User Guide

HANDLER/PROBER DRIVER BUILDER

APAC STS COE

NATIONAL INSTRUMENTS

Revision History

Rev.	Date	Owner	Notes
1.0	18th, April, 2019	BAI Shuang	The DROP1 revision V1.0. **GPIB-Handler** types **ONLY**
2.0	5th, Nov., 2019	BAI Shuang	GPIB and TTL Handler Drivers

Introduction

Developing different kind of Handler/Prober Drivers takes a lot of effort on the online test project. Therefore, we developed a Handler/Prober Driver Builder tool to help engineers do this work, sharply improve the development efficiency and decrease the workload of engineers.

The chart below shows the typical effort and development time that an engineer would take to develop a handler driver on STS.

Items	Traditional	Using this tool
Read handler/prober datasheet	1 hour	1 hour
Know how the driver work in Teststand	8~24 hours	No need
Develop software and debug	40~80 hours	No need
Read user guide and configure the tool	No need	1~2 hours
Total effort	***TSM structure knowledge*** *and* ***coding skills*** *with* ***49~105 hours*** *engineer time*	***2~3 hours****, no need to know TSM and coding.*

Moreover, the tool will not change the structure, thus, if you already have a driver in work, you can still use it. The reusability and maintenance are much better using this tool. Users can find version number to verify the models and seek for help from our consultant team.

This user guide is going to guide engineers how to use the Handler/Prober Driver Builder tool to build up a handler/prober driver.

Note: This tool will cover 3 main categories of drivers: GPIB-Handler, TTL-Handler and GPIB-Prober. The current v2.0 version can only support GPIB-Handler and TTL-Handler. Further version will complete the left functions.

Target people of this User Guide:

This application is intended for those who are going to do the following work:

- Looking for a handler/prober driver for the current project
- Developing a handler/prober driver which is not exist yet.
- Modifying the codes of a handler/prober driver if this builder tool is not suitable.

Limitations:

- This application currently supports handler driver with limited sites number and bin range. For GPIB Handler, max 32 sites, for TTL Hander, max 4 sites. Detailed information refers to Chapter 2-1. Scope and Setup.
- The application can only generate basic communication of driver functions. More complex functions may not available using this tool. Once you would like to build a driver with extended functions that are not included in the document, please query APAC COE team for consultant.

Contents

Contents 3

1. Overview 4

2. Scope and Setup 5

2-1. Scope of this tool 5

2-2. Installation 5

2-3. How to use the driver while test? 6

Step1. Generate the driver 6

Step2. Find it in the Teststand configure station 6

Step3. Examine the parameters 7

Step4. Execute test 7

3. Quick Start 8

3-1 Configure GPIB Interface 8

3-2 Choose driver model 9

4. Configure GPIB-Handler Driver 11

4-1. Panel configuration and parameter introduction 11

The first part - Status Query 12

The second part - SOT & Sites Query 12

The third part - Binning 13

4-2. Instance analysis 16

Step1. Read the datasheet 16

Step2. Find key node and configure the key words 17

Step3. Examine and generate 22

Step4. Test 23

5. Configure TTL-Handler Driver 25

5-1.Panel configuration and parameter introduction 25

5-2. Instance analysis 26

Step1. Read the datasheet 26

Step2. Configure the panel 27

Step3. Examine and generate 28

Step4. Test 28

Reference 29

Additional Information 29

1. Overview

Handler/Prober Driver is a type of software installed on the tester. It communicates with handler/prober when executing test, helps handling the DUTs/wafers, presents BIN results information and repeats this procedure.

NI Teststand has integrated the API in TSM adds-on, which helps load the Handler/Prober Driver Entry Point sequences to build up a driver. However, the type of handler/prober varies on the market. There's no "One Driver" can cover all the types of handler/prober. Therefore, engineer used to write handler drivers by themselves. The development time may vary from several days to several weeks depends on the capacity of the developer and the complexity of the handler/prober. Moreover, because the code was specific to certain type of handler, even the same series of handlers may have different parameter requirements in different cases, so the reusage is unsatisfactory. In each new case, engineers may need to code the program from beginning. Thus, it cost more effort on the projects.

To solve the problem, we developed a Handler/Prober Driver Builder to help engineers configure the driver instead of coding, in which way it can sharply shorten the development time and lighten the workload of the engineers.

This guide is going to introduce the scope of this tool and the method of application. Follow the figure 1.1 to quick search what should you do when you are going to develop a new driver using this tool.

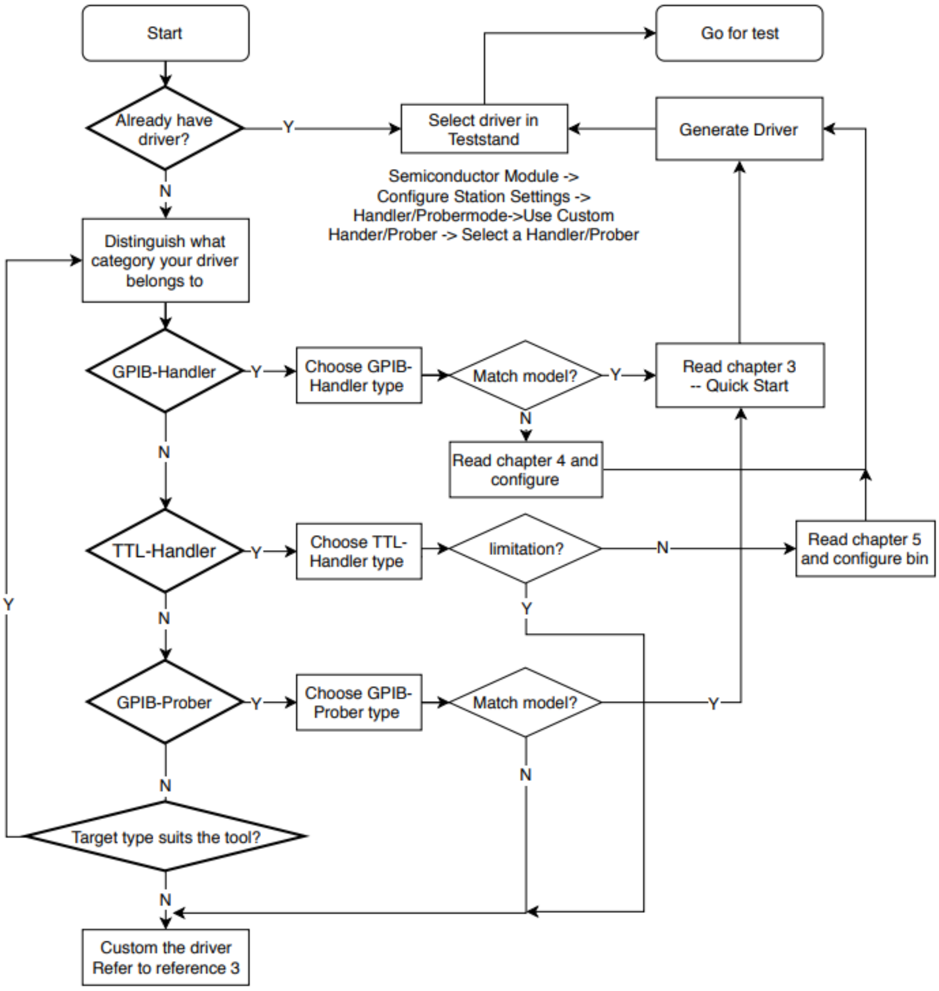

Start
Go for test
Already have driver?
Y
Select driver in Teststand
Generate Driver
N
Semiconductor Module -> Configure Station Settings -> Handler/Probermode->Use Custom Hander/Prober -> Select a Handler/Prober
Distinguish what category your driver belongs to
GPIB-Handler
Y
Choose GPIB-Handler type
Match model?
Y
Read chapter 3 -- Quick Start
N
Read chapter 4 and configure
N
TTL-Handler
Y
Choose TTL-Handler type
limitation?
N
Read chapter 5 and configure bin
Y
Y
N
GPIB-Prober
Y
Choose GPIB-Prober type
Match model?
Y
N
N
Target type suits the tool?
N
Custom the driver Refer to reference 3

Figure 1.1 Tool guidance flowchart

2. Scope and Setup

2-1. Scope of this tool

The handler/prober drivers can be divided into three categories: GPIB-Handler, TTL-Handler and GPIB-Prober. **This version (v2.0) can only support:**

GPIB-Handler with less than or equal to 32 sites under "Fullsite:xxxxxxxx" format. The corresponding bin range is from 0 to 15. If the site information is passed by SRQ message, it can only contain 4sites. The corresponding bin range is from 0 to 255.

TTL-Handler with less than or equal to 4 sites. It supports two bin categories – Binary and Parallel(also called Line). For Binary it supports bin range from 0~255. For Parallel it supports bin range from 0~15.

The tool is ideally designed for all GPIB-hander and TTL-Handler, although there could be mismatch. However, some known types of handler that this tool has verified are listed below:

GPIB-Handler name	**Simulation verification**	**Online verification**
MT93xx/95xx series	√	
Seiko EPSON NS-6000/7000/8000/9000 series	√	√
Hontech 904x/704x series	√	√
SRM_GPIB	√	√
ISMECA NY20	√	
esmo - talos	√	√
TTL-Handler name	**Simulation verification**	**Online verification**
MT93xx/95xx series	√	
Seiko EPSON NS-6000/7000/8000/9000 series	√	
Hontech 904x/704x series	√	
SRM_TTL	√	√
ISMECA_TTL	√	√

2-2. Installation

Double click the HandlerDiverBuilder.exe and use default address to automatically install the software. You can find the file in the teststand data folder: C:\Users\Public\Documents\National Instruments\TestStand 2017 (64-bit)\Components\Modules\NI_SemiconductorModule\HandlersAndProbers. After installation, there will be one folder named as "Standard VIs for HandlerProberDriverBuilder" and a sequence named by your configuration of the handler driver as well, with suffix .seq. The folder consists of configuration files and handler driver VIs. The sequence file is the entry point for specific handler driver used in teststand. Generally, you do not need to modify these files unless you have extra consideration.

Several most common GPIB Drivers has already been packed in the installer under the "NI-STD-GPIB-Handler Model File" folder, you might use them directly after you installed the tool.

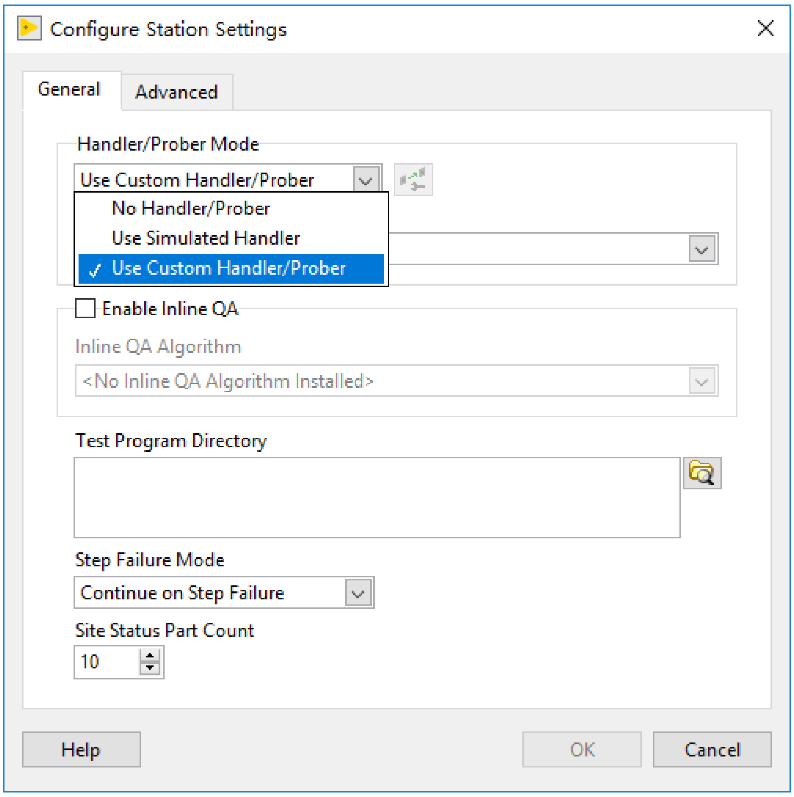
2-3. How to use the driver while test?

Step1. Generate the driver

Follow the instruction and generate the driver files for specific handler/prober. Refer to chapter 3,4 and 5 to see the details.

Step2. Find it in the Teststand configure station

Open Teststand, select “configure station” under “Semiconductor module” menu button if you have already installed TSM adds-on or STS bundle.


Configure Station Settings

Handler/Prober Mode: Use Custom Handler/Prober

- No Handler/Prober
- Use Simulated Handler
- ✓ Use Custom Handler/Prober

□ Enable Inline QA

Inline QA Algorithm: <No Inline QA Algorithm Installed>

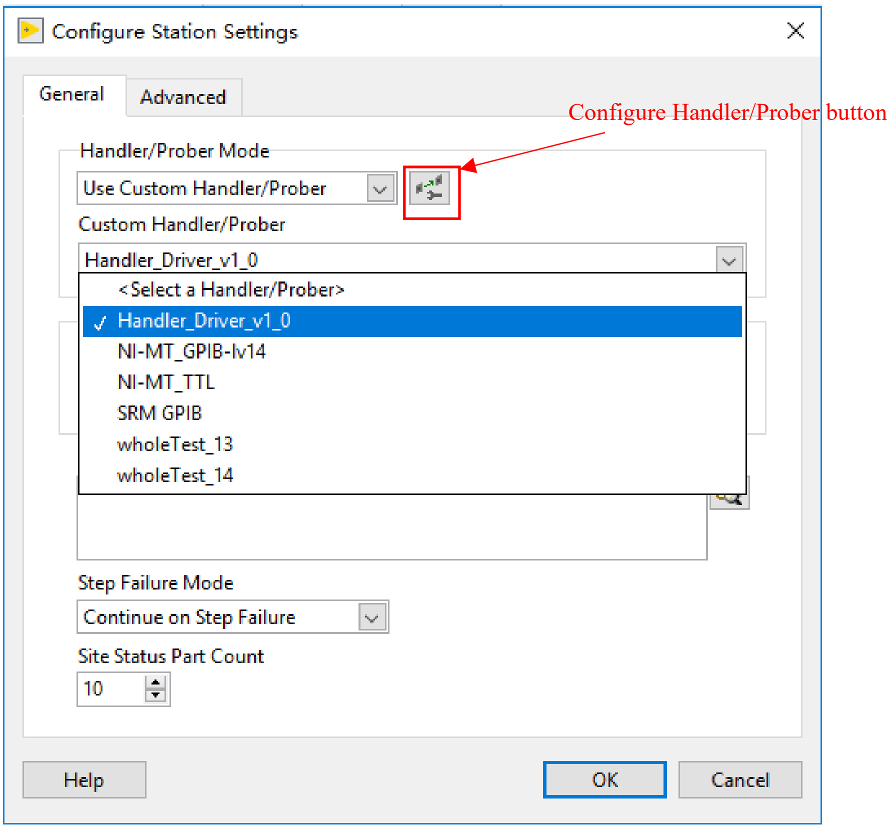
Test Program Directory:

Step Failure Mode: Continue on Step Failure

Site Status Part Count: 10

Figure 2.1 Configure Handler/Prober Mode

Choose “Use Custom Handler/Prober” under Handler/Prober Mode list in the General page. Then, choose the name of your handler/prober under the Custom Handler/Prober list. See figure 2.2.



Handler/Prober Mode: Use Custom Handler/Prober

Custom Handler/Prober: Handler_Driver_v1_0

- <Select a Handler/Prober>
- ✓ Handler_Driver_v1_0
- NI-MT_GPIB-lv14
- NI-MT_TTL
- SRM GPIB
- wholeTest_13
- wholeTest_14

Step Failure Mode: Continue on Step Failure

Site Status Part Count: 10

Figure 2.2 choose custom Handler/Prober

Step3. Examine the parameters

Users can click the "Configure Handler/Prober" button(shown in figure 2.2) at the right side of Handler/Prober Mode to check the parameters that will be used in test. Then, click ok to continue test.

Step4. Execute test

Click OK then run the test sequence file (your test program).

TTL Handler and other handlers may have different panel shown at the "Configure Handler/Prober", but the purpose is the same.

3. Quick Start

3-1 Configure GPIB Interface

YOU CAN IGNORE 3-1 IF YOU USE TTL-Handler

GPIB-Handler and GPIB-Prober users should configure GPIB in MAX first:

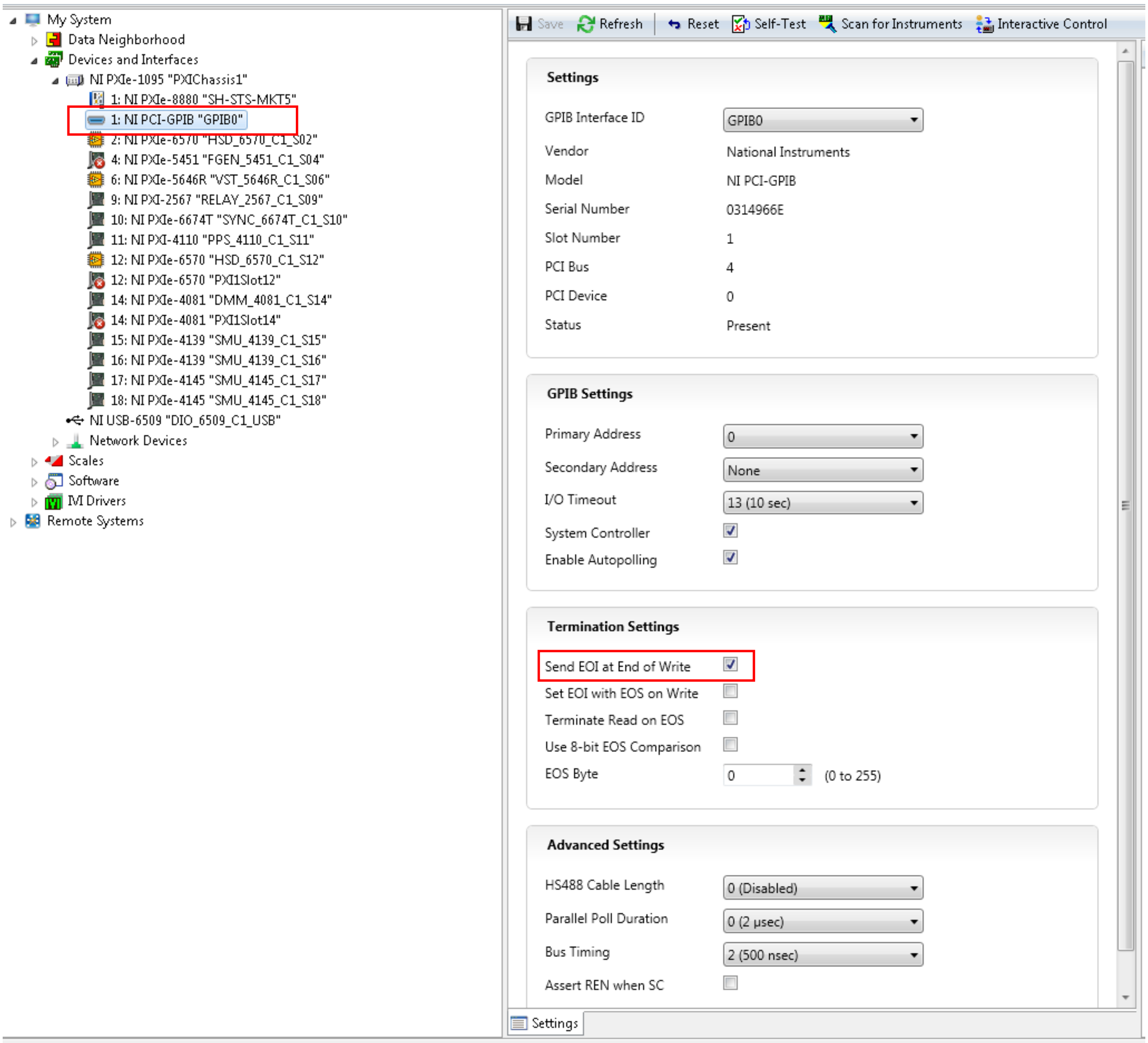
1.) Connect GPIB bus cable to handler.

2.) Then, set System Address TRUE and click Scan for Instruments. Other settings set default generally.

3.) Then, the handler should be found and listed under the GPIB device. You can see the address number of it, and that is the GPIB ID address. See figure 3.1.

4.) You can rename the instrument name or keep it as origin.

5.) Some handlers do not need EOI, you may uncheck the choice of Send EOI at End of Write.



- My System
  - Data Neighborhood
  - Devices and Interfaces
    - NI PXIe-1095 "PXIChassis1"
      - 1: NI PXIe-8880 "SH-STS-MKT5"
      - 1: NI PCI-GPIB "GPIB0"
      - 2: NI PXIe-6570 "HSD_6570_C1_S02"
      - 4: NI PXIe-5451 "FGEN_5451_C1_S04"
      - 6: NI PXIe-5646R "VST_5646R_C1_S06"
      - 9: NI PXI-2567 "RELAY_2567_C1_S09"
      - 10: NI PXIe-6674T "SYNC_6674T_C1_S10"
      - 11: NI PXI-4110 "PPS_4110_C1_S11"
      - 12: NI PXIe-6570 "HSD_6570_C1_S12"
      - 12: NI PXIe-6570 "PXI1Slot12"
      - 14: NI PXIe-4081 "DMM_4081_C1_S14"
      - 14: NI PXIe-4081 "PXI1Slot14"
      - 15: NI PXIe-4139 "SMU_4139_C1_S15"
      - 16: NI PXIe-4139 "SMU_4139_C1_S16"
      - 17: NI PXIe-4145 "SMU_4145_C1_S17"
      - 18: NI PXIe-4145 "SMU_4145_C1_S18"
    - NI USB-6509 "DIO_6509_C1_USB"
    - Network Devices
  - Scales
  - Software
  - IVI Drivers
- Remote Systems

Settings

GPIB Interface ID	GPIB0
Vendor	National Instruments
Model	NI PCI-GPIB
Serial Number	0314966E
Slot Number	1
PCI Bus	4
PCI Device	0
Status	Present

GPIB Settings

Primary Address	0
Secondary Address	None
I/O Timeout	13 (10 sec)
System Controller	☑
Enable Autopolling	☑

Termination Settings

Send EOI at End of Write	☑
Set EOI with EOS on Write	□
Terminate Read on EOS	□
Use 8-bit EOS Comparison	□
EOS Byte	0 (0 to 255)

Advanced Settings

HS488 Cable Length	0 (Disabled)
Parallel Poll Duration	0 (2 μsec)
Bus Timing	2 (500 nsec)
Assert REN when SC	□

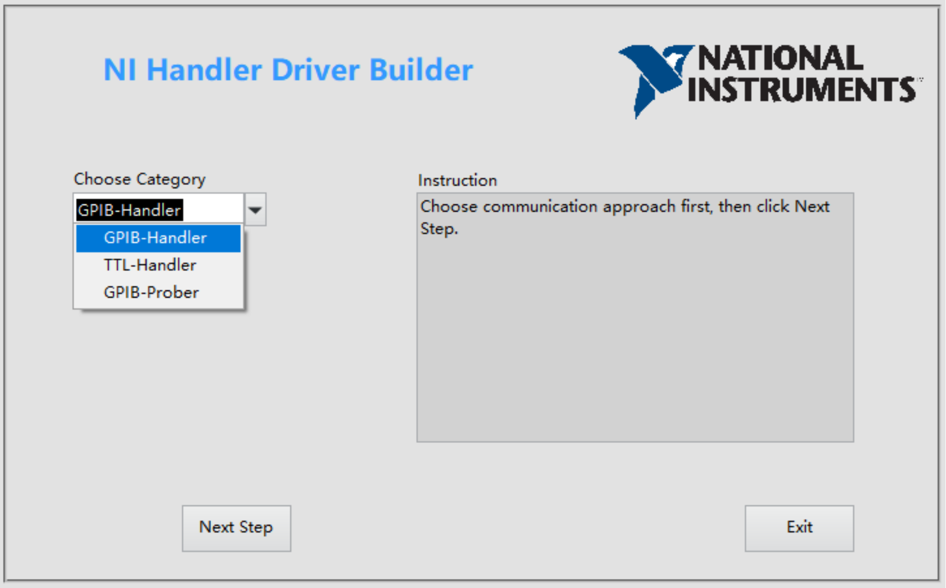
Figure 3.1 Configure GPIB device in MAX

Note: the instrument will appear under NI PCI-GPIB "GPIB0" after clicking "Scan for instruments"

3-2 Choose driver model

There are several models of existing handler/prober driver models installed with this tool. Users can choose the specific type.

Step1. Log in the HandlerDriverBuilder. Choose the handler/prober category, then, click Next Step.



Instruction

Choose communication approach first, then click Next Step.

Figure 3.2 Entry Point for choosing handler/prober category

Step2.

A.) If you chose GPIB-Handler:

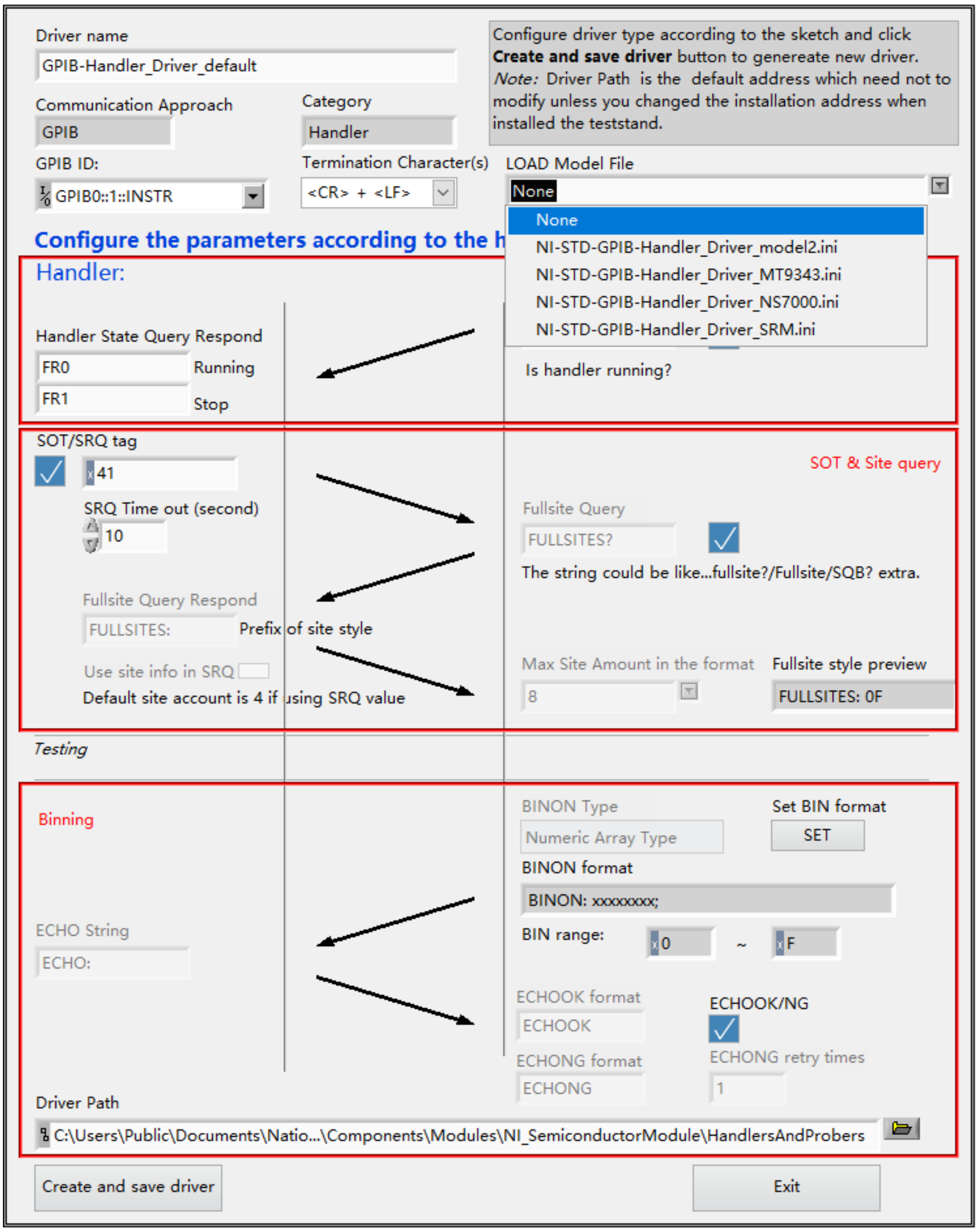
Click Next Step and a panel will pop out. See figure 3.3. If your target is among these types,

1.) Choose the one and rename as you wish.

2.) Choose the correct GPIB channel, under GPIB ID.

3.) Then, click “Create and save driver” button to generate the driver.

4.) A standard driver has been generated and you can use it in Teststand.


LOAD Model File
None
NI-STD-GPIB-Handler_Driver_model2.ini
NI-STD-GPIB-Handler_Driver_MT9343.ini
NI-STD-GPIB-Handler_Driver_NS7000.ini
NI-STD-GPIB-Handler_Driver_SRM.ini
SET
Create and save driver
Exit

Driver name: GPIB-Handler_Driver_default

Configure driver type according to the sketch and click **Create and save driver** button to genereate new driver. *Note:* Driver Path is the default address which need not to modify unless you changed the installation address when installed the teststand.

Communication Approach: GPIB

Category: Handler

GPIB ID: GPIB0::1::INSTR

Termination Character(s): <CR> + <LF>

LOAD Model File: None

Configure the parameters according to the h

Handler:

Handler State Query Respond: FR0 Running; FR1 Stop

Is handler running?

SOT/SRQ tag ☑ 41

SRQ Time out (second): 10

SOT & Site query

Fullsite Query: FULLSITES? ☑

The string could be like...fullsite?/Fullsite/SQB? extra.

Fullsite Query Respond: FULLSITES: Prefix of site style

Use site info in SRQ □

Default site account is 4 if using SRQ value

Max Site Amount in the format: 8

Fullsite style preview: FULLSITES: 0F

Testing

Binning

BINON Type: Numeric Array Type

Set BIN format

BINON format: BINON: xxxxxxxx;

BIN range: 0 ~ F

ECHO String: ECHO:

ECHOOK format: ECHOOK

ECHOOK/NG ☑

ECHONG format: ECHONG

ECHONG retry times: 1

Driver Path: C:\Users\Public\Documents\Natio...\Components\Modules\NI_SemiconductorModule\HandlersAndProbers

Figure 3.3 Load model file

What's more, users can configure their own driver based on these models. If your driver can be replaced by or the format is like the existing drivers in the list, you can choose one model and modify the parameters as required. Rename it and generate as a new driver. Refer to Chapter 4. Configure GPIB-Handler Driver for detailed instruction.

B.) If you chose TTL-Handler:

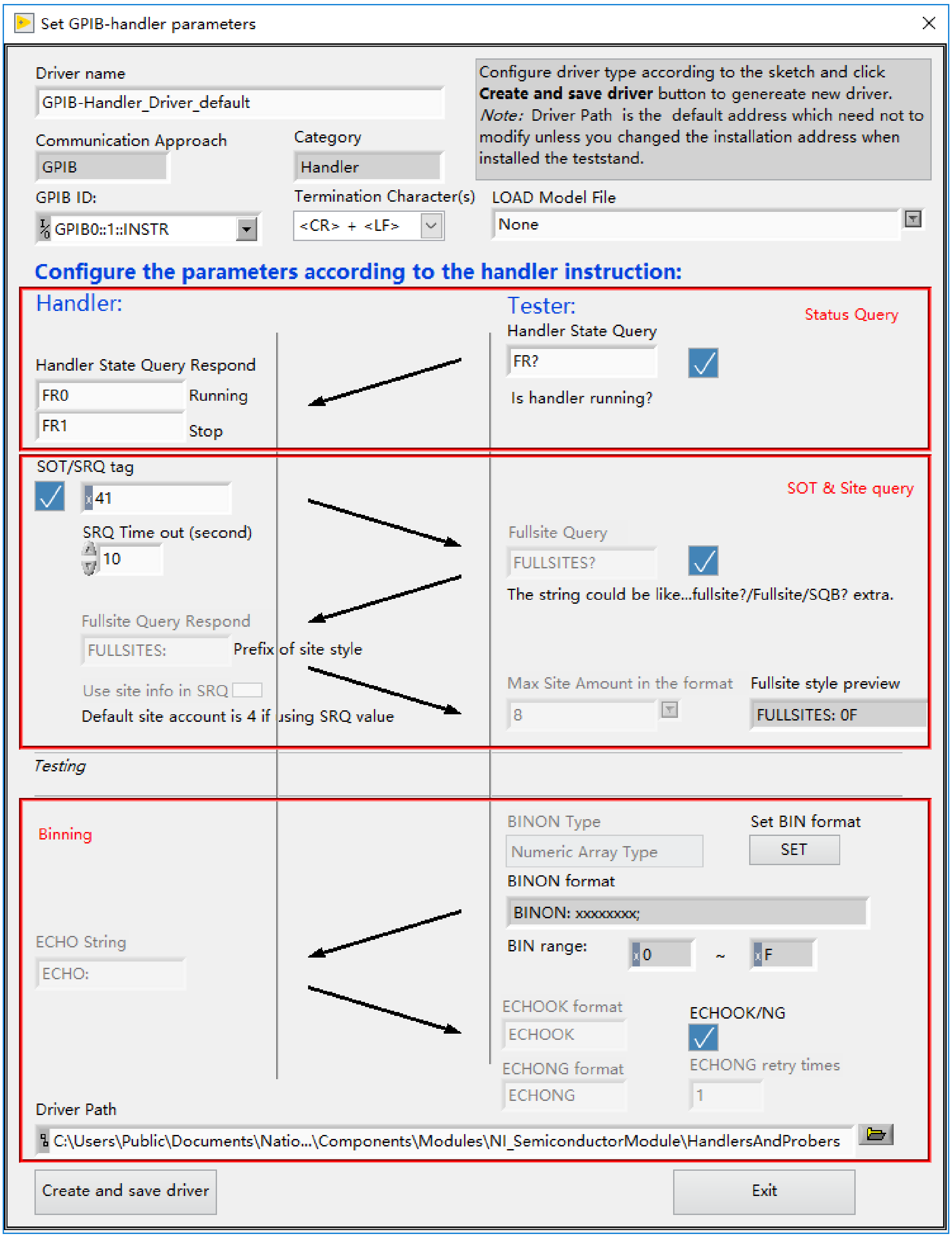
Click Next Step and a panel will pop out. Please refer to Chapter 5. Configure TTL-Handler Driver. The panel is the same. If you do not want to configure it here, you can click the "Configure Handler/Prober" button and configure the parameters later in the pop out panel as well.

4. Configure GPIB-Handler Driver

4-1. Panel configuration and parameter introduction

If you do not need to load any existing models, or, if you have more complex configurations than just models, choose “None” and configure from the default values.

Open HandlerDiverBuilder.exe then the panel will pop out. Engineers can configure the parameters according to the datasheet of the target handler driver. Figure 4.1 shows the configuration panel.



Set GPIB-handler parameters

Driver name: GPIB-Handler_Driver_default

Communication Approach: GPIB

Category: Handler

GPIB ID: GPIB0::1::INSTR

Termination Character(s): <CR> + <LF>

LOAD Model File: None

Configure driver type according to the sketch and click **Create and save driver** button to genereate new driver. *Note:* Driver Path is the default address which need not to modify unless you changed the installation address when installed the teststand.

Configure the parameters according to the handler instruction:

Handler:

Tester:

Status Query

Handler State Query: FR? ✓

Is handler running?

Handler State Query Respond

FR0 Running

FR1 Stop

SOT/SRQ tag ✓ 41

SRQ Time out (second): 10

SOT & Site query

Fullsite Query: FULLSITES? ✓

The string could be like...fullsite?/Fullsite/SQB? extra.

Fullsite Query Respond: FULLSITES: Prefix of site style

Use site info in SRQ □

Default site account is 4 if using SRQ value

Max Site Amount in the format: 8

Fullsite style preview: FULLSITES: 0F

Testing

Binning

BINON Type: Numeric Array Type

Set BIN format: SET

BINON format: BINON: xxxxxxxx;

BIN range: 0 ~ F

ECHO String: ECHO:

ECHOOK format: ECHOOK

ECHOOK/NG ✓

ECHONG format: ECHONG

ECHONG retry times: 1

Driver Path: C:\Users\Public\Documents\Natio...\Components\Modules\NI_SemiconductorModule\HandlersAndProbers

Figure 4.1 configuration panel

On the top of the panel, it shows some overall information:

Driver name: Name your driver. It should not be the same in the target folder, otherwise the tool may mention you whether to replace it.

Communication approach: GPIB. Not configurable.

Category: Handler. Not configurable.

GPIB ID: Choose or type the GPIB address. Make sure it is the name of the handler that your driver will communicate with.

Termination character(s): The termination char of the string replies to the handler.

LOAD Model File: Choose to load the model configuration installed with the tool.

Then, it shows the configuration main body. The process of handler driver consists of three parts -- **Status Query**, **SOT & Sites Query** and **Binning**. The dialog between handler and tester is separated using arrows, which indicates the speaker and the listener. Following the arrow process can build up a specific handler driver that suits the target handler. The details are described as below:

The first part - Status Query

This part is optional. It is a handshake event that some handlers have but some may not have. If your driver does not need this part, uncheck the tick at right side, then the whole part will become gray which means disabled.

Handler State Query: Configure the string that sent from the tester to the handler querying running status.

Handler State Query Respond: Configure the expected respond string. **Run** indicates the response string that the handler will reply to tester if it is running and is ready for tests. **No Run** indicates the response string that the handler will reply to tester if the handler is not ready for tests.

The second part - SOT & Sites Query

In this part, handler sends SOT signal and passes the sites information to the tester. Tester send query information to handler (in some case) using the configured string. The parameters are shown below:

SOT/SRQ tag: Using SRQ message to send “Start of Test” signal.

SRQ Time out (second): Set SRQ time out by second. Over time will terminate the test.

Fullsite Query: Set the site query string in the configurable bar according to the datasheet.

Fullsite Query Respond: Set the site query respond prefix string. The site format is fixed, presented as several 4bit hex numbers, each bit represents one site status from lower bit. Bit value 1 means enabled whereas 0 means disabled. For example, 0x05 represent for 0b00000101, means that site 1 and site 3 is enabled. Here, if we fill “FULLSITE:” and set max site account 8, the message sent could be “FULLSITE:05”, depends on the real site status, which tells the tester site1 and site 3 are enabled. If there is no prefix string, leave it empty.

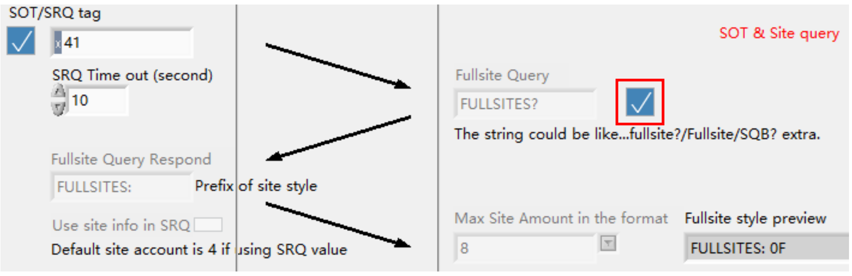
Use site info in SRQ: Not configurable. Available when **Fullsite Query Respond** is unavailable.

Max Site Amount in the format: Set the maximum number of sites, which is the required presenting sites in the datasheet. This parameter may vary from different types of binning. Numeric array type supports 32 sites, 16sites, 8sites, and 4 sites while String type supports only 8sites and 4 sites. The details see **The third part - - Binning.**

In general, handler sends SOT signal by SRQ (*Service Request Query, a GPIB function, which is like interrupt line. More information please refer to GPIB help documents*) message first, then it waits for tester sending fullsite query message. When received fullsite query message, handler sends site information back.

However, some handler may not use SRQ to transfer SOT signal, or, there's even no such SOT message, **Fullsite Query Respond** may be regarded as SOT. In this case, the **SOT/SRQ tag** should be unchecked as disabled, which means do not use SRQ message as SOT.

Some handlers use SRQ message for both SOT and site information. The site information is contained in the SRQ message, for example, SRQ 0x41(0b01000001) represent for SOT(higher 4 bit 0100) and site information (lower 4bit 0001 represent for site1 enable) . The **Fullsite Query** and **Fullsite Query Respond** are not needed. Under this circumstance, you should uncheck the **Fullsite Query** tick to disable the two part, see figure 4.2.

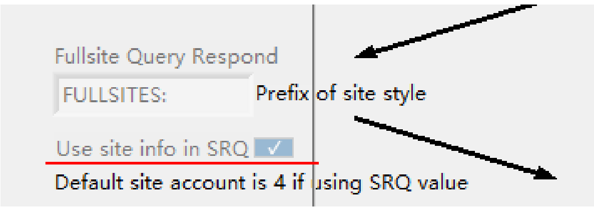


SOT/SRQ tag
☑ 41
SRQ Time out (second)
10
Fullsite Query Respond
FULLSITES: Prefix of site style
Use site info in SRQ □
Default site account is 4 if using SRQ value

SOT & Site query
Fullsite Query
FULLSITES? ☑
The string could be like...fullsite?/Fullsite/SQB? extra.
Max Site Amount in the format
8
Fullsite style preview
FULLSITES: 0F

Figure 4.2 The tick of **Fullsite Query** and **Fullsite Query Respond**

If the **Fullsite Query** is unchecked, the **Use site info in SRQ** is checked automatically. If so, the lower 4 bits of SRQ message indicates 4 sites, bit 0 indicates site1, bit 1indicates site2, bit 2 indicates site3, bit 3 indicates site4. See figure 4.3.



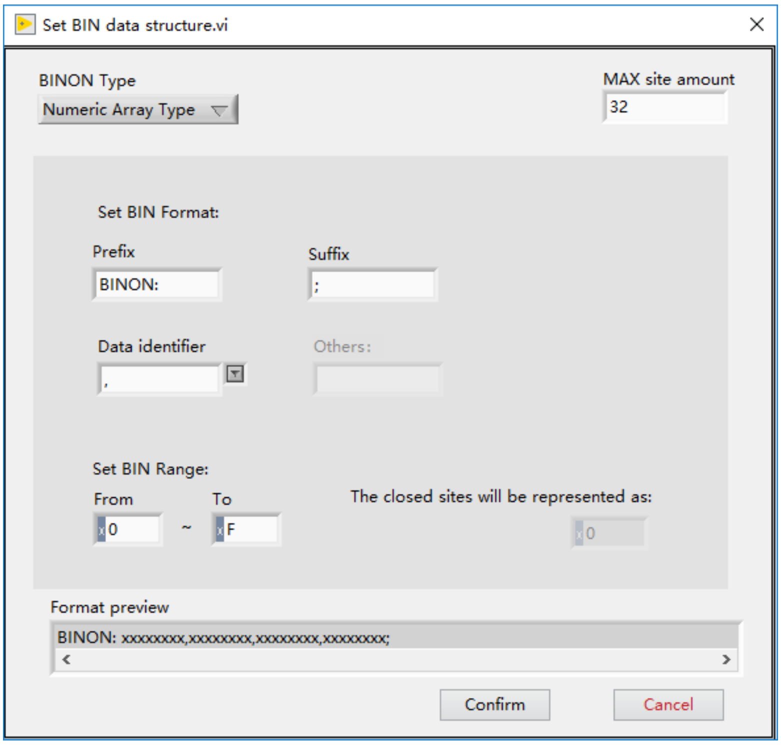
Fullsite Query Respond
FULLSITES: Prefix of site style
Use site info in SRQ ☑
Default site account is 4 if using SRQ value

Figure 4.3 **Use site info in SRQ** enabled when **Fullsite Query Respond** is disabled

The third part - Binning

In this part, tester returns the test binning result to handler. User should set one of the two types of binning format - **Numeric Array Type** and **String Type**.

1). Click **Set BIN format** button, then a window pops out, choose BINON Type first. If your binning format is like "BINON: xxxxxxxx,xxxxxxxx", where "x" represents for the bin number, use **Numeric Array Type**. The configuration panel is shown in figure 4.4.



BINON Type: Numeric Array Type

MAX site amount: 32

Set BIN Format:

Prefix: BINON:

Suffix: ;

Data identifier: ,

Others:

Set BIN Range:

From x0 ~ To xF

The closed sites will be represented as: x0

Format preview: BINON: xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx;

Figure 4.4 Set binning format as **Numeric Array Type**

Numeric array type supports 32 sites, 16sites, 8sites, and 4 sites. The MAX site amount should be chosen at the previous panel. It is read-only here.

Prefix: Set prefix string of the binning message.

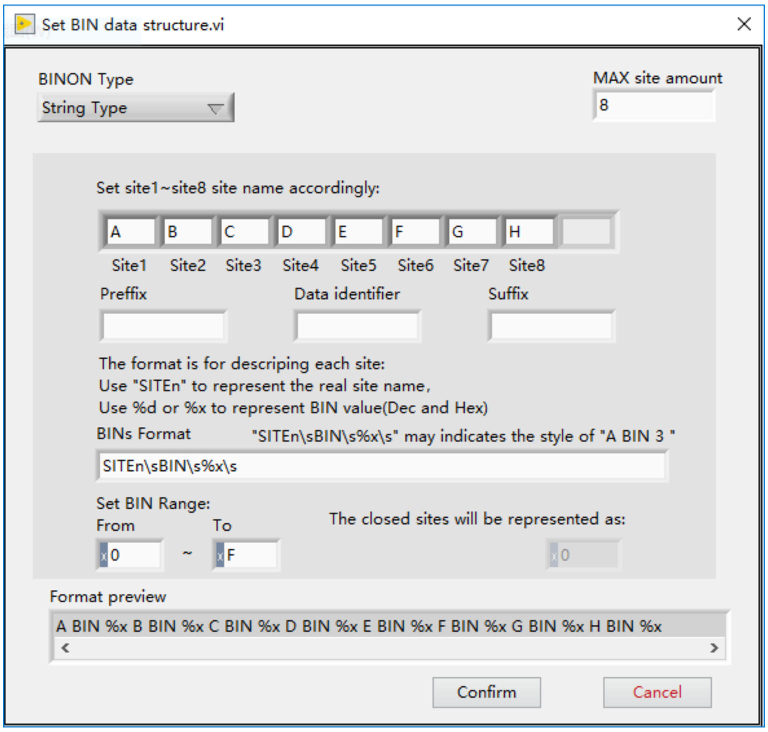
Suffix: Set suffix string of the binning message. If no suffix, leave it empty.

Data identifier: Set data identifier of the binning message. It can be set as ".", ",", ";", "\s" (space) or "others". If you choose others, fill the chars in the bar "Others" on the right. It only separates the 8 bins group at the intervals.

Set BIN Range: Set continuously bin range from 0x00 to 0x0F (max). Bin number bigger than the max will be regulated to the max number. Bin number smaller than the min number will be regulated to the min number. The closed sites generally return -1, and will be regulated to 0 as default, which is not configurable.

Format preview: It shows the represented format of what users set.

2.) If your binning format is like "A BIN 2 B BIN 5", bin number follows the defined site name as a string group, then the group links one by one, use **String Type**.



Set BIN data structure.vi

BINON Type: String Type

MAX site amount: 8

Set site1~site8 site name accordingly:

Site1	Site2	Site3	Site4	Site5	Site6	Site7	Site8
A	B	C	D	E	F	G	H

Preffix:

Data identifier:

Suffix:

The format is for descriping each site:
Use "SITEn" to represent the real site name,
Use %d or %x to represent BIN value(Dec and Hex)

BINs Format "SITEn\sBIN\s%x\s" may indicates the style of "A BIN 3 "

SITEn\sBIN\s%x\s

Set BIN Range:
From x0 ~ To xF

The closed sites will be represented as: x0

Format preview

A BIN %x B BIN %x C BIN %x D BIN %x E BIN %x F BIN %x G BIN %x H BIN %x

Figure 4.5 Set binning format as **String Type**

String type supports only 8sites or 4 sites as MAX site amount. The MAX site amount should be chosen at the previous panel. It is read-only here. The closed site will return as -1, the corresponding site bin group will not be sent in the string.

Set site1-site8 site name accordingly: Name each site from site 1 to site 8 as the datasheet defines. It can be “A” “B” or “1” “2” etc.

Prefix: Set prefix string of the binning message.

Data identifier: Set data identifier as you need to separate the bin format groups.

Suffix: Set suffix string of the binning message.

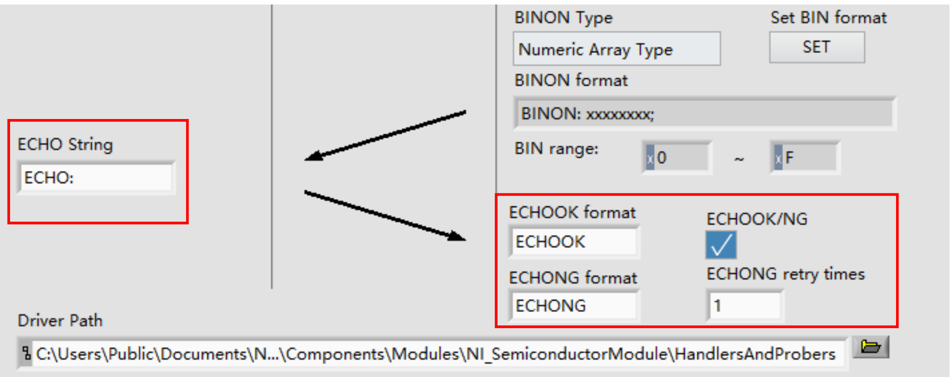
BINs Format: Set the bin group style. Use "SITEn" to represent the real site name. Use %d or %x to represent BIN value (Dec or Hex). For example, "SITEn\sBIN\s%x\s" may indicates the style of "A BIN 3 ", SITEn is replaced by A, %x is replaced by 3, the bin value.

Set BIN Range: Set continuously bin range from 0x00 to 0xFF (max). Bin number bigger than the max will be regulated to the max number. Bin number smaller than the min number will be regulated to the min number. The closed sites generally return -1, and will be regulated to 0 as default, which is not configurable.

Format preview: It shows the represented format of what users set.

Once finished the configuration, click **Confirm** to save the changes. Otherwise click **Cancel**, then it will not save the changes. When you finished configuring this panel, it will close and show the changes at **BINON format** on the main panel.

Some handlers have Echo part to echo the bin information to make sure the communication. check ECHOOK/NG tick to enable or disable this function.


BINON Type: Numeric Array Type

Set BIN format: SET

BINON format: BINON: xxxxxxxx;

BIN range: 0 ~ F

ECHO String: ECHO:

ECHOOK format: ECHOOK

ECHOOK/NG: ☑

ECHONG format: ECHONG

ECHONG retry times: 1

Driver Path: C:\Users\Public\Documents\N...\Components\Modules\NI_SemiconductorModule\HandlersAndProbers

Figure 4.6 ECHO settings

ECHO String: Set the prefix string of ECHO string. The bin information follows this prefix.

ECHOOK format: Set reply string from tester if ECHO message is correct. The binning will be over.

ECHONG format: Set reply string from tester if ECHO message is incorrect. Tester will send bin information one more time.

ECHONG retry times: Set retry times of ECHO. Hander will send echo message back until it is correct or up to this retry times.

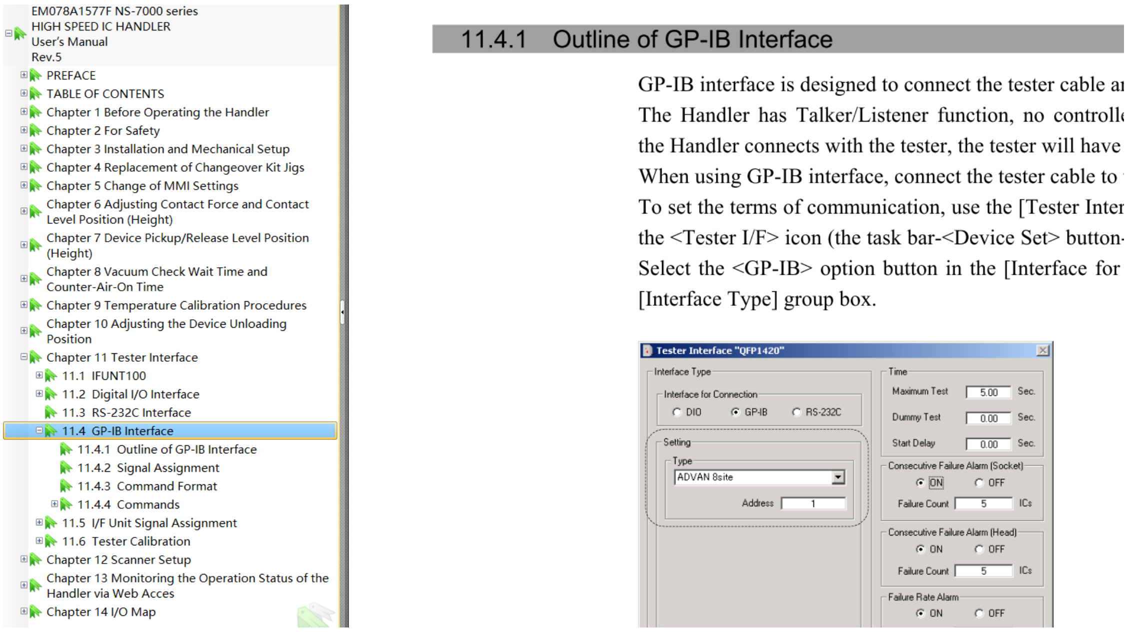
The **Driver Path** is the target file folder that these configuration files will be saved to by default. You should not change it if you did not change the default teststand configuration.

4-2. Instance analysis

Use Seiko EPSON NS-7000 handler as an example to help explain how to configure the driver. Generally, we use 4 steps to complete the process.

Step1. Read the datasheet

Once you got a datasheet of a handler, the information of what we need to builder a handler driver is usually in INTERFACE chapter. See figure 4.7.


EM078A1577F NS-7000 series
HIGH SPEED IC HANDLER
User's Manual
Rev.5

- PREFACE
- TABLE OF CONTENTS
- Chapter 1 Before Operating the Handler
- Chapter 2 For Safety
- Chapter 3 Installation and Mechanical Setup
- Chapter 4 Replacement of Changeover Kit Jigs
- Chapter 5 Change of MMI Settings
- Chapter 6 Adjusting Contact Force and Contact Level Position (Height)
- Chapter 7 Device Pickup/Release Level Position (Height)
- Chapter 8 Vacuum Check Wait Time and Counter-Air-On Time
- Chapter 9 Temperature Calibration Procedures
- Chapter 10 Adjusting the Device Unloading Position
- Chapter 11 Tester Interface
  - 11.1 IFUNT100
  - 11.2 Digital I/O Interface
  - 11.3 RS-232C Interface
  - 11.4 GP-IB Interface
    - 11.4.1 Outline of GP-IB Interface
    - 11.4.2 Signal Assignment
    - 11.4.3 Command Format
    - 11.4.4 Commands
  - 11.5 I/F Unit Signal Assignment
  - 11.6 Tester Calibration
- Chapter 12 Scanner Setup
- Chapter 13 Monitoring the Operation Status of the Handler via Web Acces
- Chapter 14 I/O Map

11.4.1 Outline of GP-IB Interface

GP-IB interface is designed to connect the tester cable ar
The Handler has Talker/Listener function, no controll
the Handler connects with the tester, the tester will have
When using GP-IB interface, connect the tester cable to
To set the terms of communication, use the [Tester Inter
the <Tester I/F> icon (the task bar-<Device Set> button
Select the <GP-IB> option button in the [Interface for
[Interface Type] group box.

Tester Interface "QFP1420"

Interface Type
Interface for Connection: ○ DIO ● GP-IB ○ RS-232C
Setting
Type: ADVAN 8site
Address: 1

Time
Maximum Test: 5.00 Sec.
Dummy Test: 0.00 Sec.
Start Delay: 0.00 Sec.

Consecutive Failure Alarm (Socket): ● ON ○ OFF
Failure Count: 5 ICs

Consecutive Failure Alarm (Head): ● ON ○ OFF
Failure Count: 5 ICs

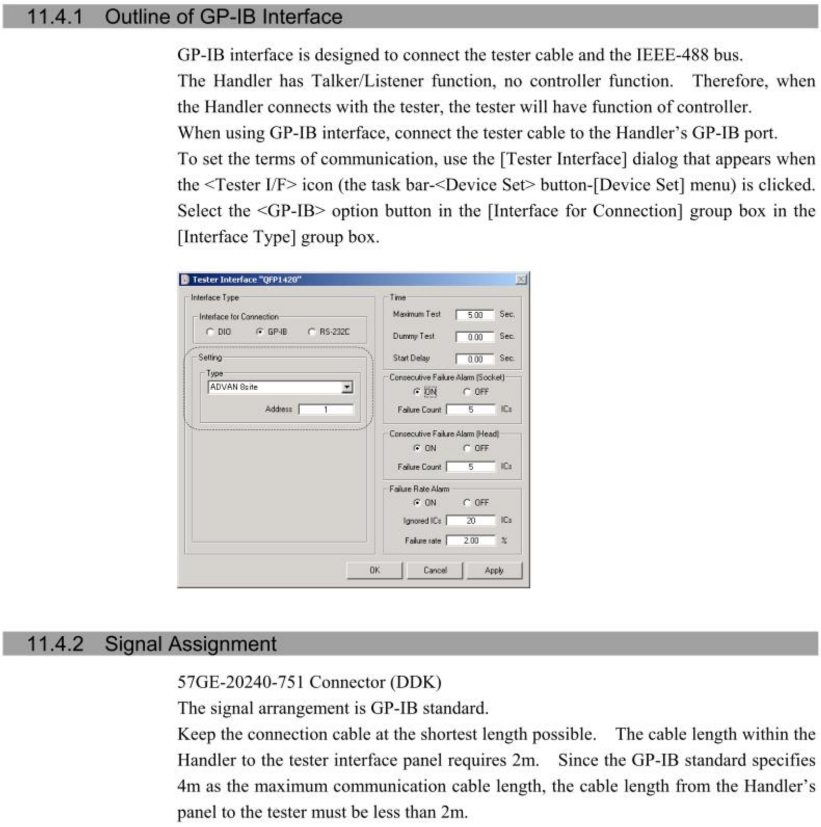
Failure Rate Alarm: ● ON ○ OFF

Figure 4.7 Bookmark of NS-7000 User's Manual

Some handlers have more than one method of handler communication (the most common ones are GPIB, Digital I/O and RS232). Select the section you need. Here we will read the GPIB Interface part.

Step2. Find key node and configure the key words

Here shows some key information and we will see that how to configure out a corresponding handler driver using these descriptions.



11.4.1 Outline of GP-IB Interface

GP-IB interface is designed to connect the tester cable and the IEEE-488 bus.
The Handler has Talker/Listener function, no controller function. Therefore, when the Handler connects with the tester, the tester will have function of controller.
When using GP-IB interface, connect the tester cable to the Handler's GP-IB port.
To set the terms of communication, use the [Tester Interface] dialog that appears when the <Tester I/F> icon (the task bar-<Device Set> button-[Device Set] menu) is clicked.
Select the <GP-IB> option button in the [Interface for Connection] group box in the [Interface Type] group box.

Tester Interface "QFP1420"

Interface Type
Interface for Connection: ○ DIO ● GP-IB ○ RS-232C
Setting
Type: ADVAN 8site
Address: 1

Time
Maximum Test: 5.00 Sec.
Dummy Test: 0.00 Sec.
Start Delay: 0.00 Sec.

Consecutive Failure Alarm (Socket): ● ON ○ OFF
Failure Count: 5 ICs

Consecutive Failure Alarm (Head): ● ON ○ OFF
Failure Count: 5 ICs

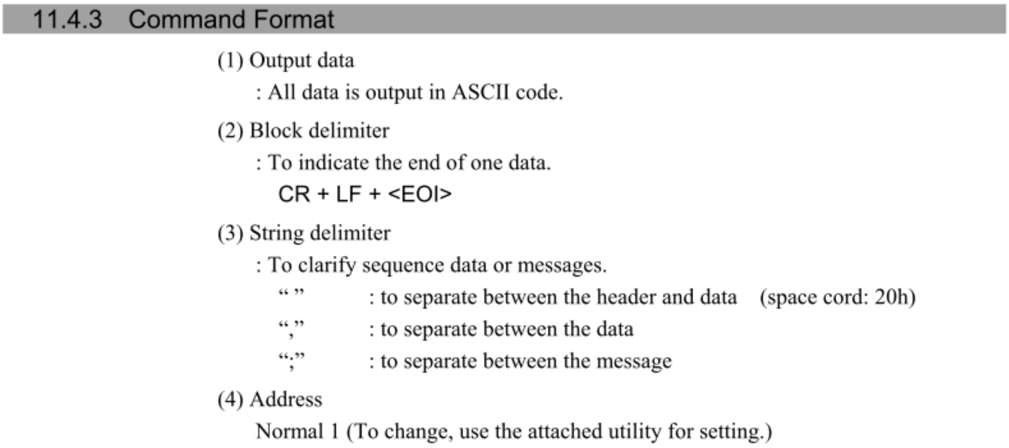
Failure Rate Alarm: ● ON ○ OFF
Ignored ICs: 20 ICs
Failure rate: 2.00 %

11.4.2 Signal Assignment

57GE-20240-751 Connector (DDK)
The signal arrangement is GP-IB standard.
Keep the connection cable at the shortest length possible. The cable length within the Handler to the tester interface panel requires 2m. Since the GP-IB standard specifies 4m as the maximum communication cable length, the cable length from the Handler's panel to the tester must be less than 2m.

Figure 4.8 Outline information about GPIB Interface

Figure 4.8 shows some general topic about GPIB and its interface used on the handler. The information is not relevant to the configuration, but users should read it to see conditions. This tool suits to IEEE-488 bus.



11.4.3 Command Format

(1) Output data
: All data is output in ASCII code.

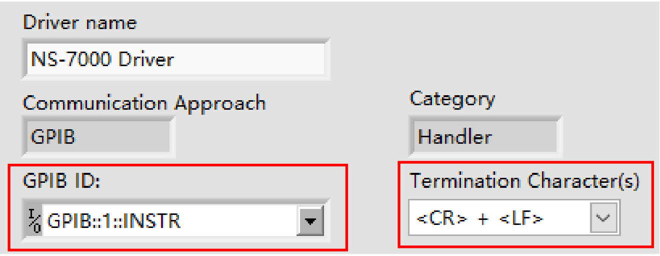
(2) Block delimiter
: To indicate the end of one data.
CR + LF + <EOI>

(3) String delimiter
: To clarify sequence data or messages.
" " : to separate between the header and data (space cord: 20h)
"," : to separate between the data
";" : to separate between the message

(4) Address
Normal 1 (To change, use the attached utility for setting.)

Figure 4.9 Command format

Figure 4.9 shows the general command format over the driver. Some may change according to real device. What we can get is that setting the Block delimiter (Termination character) as CR+LF, and GPIB address can be set as 1. See figure 4.10.



Driver name

NS-7000 Driver

Communication Approach

GPIB

Category

Handler

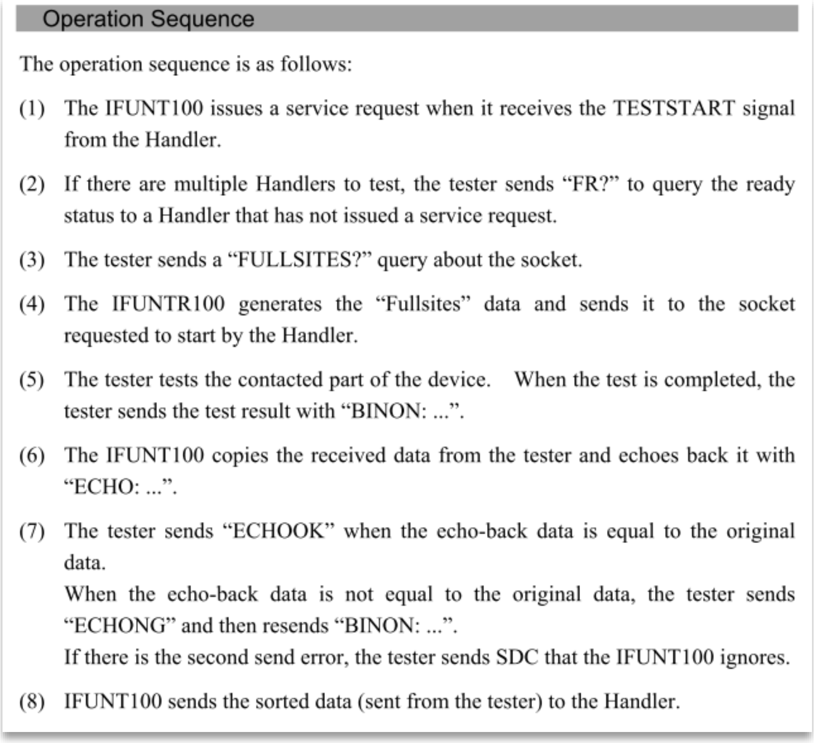
GPIB ID: GPIB::1::INSTR

Termination Character(s)

<CR> + <LF>

Figure 4.10 Set GPIB ID and Termination Character

Then, we should find the operation sequence. See figure 4.11 and 4.12. The sketch is easier to understand by the sequence and explanation from the mark number. See the sketch to match the parameters that we will configure.



Operation Sequence

The operation sequence is as follows:

(1) The IFUNT100 issues a service request when it receives the TESTSTART signal from the Handler.

(2) If there are multiple Handlers to test, the tester sends “FR?” to query the ready status to a Handler that has not issued a service request.

(3) The tester sends a “FULLSITES?” query about the socket.

(4) The IFUNTR100 generates the “Fullsites” data and sends it to the socket requested to start by the Handler.

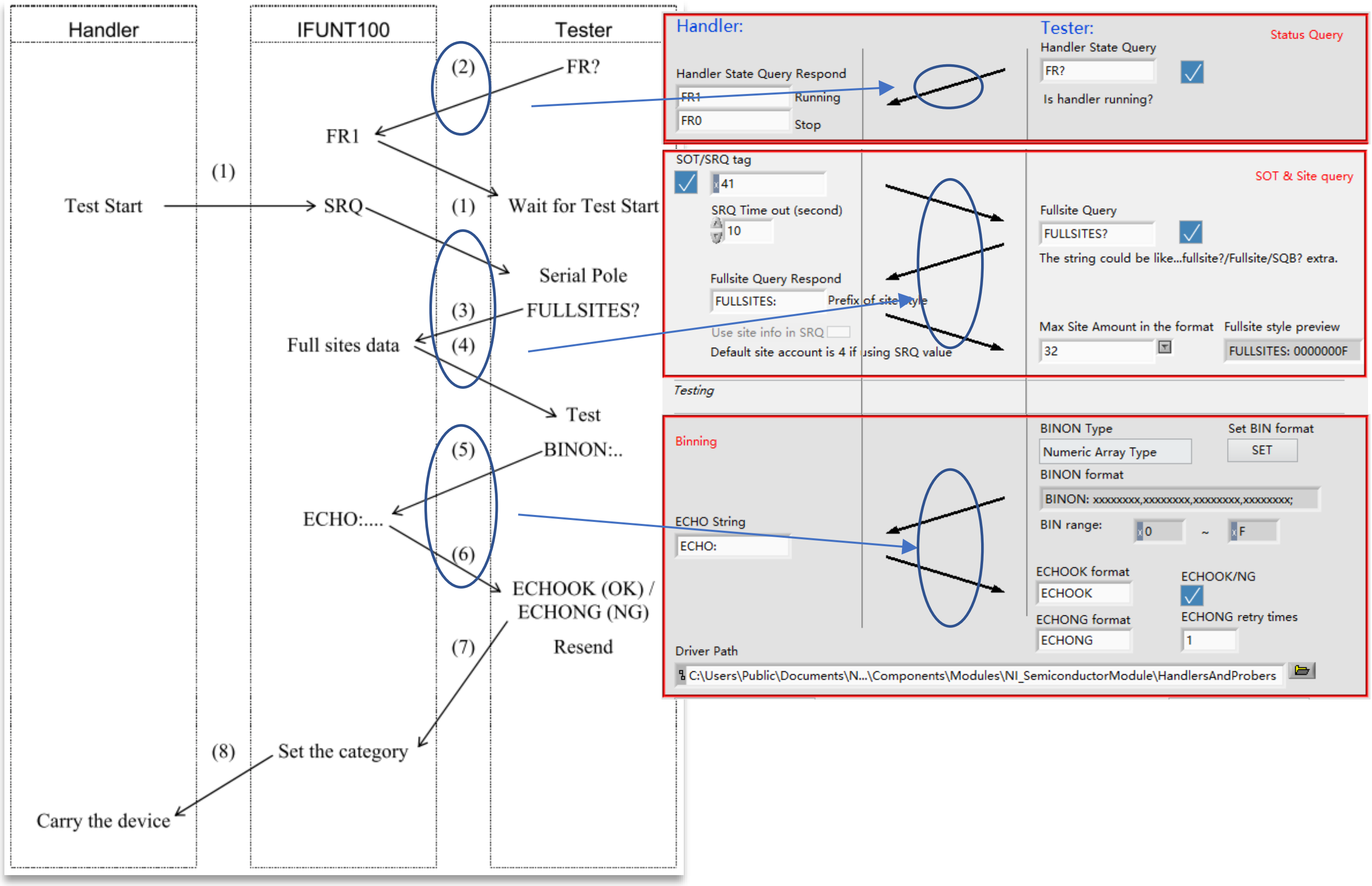
(5) The tester tests the contacted part of the device. When the test is completed, the tester sends the test result with “BINON: ...”.

(6) The IFUNT100 copies the received data from the tester and echoes back it with “ECHO: ...”.

(7) The tester sends “ECHOOK” when the echo-back data is equal to the original data.
When the echo-back data is not equal to the original data, the tester sends “ECHONG” and then resends “BINON: ...”.
If there is the second send error, the tester sends SDC that the IFUNT100 ignores.

(8) IFUNT100 sends the sorted data (sent from the tester) to the Handler.

Figure 4.11 Operation Sequence



Handler:

Handler State Query Respond

FR1 Running

FR0 Stop

Tester:

Handler State Query

FR? ☑

Is handler running?

SOT/SRQ tag

☑ 41

SRQ Time out (second)

10

Fullsite Query Respond

FULLSITES: Prefix of site style

Use site info in SRQ □

Default site account is 4 if using SRQ value

Fullsite Query

FULLSITES? ☑

The string could be like...fullsite?/Fullsite/SQB? extra.

Max Site Amount in the format: 32

Fullsite style preview: FULLSITES: 0000000F

BINON Type: Numeric Array Type

Set BIN format: SET

BINON format

BINON: xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx;

BIN range: 0 ~ F

ECHO String

ECHO:

ECHOOK format: ECHOOK

ECHOOK/NG ☑

ECHONG format: ECHONG

ECHONG retry times: 1

Driver Path

C:\Users\Public\Documents\N...\Components\Modules\NI_SemiconductorModule\HandlersAndProbers

Figure 4.12 Operation Sequence with sketch map

In figure 4.12 we match the communication sequence and configured the parameters according to the user guide in figure 4.11,4.14 and 4.15:

NS-7000 Handler Status Query needs “FR?”, so we check the tick and set “FR?” in the query bar.

Expected response for running is “FR1” so we set “FR1” in Running bar; stop is represented by “FR0” so we set “FR0” in Stop bar.

NS-7000 handler send SRQ message 0x41 as SOT signal to start the test, then we check the tick and set 41 in hex. SRQ time depends on the application and here we use default 10 seconds.

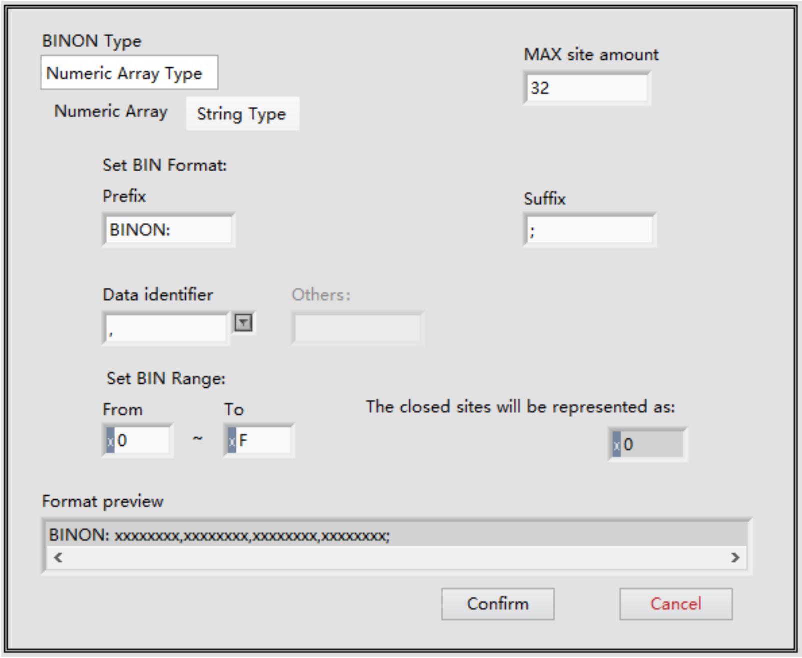
After receive the SOT signal, tester should send fullsite query, using string “FULLSITE?”. Thus, we check the tick and set “FULLSITE?” in the bar.

According to figure 4.15, we know that there is a prefix string “Fullsite ”, so we fill “Fullsite ” in the bar. Mention that there is a space and you should type it in too.

Then, choose Max Site Amount, NS-7000 need 32bits so we choose 32bit for it.

Binning part:

Click SET button and set the bin format. See the figure below.



BINON Type: Numeric Array Type

MAX site amount: 32

Set BIN Format:

Prefix: BINON:

Suffix: ;

Data identifier: ,

Others:

Set BIN Range:

From: x0 ~ To: xF

The closed sites will be represented as: x0

Format preview: BINON: xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx;

Figure 4.13 Configure bin settings

Choose Numeric Array Type for NS-7000, because it's "BINON:xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx" format. You may see the MAX site amount on the top right coner is already there shown 32, which cannnot be changed in this page.

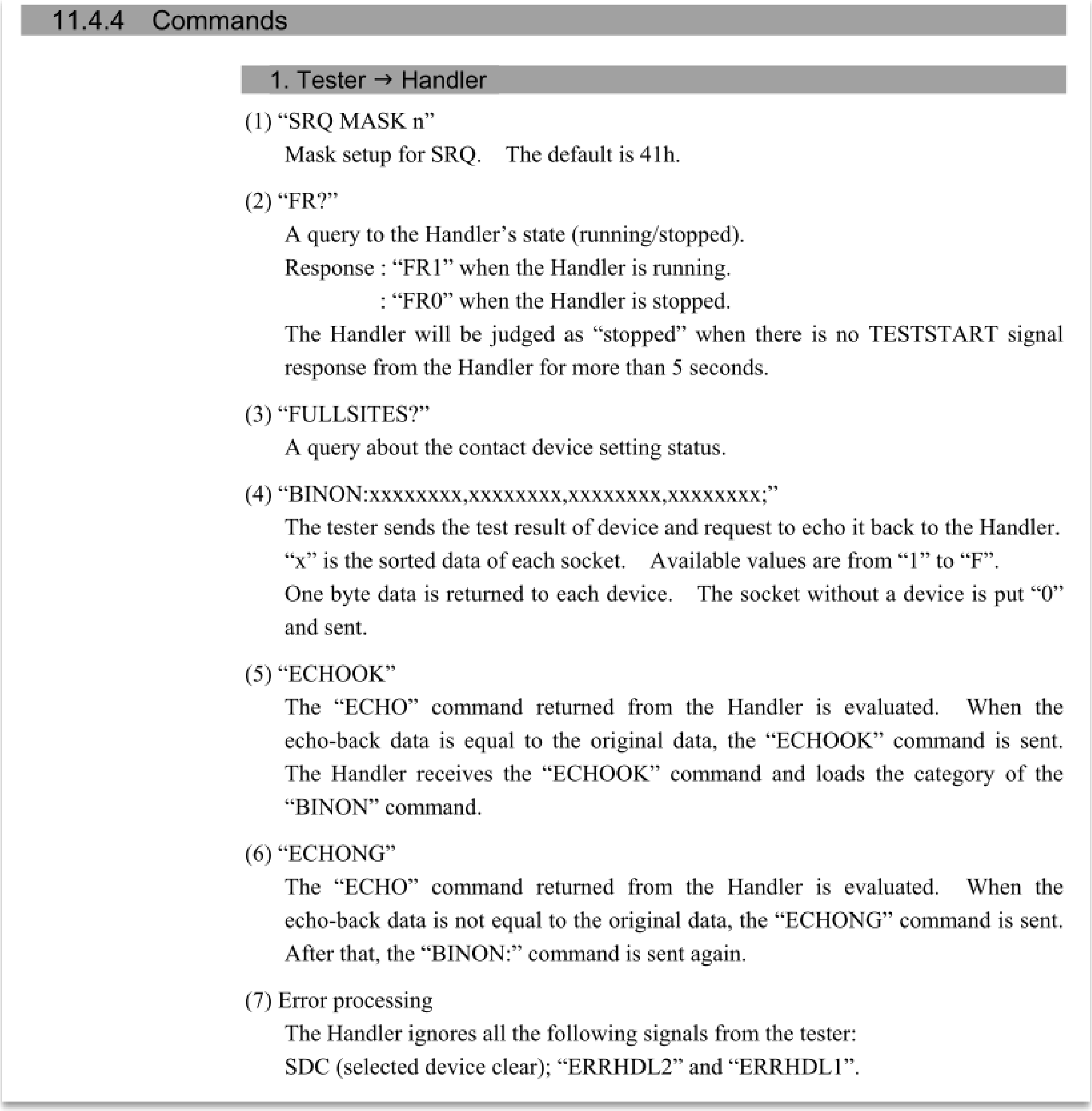
According to the datasheet shown in figure 4.14, there's a pefix "BINON:", and a suffix ";", so we set them in the corresbonding bars. The data identifier, which is used to separate the 8digits of bin data, choose "," in the pull-down list for NS-7000 that you can see in the datasheet.

Set the BIN range according to your test, or leave 0~F as default. The closed sites will be represented as 0 and its not configurable.

From the preview bar, you can see the format preview and if it is correct, click Confirm button to finish configuration of bin format.

After setting bin format, NS-7000 need to configure ECHO information. The prefix of echo data is "ECHO:". If the data comparation is correct, send "ECHOOK", so we fill "ECHOOK" in the ECHOOK format bar. If the data comparation does not succeed, send "ECHONG", so we fill "ECHONG" in the ECHONG format bar. Then, you can set ECHONG retry times acording to your own test plan.

Figure 4.14 and 4.15 explains the parameters in detail. Users can refer to them for detailed information.



11.4.4 Commands

1. Tester → Handler

(1) “SRQ MASK n”

Mask setup for SRQ. The default is 41h.

(2) “FR?”

A query to the Handler’s state (running/stopped).

Response : “FR1” when the Handler is running.

: “FR0” when the Handler is stopped.

The Handler will be judged as “stopped” when there is no TESTSTART signal response from the Handler for more than 5 seconds.

(3) “FULLSITES?”

A query about the contact device setting status.

(4) “BINON:xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx;”

The tester sends the test result of device and request to echo it back to the Handler. “x” is the sorted data of each socket. Available values are from “1” to “F”.

One byte data is returned to each device. The socket without a device is put “0” and sent.

(5) “ECHOOK”

The “ECHO” command returned from the Handler is evaluated. When the echo-back data is equal to the original data, the “ECHOOK” command is sent. The Handler receives the “ECHOOK” command and loads the category of the “BINON” command.

(6) “ECHONG”

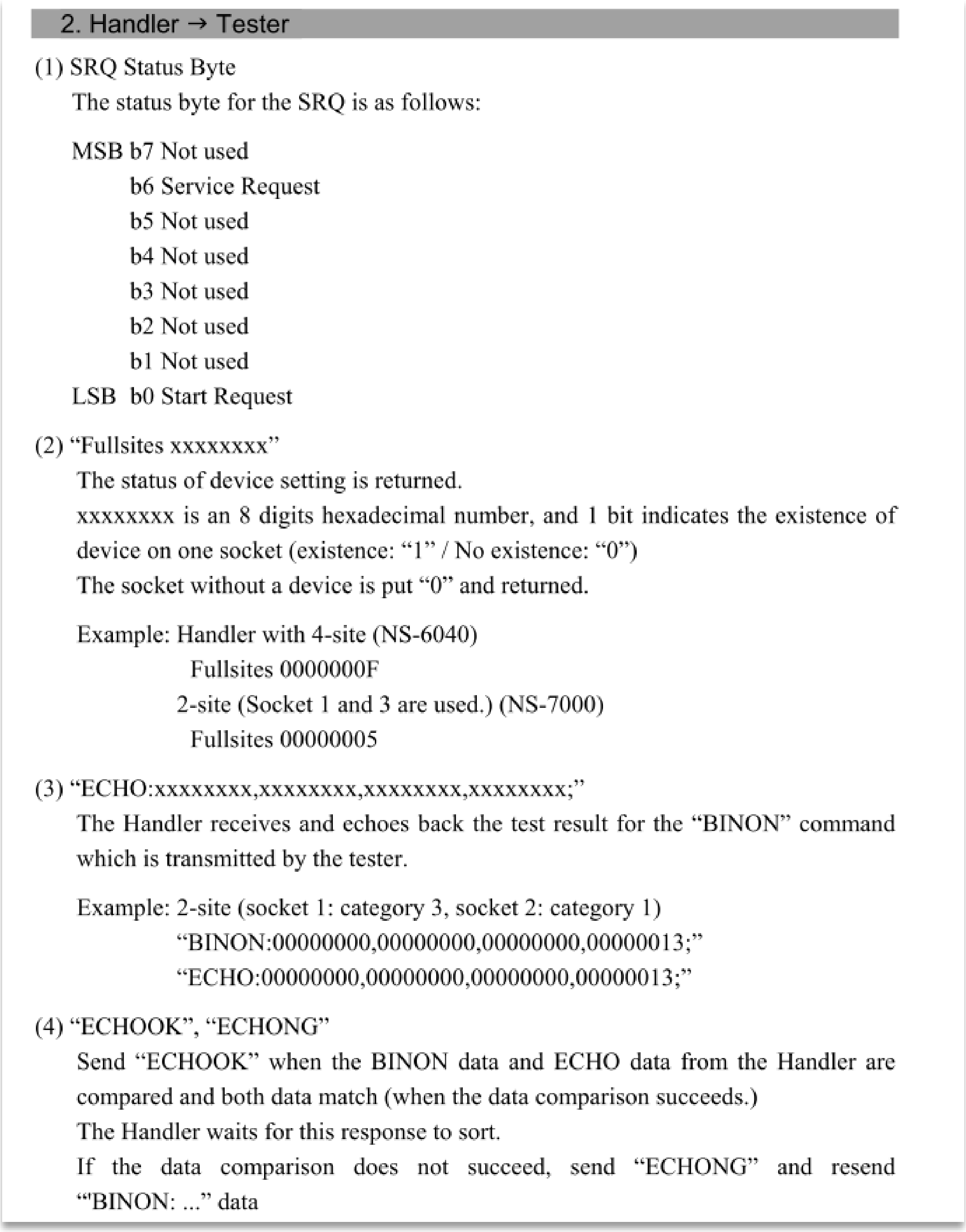
The “ECHO” command returned from the Handler is evaluated. When the echo-back data is not equal to the original data, the “ECHONG” command is sent. After that, the “BINON:” command is sent again.

(7) Error processing

The Handler ignores all the following signals from the tester:

SDC (selected device clear); “ERRHDL2” and “ERRHDL1”.

Figure 4.14 Commands - Tester send to Handler



2. Handler → Tester

(1) SRQ Status Byte

The status byte for the SRQ is as follows:

MSB b7 Not used
b6 Service Request
b5 Not used
b4 Not used
b3 Not used
b2 Not used
b1 Not used
LSB b0 Start Request

(2) "Fullsites xxxxxxxx"

The status of device setting is returned.
xxxxxxxx is an 8 digits hexadecimal number, and 1 bit indicates the existence of device on one socket (existence: "1" / No existence: "0")
The socket without a device is put "0" and returned.

Example: Handler with 4-site (NS-6040)
Fullsites 0000000F
2-site (Socket 1 and 3 are used.) (NS-7000)
Fullsites 00000005

(3) "ECHO:xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx;"

The Handler receives and echoes back the test result for the "BINON" command which is transmitted by the tester.

Example: 2-site (socket 1: category 3, socket 2: category 1)
"BINON:00000000,00000000,00000000,00000013;"
"ECHO:00000000,00000000,00000000,00000013;"

(4) "ECHOOK", "ECHONG"

Send "ECHOOK" when the BINON data and ECHO data from the Handler are compared and both data match (when the data comparison succeeds.)
The Handler waits for this response to sort.
If the data comparison does not succeed, send "ECHONG" and resend "'BINON: ..." data

Figure 4.15 Command – Handler send to Tester

Step3. Examine and generate

Examine the information that you just set, then click **Create and save driver** button to generate the driver. A window will pop up if you did successfully. See figure 4.16.

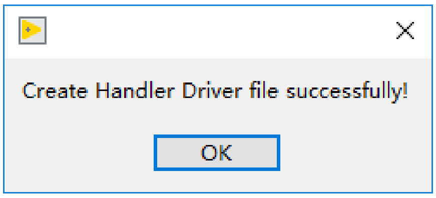

Create Handler Driver file successfully!

Figure 4.16 Create handler driver successfully

If you generate a driver that a same name one has already in the folder, one window will pop up and mention you to replace it or not. Click Yes to replace the file using the current configuration. Click Cancel will not save the configuration, then you can change the name and save it again.

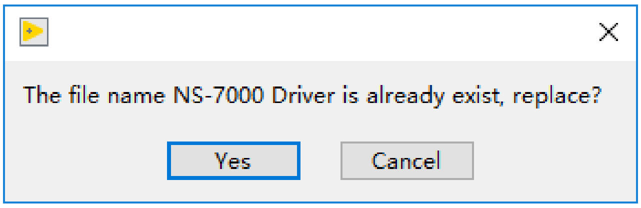

The file name NS-7000 Driver is already exist, replace?

Figure 4.17 Duplicated files

Step4. Test

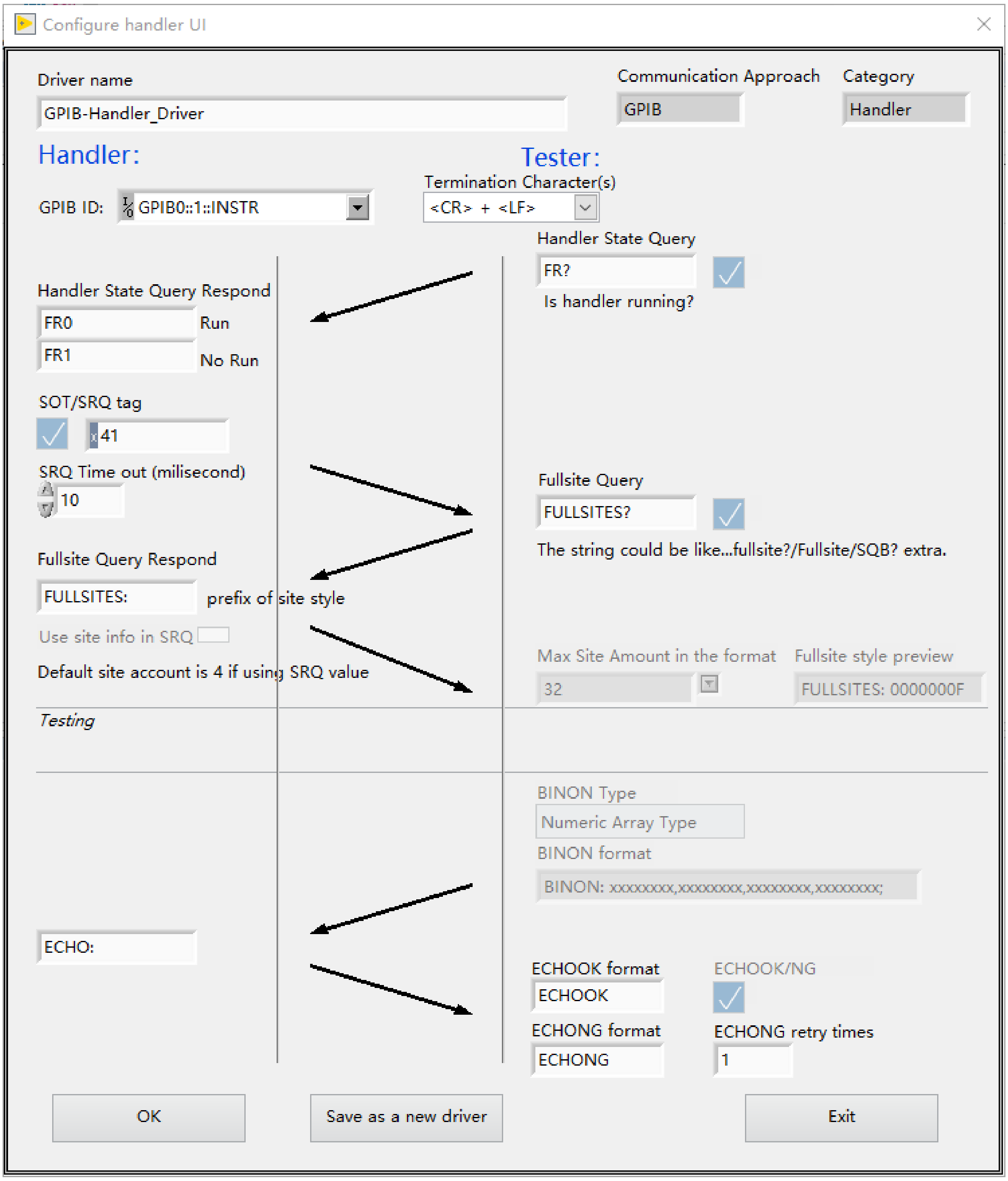
After finishing the configuration, run handler to test the communication and see if the required parameters are correct. If the function is not ideal, check the details and re-configure the driver.

Before testing, users can click the “Configure Handler/Prober” button at the right side of Handler/Prober Mode to check the parameters that will be used in test. Figure 4.18 shows the panel. Users should check the parameters, then, one of the three following events may happen:

1.If it is ok, click ok or cancel to continue test.

2.If there’s any mismatch, the strings can be modified, but the structure is unchangeable. Click OK to use the new value. But if users load it again, it will roll back to the configured values, which means it was saved temporarily. This function is usually for adjustment of drivers temporarily. If you want to change the structure of the protocol, you must re-generate the driver.

3. If you changed the parameter values and want to save the changes to make it a new driver, you should rename the driver at Driver name and click “Save as a new driver” button. Then, close the Configure Station Settings panel and re-open it. The new driver you just set is right in the Custom Handler/Prober list. Using this function can save the workable drivers you just adjusted.



Driver name: GPIB-Handler_Driver

Communication Approach: GPIB

Category: Handler

GPIB ID: GPIB0::1::INSTR

Termination Character(s): <CR> + <LF>

Handler State Query: FR? ☑

Is handler running?

Handler State Query Respond: FR0 Run; FR1 No Run

SOT/SRQ tag: ☑ 41

SRQ Time out (milisecond): 10

Fullsite Query: FULLSITES? ☑

The string could be like...fullsite?/Fullsite/SQB? extra.

Fullsite Query Respond: FULLSITES: prefix of site style

Use site info in SRQ □

Default site account is 4 if using SRQ value

Max Site Amount in the format: 32

Fullsite style preview: FULLSITES: 0000000F

Testing

BINON Type: Numeric Array Type

BINON format: BINON: xxxxxxxx,xxxxxxxx,xxxxxxxx,xxxxxxxx;

ECHO:

ECHOOK format: ECHOOK

ECHOOK/NG ☑

ECHONG format: ECHONG

ECHONG retry times: 1

Figure 2.3 Re-confirm the parameters at pre-test view

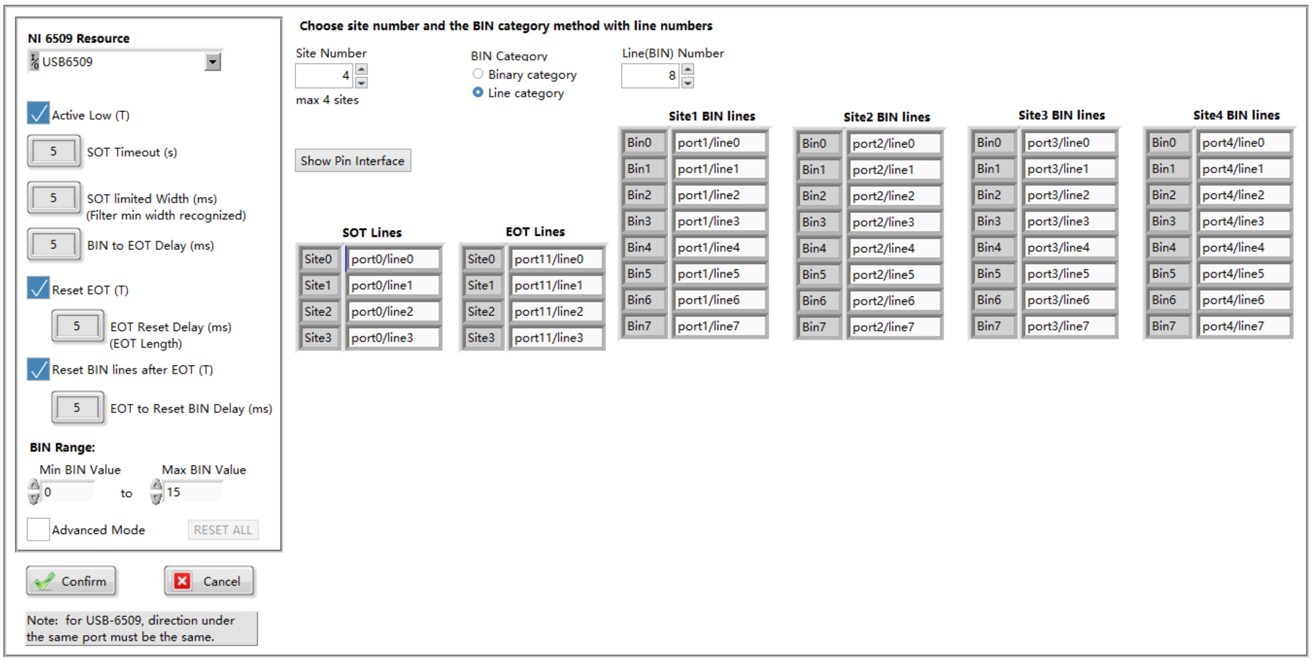
Since then, we finished the configuration of NS-7000 handler and generated the driver of it. Readers should know that, different handlers may have different parts, some parts may be not necessary, users should do it flexible according to the real cases.

There might be some handler do not suit to this tool, the data structure or communication process may be different at all. Thus, you should develop the driver by yourself. Please refer to the Reference 3 - Customize the driver if the tool does not cover. That document helps you develop a handler driver from scratch.

5. Configure TTL-Handler Driver

5-1.Panel configuration and parameter introduction

Click “Configure Handler/Prober” button then the panel will pop out. Engineers can configure the parameters according to the lines needed. Figure 5.1 shows the configuration panel.



NI 6509 Resource: USB6509

☑ Active Low (T)

5 SOT Timeout (s)

5 SOT limited Width (ms) (Filter min width recognized)

5 BIN to EOT Delay (ms)

☑ Reset EOT (T)

5 EOT Reset Delay (ms) (EOT Length)

☑ Reset BIN lines after EOT (T)

5 EOT to Reset BIN Delay (ms)

BIN Range: Min BIN Value 0 to Max BIN Value 15

□ Advanced Mode

Note: for USB-6509, direction under the same port must be the same.

Choose site number and the BIN category method with line numbers

Site Number 4 (max 4 sites)

BIN Category: ○ Binary category ● Line category

Line(BIN) Number 8

SOT Lines		EOT Lines	
Site0	port0/line0	Site0	port11/line0
Site1	port0/line1	Site1	port11/line1
Site2	port0/line2	Site2	port11/line2
Site3	port0/line3	Site3	port11/line3

Site1 BIN lines		Site2 BIN lines		Site3 BIN lines		Site4 BIN lines	
Bin0	port1/line0	Bin0	port2/line0	Bin0	port3/line0	Bin0	port4/line0
Bin1	port1/line1	Bin1	port2/line1	Bin1	port3/line1	Bin1	port4/line1
Bin2	port1/line2	Bin2	port2/line2	Bin2	port3/line2	Bin2	port4/line2
Bin3	port1/line3	Bin3	port2/line3	Bin3	port3/line3	Bin3	port4/line3
Bin4	port1/line4	Bin4	port2/line4	Bin4	port3/line4	Bin4	port4/line4
Bin5	port1/line5	Bin5	port2/line5	Bin5	port3/line5	Bin5	port4/line5
Bin6	port1/line6	Bin6	port2/line6	Bin6	port3/line6	Bin6	port4/line6
Bin7	port1/line7	Bin7	port2/line7	Bin7	port3/line7	Bin7	port4/line7

Figure 5.1 TTL-driver configuration panel

On the left of the panel, it shows configuration for:

NI 6509 Resource: Choose the instrument name that you use in the communication.

Active Low(T): Check if the active voltage level is Low. If active voltage is high, leave it uncheck.

SOT Timeout (s): Set time out value in second of waiting SOT.

SOT limited Width(ms): Set a filter width of time in millisecond to recognize affective voltage level to avoid glitches. A SOT signal length should be longer than this setting.

BIN to EOT Delay(ms): Set the time delay in millisecond to set EOT signal after BIN signal been set.

Reset EOT(T): Check if the EOT signal need to be reset.

EOT Reset Delay(ms): Set the time delay in millisecond to reset the EOT signal. This control is available when **Reset EOT(T)** is checked.

Reset BIN lines after EOT(T): Check if the BIN lines should be reset after EOT signal end.

EOT to Reset BIN Delay(ms): Set the time delay in millisecond to reset the BIN lines after EOT signal ends. This control is available when **Reset BIN lines after EOT(T)** is checked.

BIN Range: Set the minimum BIN value in **Min BIN Value** and the maximum BIN value in **Max BIN Value**. The values are natural number from 0 to 255. BIN number should be continuous from Min value to Max value. The default value is 0 for **Min BIN Value** and 15 for **Max BIN Value**. User can change the value when **Advanced Mode** is checked.

Advanced Mode: Check to modify the **BIN Range**, **SOT lines EOT lines** and the **Site BIN lines**. For default, we do not encourage user to modify these lines. Engineers should connect physical cables according to the settings listed in the chart.

RESET ALL: Reset the configuration as default value. User can change the value when **Advanced Mode** is checked.

Figure 5.2 helps understand the timing set of the parameters:

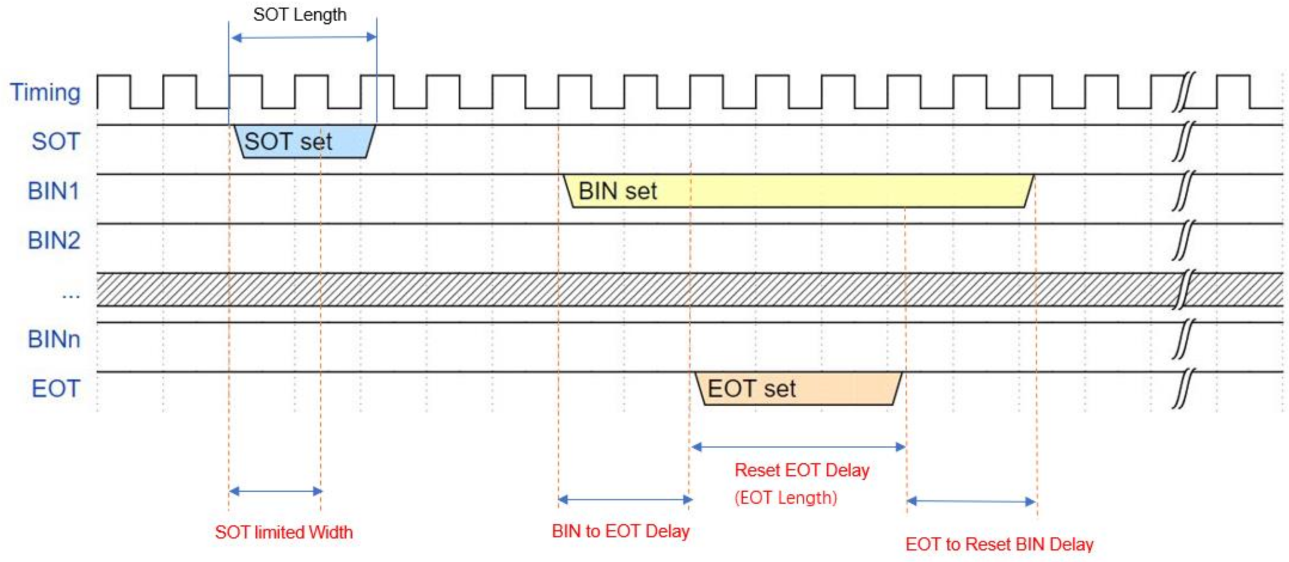


Figure 5.2 timing set of TTL driver

On the middle of the panel, it shows the Site and BIN information with Line configuration:

Site Number: Select the site numbers. Max support 4 sites.

BIN category: Choose the BIN category between Binary and Line mode. Binary mode means the BIN result is binary coded and all bin lines are used for representing the value. For example, if BIN 5, line1 and line3 is active for 5 represents 0b101 in binary code. Line mode is also known as Parallel mode, which means that each line dedicates to one BIN. For example, if BIN 5, line 5 is active.

Bit Number/ Line(BIN) Number: Select Bit Number/ Line(BIN) Number in Binary mode and Line mode correspondingly.

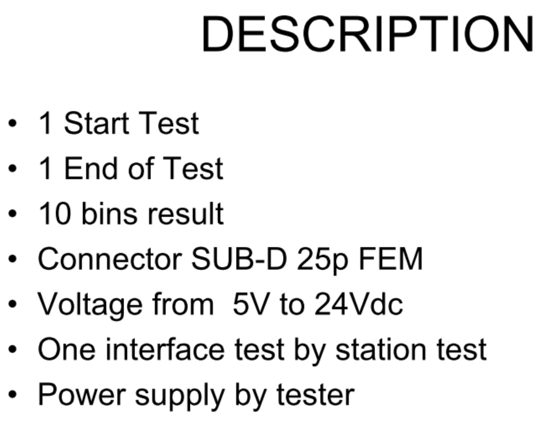
Show Pin Interface: Click to show the pin connection interface of NI 6509.

5-2. Instance analysis

Use ISMECA handler as an example to help explain how to configure the driver. Generally, we use 4 steps to complete the process.

Step1. Read the datasheet

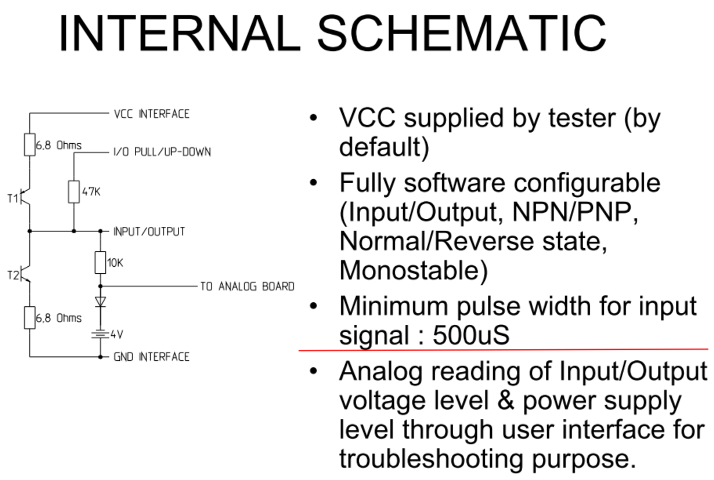
Read the datasheet, understand the timing nodes and bin category. If the timing parameter is not mentioned in the datasheet, leave it as default. For most TTL handlers, default value works. For example, figure 5.3 describes connection of bins, figure 5.4 describes electrical parameters of the handler, figure 5.5 shows the timing sequence which helps you to understand the function.



DESCRIPTION

- 1 Start Test
- 1 End of Test
- 10 bins result
- Connector SUB-D 25p FEM
- Voltage from 5V to 24Vdc
- One interface test by station test
- Power supply by tester

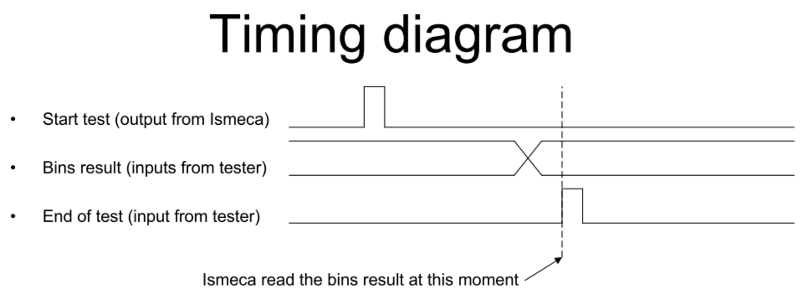
Figure 5.3 Description of bin connection



INTERNAL SCHEMATIC

- VCC supplied by tester (by default)
- Fully software configurable (Input/Output, NPN/PNP, Normal/Reverse state, Monostable)
- Minimum pulse width for input signal : 500uS
- Analog reading of Input/Output voltage level & power supply level through user interface for troubleshooting purpose.

Figure 5.4 Description of function details



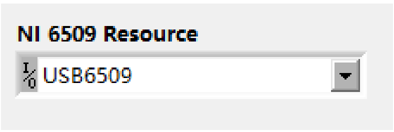
Timing diagram

Start test (output from Ismeca)
Bins result (inputs from tester)
End of test (input from tester)
Ismeca read the bins result at this moment

Figure 5.5 Illustration of timing sequence

Step2. Configure the panel

Configure the panel according to the timing node that affect the timing sequence. Don’t forget to choose the digital device interface.

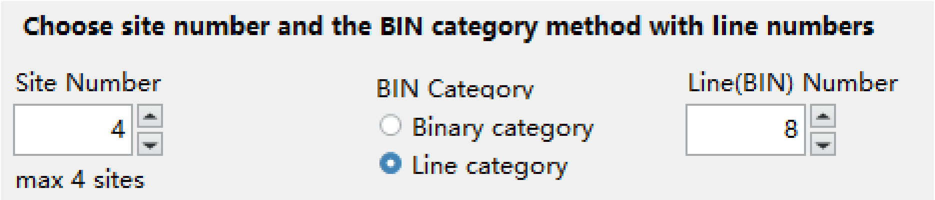


NI 6509 Resource

USB6509

Figure 5.6 Configure digital interface

The ports for Bin have already defined as default. It is recommended to use the default connection. You can choose site numbers and bin numbers at the top of the panel. The maximum site is 4 and the maximum bin line is 16. Here we choose 4 sites in line category. The bin is from bin 0 to bin 7. The ports will show up according to the settings.



Choose site number and the BIN category method with line numbers

Site Number: 4

max 4 sites

BIN Category

○ Binary category

● Line category

Line(BIN) Number: 8

Figure 5.7 Configure site number and bin number

You can define the ports as well(by using Advanced mode). However to notify that, the port of EOT SOT and Bin should be one direction. For example, if port 1 is output from tester to handler, all the settings for port 1 should be output, sending signal from tester to handler. Otherwise error will occur.

Step3. Examine and generate

Check the configuration and click ok to use this driver. If you want to configure it from default, select Advanced mode and click RESET ALL to reset the values.

Step4. Test

After finishing the configuration, run handler to test the communication and see if the required parameters are correct. If the function is not ideal, check the detail and re-generate the driver.

Since then, we finished the configuration of ISMECA TTL handler for 4site and 8 bins, generated the driver of it.

Reference

1. STS Software R&D: https://nitalk.jiveon.com/groups/sts-software-rd
2. CoE Page: https://nitalk.jiveon.com/groups/apac-semists-center-of-excellence
3. Related resources – handler driver structure and how to customize from coding: Perfore:1666 //Sales/Semi/HandlerProber/How to Write a Handler Driver.pptx.

Additional Information

NI Talk Page: **APAC Semi & STS Center of Excellence**

Email: **APAC.STS.COE@ni.com**

APAC
Semi & STS
Center of Excellence