6-6. Read questionable condition ‘CONDITION’ register command

Reads the Questionable Condition Register and puts the result in the Output Queue.

Syntax:
STATus: QUEStionable: CONDition?
Example:
STATUS: QUESTIONABLE: CONDITION?

2-6-7. Set questionable condition ‘ENABLE’ tegister command

Sets the Questionable Condition Enable Register.

Syntax:
STATus: QUEStionable: ENABle<SP><value>
Example:
STATUS: QUESTIONABLE: ENABLE 53
Notes:

The <value> is a decimal number representing the sum of all the enabled bits.
The range of <value> is 0 to 16,767.

2-6-8. Read questionable condition ‘ENABLE’ register command

Reads the Questionable Condition Enable Register.

Syntax:
STATus: QUEStionable: ENABle?
Example:
STATUS: QUESTIONABLE: ENABLE?

2-6-9. Status preset command

Presets all Operation Enable and Questionable Enable Registers.

Syntax:
STATus: PRESet
Example:
STATUS: PRESET

2-7. Using error and status registers

2-7-1. Overview: register fan-out

The GP-IB Interface board has a set of status and error registers. They are defined by the GP-IB-488.2 specification as part of the GP-IB-488.2 Common Command set required by all compliant instruments. These registers allow the GP-IB controller to examine the operational state of the supply in detail. A "fan-out" architecture is used so only one summary register needs to be read to know if an event occurred in any other register. This fan-out allows automatic test programs to efficiently manage the remote programming mode. A diagram of the register structure is shown in Figure 3-1. This diagram does not show all the registers in the GP-IB-488.2 specification. It only shows the registers typically used in the power supply.

2-7-2. Glossary of register terms

- SERVICE REQUEST: When an instrument on the GP-IB bus asserts the SRQ line in the cable, it tells the controller that it has completed its task or that an error has occurred.
- SERIAL POLL: An GP-IB function which reads back the data in an instrument's Status Byte Register. The controller should perform this function after every command to verify the command was successful.
- REGISTER QUERIES: Read the contents of registers. The contents are returned as a binary weighted decimal number.
- CONDITIONAL REGISTERS. These contain bits that are set when a condition or error occurs. The bits are only cleared when the condition or error is cleared. The contents may be read but not changed.
- ENABLE REGISTERS: The various Enable Registers can be set to allow the status and errors to be detectable by a Serial Poll.
- EVENT REGISTERS: These contain bits that are set when an event or error occurs. The bits are cleared when the contents of the register are queried.

2-7-3. Clear all status registers

*****CLS

This command clears all event registers and stored error messages. It will not affect the Conditional or the Enable registers.

2-7-4. Service requests and status byte register

The Status Byte Register contains the bits that are set when an event occurs in:

Questionable Condition Event Register,
Operational Condition Event Register,
Standard Event Status Register,
or when: a message is available in the Output Queue
or when: an Error Message is available in the Error Queue.

If any of these bits are set, and the corresponding bit is set in the Service Request Enable Register the Service Request Bite (SRQ) bit will become set.
The SRQ bit will assert a signal onto the Service Request (SRQ) line in the GP-IB cable. The controller program can detect the SRQ, read what the problem is from the power supply, and clear the SRQ.
The bit assignments for the Status Byte Register are:

Table 2-1 The Status Byte Register

BIT NUMBER	DECIMAL VALUE	BIT SYMBOL	DESCRIPTION
0	1		Bsuy-1=Busy, 0=Ready
1	2		Not Used
2	4	SYS	System Error. Message Available in Error Que.
3	8	QUE	Questionable Summary
4	16	MAV	Message Available in Output Que. Set after query message is received
5	32	ESB	Standard Event Summary
6	64	SRQ	Request For Service. Is set if SYS, QUE, MAV, ESB and/or OPR are enabled and set.
7	128	OPR	Operational Summary

2-7-4-1. The service request enable register

See the SET SERVICE REQUEST ENABLE COMMAND (*SRE). With two exceptions, the Service Request Enable Register is a mirror of the Status Byte Register. Bit 0 (Busy) and Bit 6 (SRQ) will be ignored. Also note that Bits 1 is not used in this interface and will have no effect.

To enable a Service Request, the user should refer to Table 1 and determine which events need to be enabled to cause the request, add up the decimal value for those events and supply that value to the *SRE command.

The power up value of the Service Request Enable Register is zero, which means no Service Requests are Enabled.

2-7-5. Standard event status ‘EVENT’ register

See the READ STANDARD EVENT STATUS ‘EVENT’ REGISTER COMMAND (*ESR?). The Standard Event Status Register has seven bits that indicate status and errors for the power supply and the interface.

The response message will be a binary weighted number from 0 to 255. Zero is returned if there are no errors or events. The contents of the Standard Event Status Register will be cleared to zeroes after the *ESR? Command is executed.

The bit assignments for this register are:

Table 2-2 The Standard Event Status Register

BIT NUMBER	DECIMAL VALUE	BIT SYMBOL	DESCRIPTION
0	1	OPC	Operation complete
1	2	0	Not Used
2	4	QYE	Query Error
3	8	DDE	Device Dependant Error
4	16	EXE	Execution Error
5	32	CME	Command Error
6	64	URQ	User Request
7	128	PON	Power On. Set when power is switched on.

2-7-5-1. The standard event status ‘ENABLE’ register

See the SET STANDARD EVENT STATUS ‘ENABLE’ REGISTER COMMAND (*ESE).
The STANDARD EVENT STATUS ‘ENABLE’ REGISTER is a mirror of the
STANDARD EVENT STATUS ‘EVENT’ REGISTER.

If any bit is set in the Standard Event Status Event Register and enabled in the Standard Event Status Enable Register, the event will propagate to the Status Byte Register as a Standard Event Summary. By writing a binary weighted value to the Standard Event Status Enable Register, the bits in the Standard Event Status Register may be individually enabled so only selected events will cause a service request.

The power-up default is all zeroes in the enable register. This means no status or errors will be sent to the Status Byte Register. However, even if no bits are enabled, the contents of the Standard Event Status Register may always be read with the *ESR? query.

The contents of the Standard Event Enable Register may be read by sending:

*ESE?

The response will be a bit weighted number whose bits correspond to Table 2

2-7-6. The operational registers

The Operational Registers are three 16-bit registers whose bits are not defined by the GP-IB specification but are specific to the GP-IB device.

The bit assignments for the Operational Registers are:

Table 2-3 The Operational Registers

BIT NUMBER	DECIMAL VALUE	BIT SYMBOL	DESCRIPTION
0	1	CV	Set high if Constant Voltage Operation
1	2	CC	Set high if Constant Current Operation
2	4	NFLT	No Fault
3	8	N/A	Not used
4	16	AST	Auto Start Enabled
5	64	FBE	Foldback Enable
6	64	LLO	Local Lockeout
7	128	PEM	Remote/Local Mode
8to15	N/A	N/A	Not used

2-7-6-1. The operational condition ‘CONDITION’ register

See the READ OPERATIONAL CONDITION ‘CONDITION’ REGISTER COMMAND (STATUS:OPERATION: CONDITION?) and Table 3. The bits, as listed in Table 3 reflect the conditions under which the power supply is operating.

2-7-6-2. The operational condition ‘ENABLE’ register

See the READ OPERATIONAL CONDITION ‘ENABLE’ REGISTER COMMAND (STATUS:OPERATION: ENABLE?), the SET OPERATIONAL CONDITION ‘ENABLE’ REGISTER COMMAND (STATUS:OPERATION: ENABLE) and Table 3.

The OPERATIONAL CONDITION ‘ENABLE’ REGISTER is a mirror of the OPERATIONAL CONDITION ‘CONDITION’ REGISTER.

If any bit is set in the Operational Condition ‘Condition’ Register and enabled in this register, the condition will propagate to the Operational Condition ‘Event’ Register as an event.

2-7-6-3. The operational condition ‘EVENT’ register

See the READ OPERATIONAL CONDITION ‘EVENT’ REGISTER COMMAND (STATUS:OPERATION: EVENT?) and Table 3.

The OPERATIONAL CONDITION ‘EVENT’ REGISTER is a mirror of the OPERATIONAL CONDITION ‘CONDITION’ REGISTER.

If any event is set in this Register, it will propagate to the Status Byte Register as an Operational Summary event.

2-7-7. The questionable condition registers

The Questionable Condition Registers are three 16-bit registers whose bits are not defined by the GP-IB specification but are specific to the GP-IB device.
The bit assignments for the Questionable Condition Registers are:

Table 2-4 The Questionable Registers

BIT NUMBER	DECIMAL VALUE	BIT SYMBOL	DESCRIPTION
0	1	N/A	Not used
1	2	AC	AC Fail
2	4	OTP	Over Temperature
3	8	FLD	Fold Back Protect
4	16	OVP	Over Voltage Protection
5	64	SO	Shut Off
6	64	OFF	Output Off
7	128	ENA	Output Enable
8	256	INPC	Input Overflow
9	512	INTO	Internal Overflow
10	1024	ITMC	Internal Time Out
11	2048	COMM	Internal Comm Error
12 to15	N/A	N/A	Not used

2-7-7-1. The questionable condition ‘CONDITION’ register

See the READ QUESTIONABLE CONDITION ‘CONDITION’ REGISTER COMMAND (STATUS: QUESTIONABLE: CONDITION?) and Table 4.　The bits, as listed in Table4 reflect the questionable error conditions under which the power supply is operating.

2-7-7-2. The questionable condition ‘ENABLE’ register

See the READ QUESTIONABLE CONDITION ‘ENABLE’ REGISTER COMMAND (STATUS: QUESTIONABLE: ENABLE?), the SET QUESTIONABLE CONDITION ‘ENABLE’ REGISTER COMMAND (STATUS: QUESTIONABLE: ENABLE) and Table 4.　The QUESTIONABLE CONDITION ‘ENABLE’ REGISTER is a mirror of the QUESTIONABLE CONDITION ‘CONDITION’ REGISTER.
If any bit is set in the Questionable Condition ‘Condition’ Register and enabled in this register, the condition will propagate to the Questionable Condition ‘Event’ Register as an event.

2-7-7-3. The questionable condition ‘EVENT’ register

See the QUESTIONABLE CONDITION ‘EVENT’ REGISTER COMMAND (STATUS: QUESTIONABLE: EVENT?) and Table 3.
The QUESTIONABLE CONDITION ‘EVENT ’ REGISTER is a mirror of the QUESTIONABLE CONDITION ‘CONDITION’ REGISTER.
If any event is set in this Register, it will propagate to the Status Byte Register as a Questionable Summary event.

3. SYSTEM: ERROR MESSAGES

The Status and Error Registers described in the previous section is only one of the status methods in the GP-IB board. There is also a SCPI requirement for error messages that are in the form of:

<Error Number><Comma><Quote><Error Description><Quote>

The user sends the "SYST: ERR?" query to read the error message. The messages are stored in a first-in/first-out queue. The SYST: ERR queue can buffer up to TEN error messages, although the tenth is replaced by the –350,"Queue Overflow" if an eleventh message is generated. After the queue overflow, only the first ten messages are stored and the later messages are lost.

The SYST: ERR queue is cleared by:

A. Reading the messages one at a time using "SYST: ERR?" until 0, "No error" is read.

B. The *CLS (Clear Status) command.

If any message is in the SYST: ERR queue (except "No error"), then bit 2 of the Status Byte is set. A Service Request is generated if enabled.

SYSTEM ERROR CODES

ERR No.	ERROR DESCRIPTION	ERROR EVENT	ERROR EXAMPLE
0	"No error"	No error reported	
-100	"Command error"	GP-IB receives command with unspecified error or Display micro returns C00 in response to command.	
-101	"Invalid Character"	A character was received that is not: a-z, A-Z, 0-9,?, *, :, ;, period, space, CR, LF	V%LT 50 VOLT, 50
-102	"Syntax error"	GP-IB receives unrecognized command word or Display micro returns "C01" in response to command.	BEAS: VOLT? VOLTS 150
-104	"Data type error"	GP-IB receives command parameter with wrong type of data or Display micro returns "C03" in response to command. Example: receives letter where number expected.	CURRENT NA OUTPUT DC
-109	"Missing parameter"	Valid command received but not enough parameters or Display micro returns "C02" in response to command.	VOLT
-112	"Program word too long"	Command word had more than 12 characters before separator (space or colon) was found	MEASUREVOLTAGE?
-350	"Queue Overflow"	Too many SYST: ERR messages are stored in this queue and the newest messages are discarded	
+300	"Execution error"	General execution error or Display micro returns E00 in response to command.	
+301	"PV above OVP"	Attempt to program voltage above OVP setting, so Display micro returned "E01" response.	
+302	"PV below UVL"	Attempt to program voltage below UVL setting, so Display micro returned "E02" response.	
+303	"OVP above rating"	Attempt to set OVP above max voltage rating, so Display micro returned "E03" response.	
+304	"OVP below PV"	Attempt to set OVP below voltage setting, so Display micro returned "E04" response.	
+305	"UVL below zero"	Attempt to set UVL below zero, so Display micro returned "E05" response.	
+306	"UVL above PV"	Attempt to set UVL above voltage setting, so Display micro returned "E06" response.	
+307	"On during fault"	Attempt to set supply output "ON" when a fault exists	
+320	"Fault shutdown"	General message for non-specified shutdown	
+321	"AC fault shutdown"	Brown-out or phase-loss shutdown occurred	
+322	"Over-Temperature shutdown"	Over-temperature shutdown occurred	
+323	"Fold-Back shutdown"	Fold-Back shutdown occurred	
+324	"Over-Voltage shutdown"	Over-Voltage shutdown occurred	
+325	"Analog shut-off "	Shut-Off occurred from rear panel J1	
+326	"Output-Off shutdown"	Output-Off occurred from front panel button	
+327	"Interlock Open shutdown"	Interlock Open occurred from rear panel J1	
+340	"Internal message fault"	General non-specified Internal message fault	
+341	"Input overflow"	GP-IB receive data buffer is too full. Receive buffer is 5 words, 12 chars each	
+342	"Internal overflow"	Serial receive buffer in GP-IB is full because Display micro sent too many characters	
+343	"Internal timeout"	GP-IB did not receive response from Display before timeout period	
+344	"Internal comm error"	GP-IB received bad response, or checksum error, from Display micro	
+344	"Internal checksum error"	Display micro detected checksum error and returned "C04" in response to command.	
+345	"Internal checksum error"	GP-IB detected checksum error in message from Display micro	
+3XX	"Unknown Error"	No known error reported by the Display Micro	

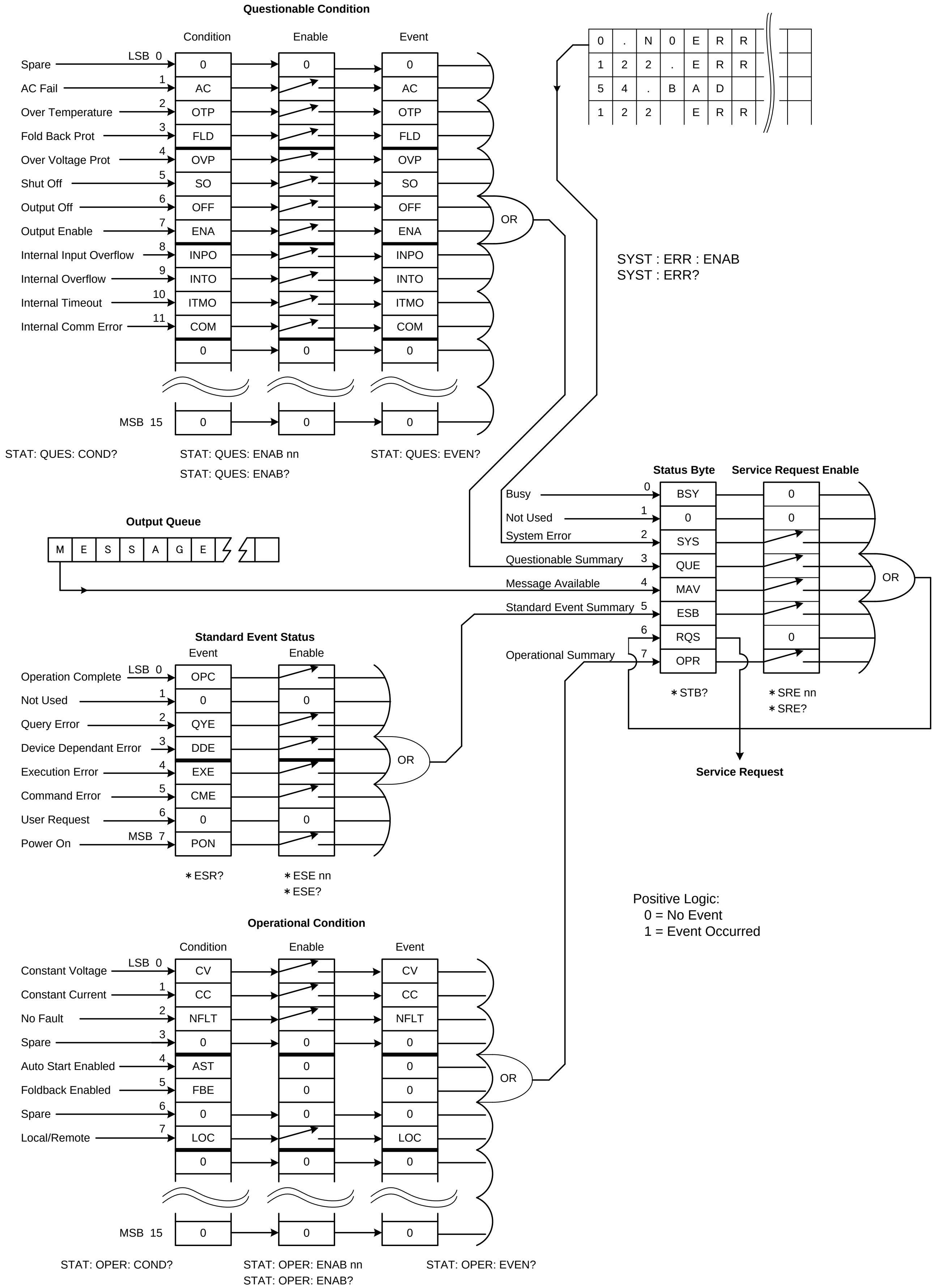


Fig 3-1 Status and Error Register Tree

Test and Measurement Solutions

TEXIO TECHNOLOGY CORPORATION

7F Towa Fudosan Shin Yokohama Bldg.

2-18-13, Shin Yokohama, Kohoku-ku,Yokohama, Kanagawa, 222-0033 Japan

http://www.texio.co.jp/