

# BIT-2100B Series Switch System

## User Manual 2.20



**BitifEye Digital Test Solutions GmbH**  
Herrenberger Strasse 130  
71034 Boeblingen, Germany

[info@bitifeye.com](mailto:info@bitifeye.com)  
[www.bitifeye.com](http://www.bitifeye.com)

## Notices

© BitifEye Digital Test Solutions GmbH 2024.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from BitifEye.

## Edition

Jan 23, 2024

## Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, BitifEye disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. BitifEye shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should BitifEye and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

## Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

## Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as “Commercial computer software” as defined in DFAR 252.227-7014 (June 1995), or as a “commercial item” as defined in FAR 2.101(a) or as “Restricted computer software” as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to BitifEye’s standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

## Safety Notices


### CAUTION


A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.


### WARNING


A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.


## Product Labels


 The BitifEye Logo.


 This electronic product is in compliance with the EMC and Safety regulations of the European Community.


 Read the instructions in the BIT-2100 datasheet, as well as this manual, before using this electronic product.

 This electronic product must be operated with AC power.

 Ethernet port to connect to a LAN.

 USB device port to connect to a USB host.

 USB host port to connect to a USB device.

 Attachment point for Kensington lock.

## Technical Assistance

If you need product assistance or if you have suggestions, contact BitifEye. You will find the contact information on the BitifEye homepage at:

<http://www.bitifeye.com>

Representatives of BitifEye are available during standard German business hours.

Before you contact BitifEye, please note the actions you took before you experienced the problem. Then describe those actions and the problem to the technical support engineer.

### Find a Mistake?

We encourage comments about this publication. Please report any mistakes to BitifEye ([support@bitifeye.com](mailto:support@bitifeye.com))

# Contents

<b>1. Introduction</b>	<b>8</b>
Safety Precautions	8
Scope and Datasheet	8
Intended Use	8
Terminology	8
Relay Types	9
What's New	9
<b>2. Specifications</b>	<b>10</b>
<b>3. Options</b>	<b>10</b>
<b>4. Usage</b>	<b>11</b>
Initial Inspection	11
Setup	11
Power	11
Display	11
Connectors and Controls	12
Primary/Secondary Functionality	13
Cleaning	14
Disposal	14
<b>5. General Operation</b>	<b>15</b>
Preface	15
VISA Connection	15
Keysight IO Libraries	15
Network Connection	15
Connecting to LAN	15
Keysight Connection Expert	16
Accessing the Web Interface	17
Using the Java GUI	18
USB Connection	19
Connecting to USB	19
Determining the VISA address	19
<b>6. Software Updates</b>	<b>20</b>
Firmware Update	20
System Upgrade	21
If anything goes wrong	22
<b>7. Remote Programming</b>	<b>23</b>
General Syntax	23
Data Formats	23
Strings	23
Integers	24
Boolean Flags	24
Keywords	24
Error Handling	24
Module Numbering	24
IEEE-488 Commands	25

*IDN?	25
*RST	25
*TST?	25
Common Commands	26
:SYSTem:ERRor?	26
:SYSTem:ERRor:COUNt?	26
:SYSTem:HELP:HEADers?	26
:SYSTem:SELFtest?	26
:SYSTem:CONFIguration?	26
Relay Control	28
Module Identifier Strings	28
Relay Identifier Strings	29
:RELay:COUNt?	30
:RELay:SLOT?	30
:RELay:TYPE?	30
:RELay:SERial?	30
:RELay:TERMinated?	30
:RELay:LATChing?	30
:RELay:PATH	31
:RELay:PATH?	31
:RELay:SWITCh:COUNt?	31
:RELay:SWITCh:TERMinated?	31
:RELay:SWITCh:LATChing?	31
:RELay:SWITCh:PATH	32
:RELay:SWITCh:PATH?	32
:RELay:SWITCh:SERial?	32
:RELay:SWITCh:NCYCles?	32
Legacy Commands	33
Network Configuration	34
:NETWork:RESet	34
:NETWork:INTerface#:CONFIgure	34
:NETWork:INTerface#:CONFIgure:MODE?	34
:NETWork:INTerface#:CONFIgure:IP?	34
:NETWork:INTerface#:CONFIgure:SUBNetmask?	35
:NETWork:INTerface#:CONFIgure:GATeway?	35
:NETWork:INTerface#:IP?	35
:NETWork:IP?	35
Programming Examples	36
Best Practice	36
Querying	37
Switching	37
<b>8. User Serviceable Parts</b>	<b>38</b>
Changing Fuses	38
Dismounting Modules	39
Mounting Modules	40
<b>9. List of Acronyms</b>	<b>41</b>

## List of Figures

Figure 1: Connectors and Controls on Front Side.....	12
Figure 2: Connectors and Controls on Rear Side.....	12
Figure 3: How to connect a primary/secondary configuration.....	13
Figure 4: The BIT-2100 shown in the Keysight C Expert.....	16
Figure 5: Instrument Web Interface.....	17
Figure 6: Connecting to the instrument with the GUI.....	18
Figure 7: Main window of the GUI.....	18
Figure 8: Virtual COM Port Address Shown in Device Manager.....	19
Figure 9: Module identifier string examples.....	28
Figure 10: Relay identifier string examples.....	29
Figure 11: Removing Fuses.....	38
Figure 12: Inserting Fuses.....	38
Figure 13: Removing screws from modules.....	39
Figure 14: Removing a module.....	39
Figure 15: Inserting a module.....	40

# 1. Introduction

## Safety Precautions

### WARNING

This manual describes the proper usage of the BIT-2100 Series Switch System. All usage beyond the specifications or the intended use is not recommended and might cause impaired safety for the user.

## Scope and Datasheet

This user manual does not cover specifications. Please also refer to the BIT-2100B Datasheet (available from the BitifEye website).

This user manual is for the BIT-2100B Series Switch System, and does not cover the BIT-2100A Series Switch System.

## Intended Use

The BIT-2100 Series Switch System is intended to be used as test and measurement equipment.

This instrument is designed as laboratory equipment, and not intended for home use. The SMA connectors allow connection to appropriate electronic circuits or other laboratory equipment, as long as the specifications (see datasheet) are not violated.

The Ethernet and USB connectors allow to connect the BIT-2100 Series Switch System to standard computer equipment for the purpose of remote control, as well as for software updates.

Typical use case might be:

- multiplex several serial high-speed serial lanes of a source device to an oscilloscope, to efficiently test all lanes of the device
- multiplex a single serial high-speed signal source to the ingress ports of a multi-port device, to test all receivers of the device
- connect a four-port vector network analyzer to an eight-lane cable, to test all lanes without manual re-connections

In all of these use cases, the benefit of using a switch system is that the tests and measurements can be conducted without manual re-connections, and they can even be fully automated with software.

## Terminology

This manual uses the following terminology:

- *mainframe*: the chassis with all internal components (power supply, CPU, control unit with display, etc.), in which up to five modules can be mounted
- *module*: one of the five modules, which contain relays, that can be mounted into a mainframe
- *relay*: an electromechanical switch; a module can hold up to six relays
- *path*: the state of a relay, i.e. which terminal is routed to which; typically indicated by an integer number; 0 indicates that all terminals are disconnected
- *terminal*: an RF (SMA or 3.5 mm) connector of a relay
- *terminated*: terminated with 50  $\Omega$  into GND, also known as “absorptive”; see also page 9
- *unterminated*: left without termination, also known as “floating” or “reflective open”; see also page 9



## Relay Types

The modules supported by the BIT-2100 Series Switch System all provide high RF bandwidth, but other than that differ in various ways:

- each relay has two to eight terminals, of which one a time can be connected to a common terminal
  - terminology: e.g. a relay with two terminals (plus the common terminal) is indicated as “2:1”
  - the “path” indicates which terminal (a number starting at 1) is currently connected to the common terminal
- some relays support an additional path (denoted as path 0), which is when all terminals at the same time are disconnected from the common terminal
- the terminals which are currently not connected to the common terminal are either terminated or unterminated (see also page 8)
- most relays are latching, i.e. retain their path after power is turned off, while some fall back into a default path (also known as “fail-safe”)
- some modules contain a single relay, some modules contain up to six independent relays

## What’s New

The most important changes of the BIT-2100B over the BIT-2100A are:

- improved mainframe: display to show IP address, user controls at the front, data connections on the rear, larger chassis so lab equipment can be put on top
- new module types: terminated 16:1, unterminated 16:1, terminated 8:1, unterminated 8:1, terminated 2:1
- smart modules: data such as relay serial numbers and switch cycle counters can be queried
- new software platform: the same reliable platform as in the BitifEye BIT-3000 DSGA
- new USB functionality: using USB-VCM instead of USB-TMC, which allows to connect two frames into a primary-secondary configuration

## 2. Specifications

For specifications of the BIT-2100 Series Switch System, please refer to the datasheet.

## 3. Options

A BIT-2100 Series Switch System consists of the following components:

- a BIT-2100B mainframe (with CPU, power supply, control unit)
- up to five modules (each with one or more relays)

## 4. Usage

### Initial Inspection

Before using the switch system, make sure that the following parts are included in the package:

- BIT-2100B mainframe
- Up to five modules, mounted in the mainframe (according to order)

### Setup

The BIT-2100 Series Switch System can be operated as a desk-top instrument or can be mounted in a standard 19" rack. Make sure the specified environmental conditions are met.

### Power

**WARNING** Only connect the AC connector at the rear side of the instrument to a power outlet using an appropriate 3-wire cable! Make sure you use a power outlet with a grounding terminal!

This device is sensitive to high voltage peaks. If there is a risk of power instability, it is recommended to add a UPS device to avoid any damage.

To turn the device on, flip the switch on the front side of the instrument to the position labeled "I". The blue LED next to the switch lights up to indicate the power state.

To turn the device off, flip the switch on the front side of the instrument to the position labeled "O".

### Display

The display will light up as soon as the instrument is turned on. As soon as the operating system boot process is done, text will appear on the display.

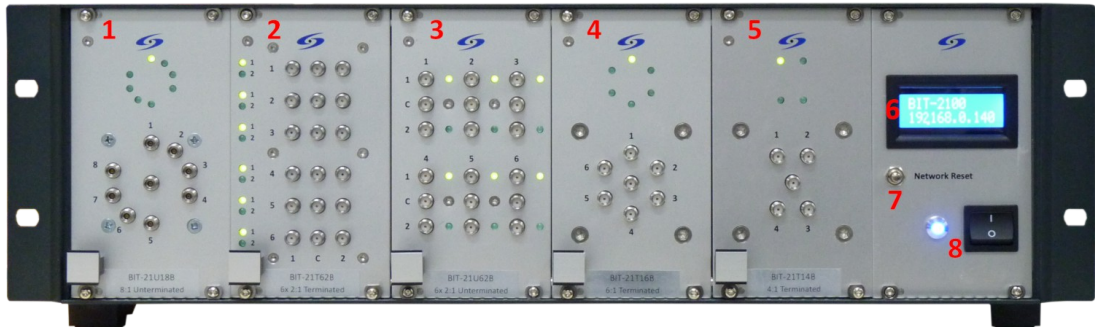
As soon as the firmware is ready, the display will change to "BIT-2100". If a network connection is available, the IP address will be displayed in the second line. If multiple IP addresses are available, the display will cycle through the available IPs.

When the instrument is in a primary/secondary configuration (see page 13), an "P" (primary) or "S" (secondary) will be shown on the top-right corner of the display. Note that it might take up to two minutes until the display updates after connecting or disconnecting a primary/secondary configuration.

During and after a system upgrade, the display will indicate the progress (see also page 21).

## Connectors and Controls

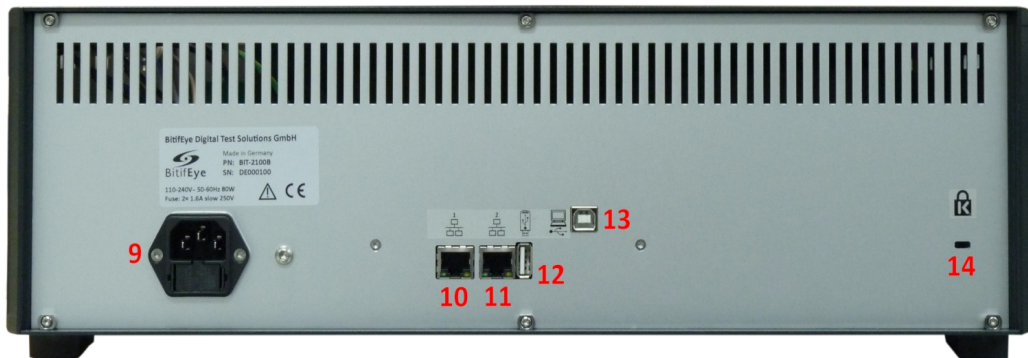
Figure 1 shows the connectors and controls at the front side of the instrument.



**Figure 1: Connectors and Controls on Front Side**

- 1-5 Modules 1 through 5 (page 10)
- 6 Display (page 11)
- 7 LAN reset (page 15)
- 8 Main power switch and indicator LED (page 11)

Figure 2 shows the connectors and controls at the rear side of the instrument.



**Figure 2: Connectors and Controls on Rear Side**

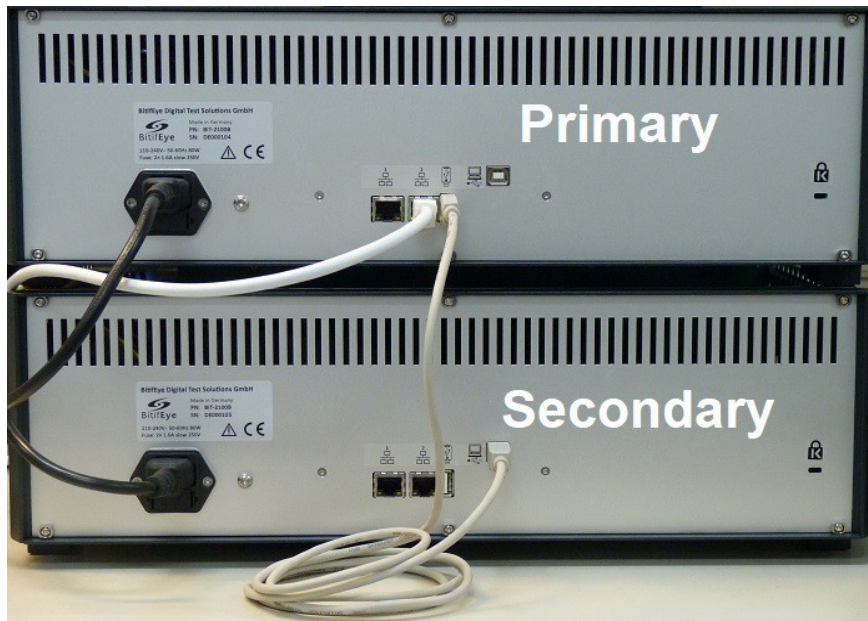
- 9 Power inlet with fuse box (page 11)
- 10, 11 Ethernet connectors 1 and 2 for remote control (page 15) and firmware updates (page 20)
- 12 USB connector for system upgrades (page 21), and for the primary-side of a primary/secondary setup (page 13)
- 13 USB connector for remote control (page 19), and for the secondary-side of a primary/secondary setup (page 13)
- 14 Kensington lock attachment

## Primary/Secondary Functionality

A mainframe can only hold up to five modules, so for more complex setups a second mainframe is required. The BIT-2100B allows you to connect one mainframe (the “primary”) to your computer as usual, and connect another mainframe (the “secondary”) to the primary mainframe via USB, which allows you to control the relays of both mainframes with only one remote connection:

- connect both instruments to AC power
- connect the instrument you want to use as a primary to your PC as usual
  - you can use USB or LAN; see next sections
- connect the two instruments using a USB A-to-B cable:
  - use the type B (rectangular) connector on the primary instrument
  - use the type A (square shaped) connector on the secondary instrument

Such a connection is shown in figure 3.



**Figure 3: How to connect a primary/secondary configuration**

In this setup, even though your computer is only connected to the primary mainframe, the relays in the secondary frame can be controlled via the remote commands (page 23) sent to the primary mainframe. The primary/secondary feature focuses on relay control, not the entire functionality; it is explicitly noted in this manual when a software feature includes secondary mainframes.

Connecting more than one secondary to a primary, or daisy-chaining multiple secondary, is *not* supported.

It is *not* recommended to establish another VISA connection via LAN to the secondary while it is in a primary/secondary configuration, as it might interfere with the communication between the primary and the secondary.

The current status of the primary/secondary configuration is also shown on the display, see page 11.

## **Cleaning**

To clean any part of the BIT-2100 Series Switch System or its components, first turn it off and remove all connections, including AC power, then clean it only use a dry or damp cloth.

## **Disposal**

All electric and electronic equipment are required to be separated from normal waste for disposal.

## 5. General Operation

### Preface

The following instructions and feature descriptions apply to a BIT-2100 Series Switch System with firmware revision 2.00 or later.

### VISA Connection

The BIT-2100 is intended to be programmed by sending SCPI command through a VISA connection.

To establish a VISA connection and start sending remote commands, you need to conduct the following steps:

- install appropriate software on your PC (see section “Keysight IO Libraries”)
- connect the BIT-2100 to your PC, either through a network (see page 15) or via USB (see page 19)
- obtain the instrument’s VISA address (also explained in the corresponding sections mentioned in previous point)
- connect to the instrument and send SCPI commands (see page 23)

#### **Keysight IO Libraries**

In order to have all necessary drivers and software, it is recommended to install the Keysight IO Libraries Suite. This manual uses the Keysight IO Libraries Suite 2018 as a reference.

### Network Connection

You can establish a VISA socket or VXI-11 connection via the network.

Additionally, the instrument will create a locally accessible web interface (see page 17), which provides information about the instrument, and allows to conduct firmware updates and system upgrades (see page 20).

#### **Connecting to LAN**

Connecting to a BIT-2100 can simply be done via Ethernet. You can connect over a company network, or directly to a PC. You can use cross-link cables as well as regular cables.

The instrument has two Ethernet connectors, labeled “1” and “2”. The default configurations are:

- left connector, “1”: static IP 192.168.5.100/24
- right connector, “2”: DHCP and auto-IP

The current IP address will be shown on the instrument’s display.

Most software applications will allow you to use a host name instead of an IP address, which might be more convenient for the user. Each BIT-2100 unit has an individual host name with the format “bit2100-serial.local”, where “serial” is the serial number that you can find on the type plate of the instrument (see also figure 12). For example, if your instrument has serial number DE000042, you use the host name “bit2100-de000042.local” to access the instrument via a web browser or VISA connection.

The Ethernet configuration of each port can be changed, either using the web interface (see page 17), or remote commands (see page 34). You can bring the Ethernet connectors back into their default configuration by pressing and holding the “LAN Reset” push-button at the front side of the instrument for at least five seconds. The reset will be indicated on the display.

## Keysight Connection Expert

The instrument should also be found by the Keysight Connection Expert (which is part of the Keysight IO Libraries Suite), as shown in Figure 4.

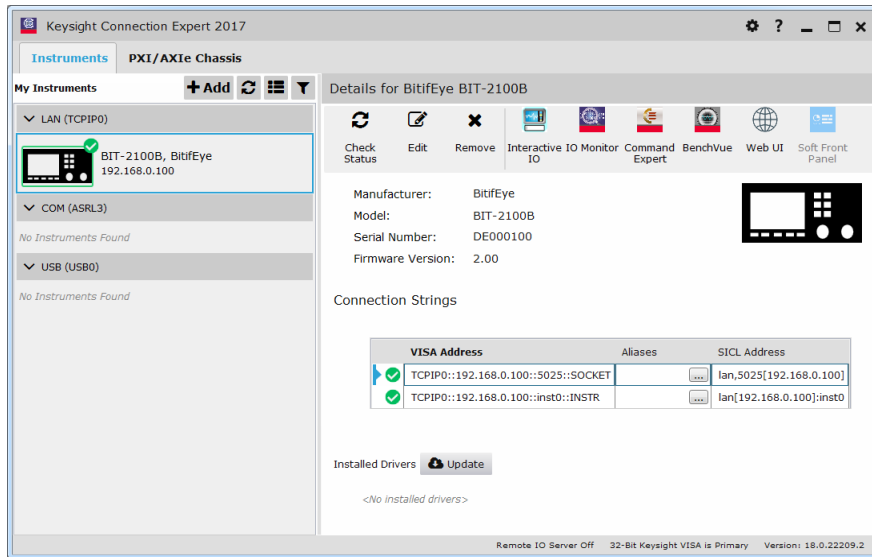


Figure 4: The BIT-2100 shown in the Keysight C Expert

With Keysight Connection Expert 2022, the steps to add a BIT-2100 to the “My Instruments” list are:

- make sure the instrument is connected to the network properly
- in the Keysight Connection Expert, click “+ Add”, then “LAN instrument”
- select your BIT-2100 unit from the list, click “OK”

Once it is in the “My Instruments” list, you can select it, and determine the VISA address, send SCPI commands, or access the instrument web interface.

The VISA address can either be for the socket protocol, or for the VXI-11 protocol:

- Example for a socket address: “TCPIP::192.168.0.100::5025::SOCKET”
- Example for a VXI-11 address: “TCPIP::192.168.0.100::inst0::INSTR”

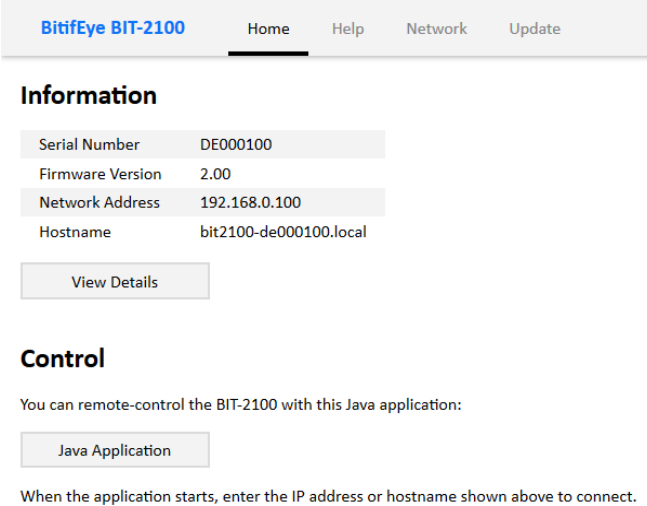
Most applications should work with either of the two protocols. You can also replace the IP that is encoded in the VISA address with the host name of the instrument.



## Accessing the Web Interface

The web interface of the BIT-2100 can be accessed from every web browser in the local network by entering the device's IP address or host name in the browser's address bar.

To access the web interface, open a new tab in your web browser, and enter the address (either IP or hostname). Make sure your browser does not auto-complete anything after the address (e.g. if your browser auto-completes the IP to "http://192.168.0.100/update.htm", remove the "/update.htm" before hitting enter).



The screenshot shows the BitifEye BIT-2100 web interface. At the top, there is a navigation bar with the title "BitifEye BIT-2100" and four menu items: "Home", "Help", "Network", and "Update". The "Home" menu item is currently selected. Below the navigation bar, the "Information" section is displayed, containing a table with the following data:

Serial Number	DE000100
Firmware Version	2.00
Network Address	192.168.0.100
Hostname	bit2100-de000100.local

Below the table is a "View Details" button. The "Control" section follows, with the text "You can remote-control the BIT-2100 with this Java application:" and a "Java Application" button. Below the button, it says "When the application starts, enter the IP address or hostname shown above to connect."

**Figure 5: Instrument Web Interface**

Note that JavaScript is required for proper display of the web interface.

Items you can access from the web interface:

- Home: details about the instrument, e. g. serial numbers, firmware version, etc., and access to the Java GUI (see next section)
- Network: change network settings, identify instrument (by making the display blink)
- Help: download user manual and data sheet, VISA addresses, software licenses
- Update: firmware update and system upgrade, see also page 20

## Using the Java GUI

On the start page of the web interface, you can download a .jar-file to your computer. If you have Java installed, you can execute this .jar-file to start the GUI.

The GUI will first show a dialog to prompt you to enter the IP address or hostname of the instrument (see figure 6). If you cancel this dialog, the GUI will operate in offline mode, where it simulates the behavior of a typical instrument.

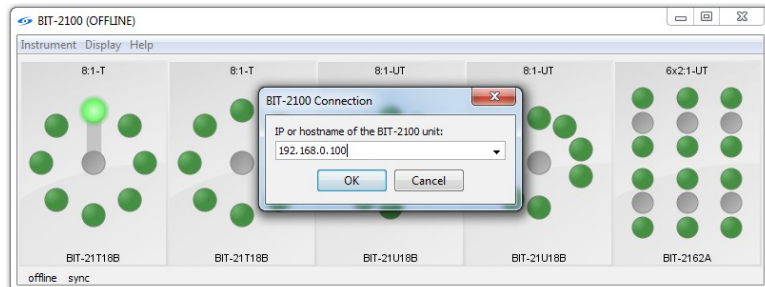


Figure 6: Connecting to the instrument with the GUI

When being connected to the instrument, you can click the circles which represent the connectors and LEDs of the switch modules (see figure 7). You can click the type labels on the modules to get details about the specific module. If the connected instrument is a primary, you will see tabs on the right of the window where you can switch to the secondary.

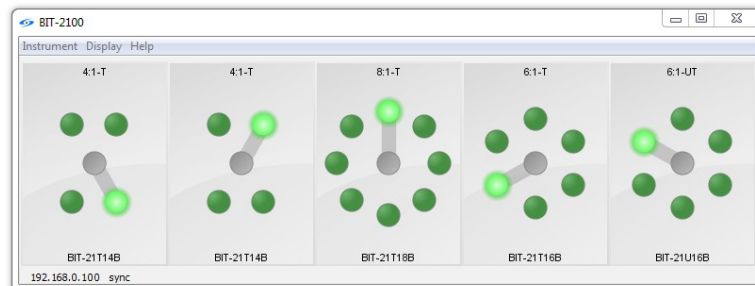


Figure 7: Main window of the GUI

You can access more features from the main menu, including:

- Instrument: connect/disconnect; get more information about the instrument; initiate a reset; initiate a self-test; open the diagnostics window
- Display: enable/disable synchronization of the GUI with the instrument, which might be helpful with slow connections; enable/disable GUI lock, which might be helpful if you don't want to accidentally interfere with another remote connection
- Help: get more information about the GUI

In the diagnostic window, you can inspect detailed error messages, issue your own SCPI commands, or turn on logging to learn how the GUI uses SCPI commands.

The 16:1 modules (terminated/unterminated) are currently not supported in the Java GUI.

## USB Connection

The BIT-2100 can be controlled from any PC via a USB connection.

Note that there are two USB connectors on the rear side of the instrument. Only the type B (square shaped) connector can be used to establish a remote connection.

The type A (rectangular) connector can be used for system upgrades (see page 21). The two connectors together can also be used for primary/secondary connections (see page 13).

### Connecting to USB

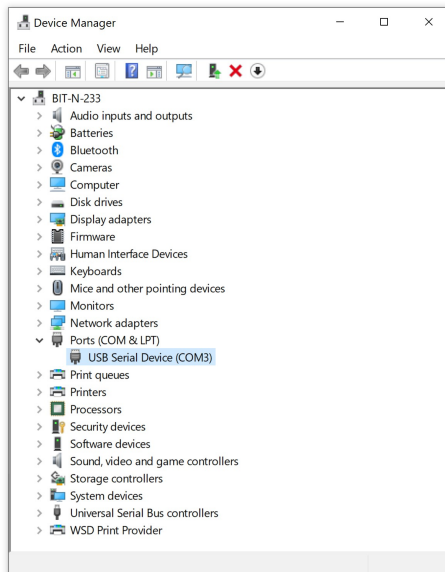
Connecting only requires a standard USB type A to USB type B connector cable.

When a BIT-2100 is connected to a PC for the first time, drivers are required. The BIT-2100 registers as a virtual serial port device. This driver is available from the BitifEye website.

### Determining the VISA address

The VISA address is typically “ASRL#:INSTR”, where “#” is the COM port address. The Keysight Connection Expert will automatically show an item for this port under “My Instruments”. Click “+ Add”, “Serial Instrument on ASRL *portnumber*” to connect to the instrument.

If multiple ASRL ports (i.e. COM ports) are shown, you can determine the COM port address by opening the “device manager” in the Windows computer settings (see Figure 8). When the proper drivers are installed, one of the listed COM ports will be shown as “BitifEye BIT-2100”, otherwise it will show up as something like “”.



**Figure 8: Virtual COM Port Address Shown in Device Manager**

The Keysight Connection Expert will show a VISA address for the instrument, e.g. “ASRL10:INSTR”. With this VISA address, a VISA connection can be established to send SCPI commands.

## 6. Software Updates

There are two kinds of software updates for the BIT-2100:

- A “Firmware Update” provides the latest firmware to the instrument, providing latest bug-fixes and features
- A “System Upgrade” provides a new operating system for the instrument, which should rarely be necessary

To install a firmware update or a system upgrade, click the “Update” button on the menu panel of the web interface.

### CAUTION

Only use appropriate update/upgrade files from BitifEye. Do not disconnect power during the update process.

### Firmware Update

For a firmware update, an update file with the extension “.fwu”, furnished by BitifEye, is required. Follow these steps:

- enter the update page on the instrument web interface
- under “Firmware Update”, select the .fwu-file, then click “Update”
  - make sure you only use .fwu-files that are intended for the use with a BitifEye BIT-2100B
- wait until the update is complete

Note that the instrument will be reset during this process, and all remote connections will be dropped.

During the firmware update, the instrument’s display will show something like this (except during the first update after a system upgrade):

```
Updating...  
192.168.0.100
```

Once the update is complete, you will see the regular message from the firmware:

```
BIT-2100  
192.168.0.100
```

The web interface will keep updating and shows the progress of the update. Once the web interface reports that the update was successful, you can continue using your instrument as normal. No power-cycle is required.

## System Upgrade

System upgrades allow new features to be provided that affect the underlying operating system that runs on the CPU of the BIT-2100. A system upgrade file with the file name “image.stream”, furnished by BitifEye, and a USB thumb drive, is required

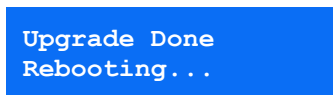
Follow these steps:

- copy the file “image.stream” on a USB thumb drive
  - the USB thumb drive must contain exactly one partition which is formatted with FAT32 or ext4
  - the file must be in the top-level directory, with the filename unchanged
  - make sure you only use image.stream files that are intended for the use with a BitifEye BIT-2100B
- plug the USB thumb drive into the USB type A connector at the rear side of the instrument
- if the instrument web interface is accessible:
  - enter the update page on the instrument web interface
  - under “System Upgrade”, click Upgrade
  - follow the instructions on the screen
- otherwise:
  - turn the instrument off
  - press and hold the LAN reset button, and turn the instrument on
  - hold the LAN reset button until the display indicates that the system upgrade is in progress
- the instrument will reboot into a special upgrade system; you can follow the progress on the instrument’s display, as the instrument web interface won’t be available during this process
- once the upgrade is done, the instrument will reboot into the freshly upgraded operating system

During the system upgrade, the instrument’s display will show something like this:



Once the update is complete, it should show something like this:



Note that as a next step, a firmware update is required; this will also be shown on the display:



```
FW-Upd. !  
192.168.0.100
```

To access the web interface, open a new tab in your web browser, and enter the IP address. Make sure your browser does not auto-complete anything after the address (e.g. if your browser auto-completes the IP to “http://192.168.0.100/update.htm”, remove the “/update.htm” before hitting enter).

Note that the network interface is reset during a system upgrade. Due to this, depending on your network infrastructure, using the hostname might not work immediately after a system upgrade. If this happens, it is recommended to use the IP address instead of the hostname to avoid trouble.

Now follow the instructions in the previous section (page 20) to conduct the firmware update.

## If anything goes wrong

If a firmware update goes wrong, it is recommended that you read the displayed errors carefully. If possible, try to carry out the firmware update again. In the case of a failure, please send the reported errors to BitifEye support.

If a system upgrade fails, or if the system becomes unusable after a failed firmware update, you can repeat the system upgrade any time.

In the case of a failure, please send the errors reported on the instrument display to BitifEye support.

## 7. Remote Programming

This section describes the commands that are available for remote programming. Any language/framework that is capable of handling such instructions can be used for programming the BIT-2100.

Alternatively, you can use the Keysight Interactive IO tool, which is part of the Keysight IO Libraries.

### General Syntax

SCPI commands consist of the following parts:

- a header, which consists of mnemonics
  - each mnemonic is preceded by a colon, where the first colon is optional
  - a mnemonic may require an integer index, which is indicated by a # symbol in the documentation
- the header can be followed by a question mark, making it a query
- zero or more parameters
  - if any parameters are given, the first argument is preceded by a white-space
  - if multiple parameters are given, they are separated by commas

There are also a few IEEE-488 commands, which start with an asterisk instead of a semicolon.

Multiple commands can be concatenated into a transaction, by separating them with semicolons. However, note that the leading colon for subsequent commands is not optional anymore: if it is omitted, it is assumed that the preceding mnemonics are the same as for the first command.

Mnemonics are documented with an upper-case part and a lower-case part. You can either use only the letters in upper-case, or all letters together. The instrument handles the mnemonics case-insensitive.

Note that in this document, explicit line breaks are printed in C-syntax: line-feed characters (ASCII code 10) are printed as `\n`, carriage-return characters (ASCII code 13) are printed as `\r`.

Note that optional parts of the SCPI syntax are documented in square brackets [ ], and parameters are documented in angle brackets < >. Parameter selections are documented with pipes | in between.

### Data Formats

The following argument types are supported:

- strings
- integers
- Boolean flags
- keywords

#### *Strings*

Strings are always quoted with double-quotes (").

Example: `"some string"`.

## Integers

Plain integers.

Examples:

- 42
- -10

## Boolean Flags

Boolean flags can only be 0 or 1; alternatively, “on” or “off”.

Examples:

- 0
- 1
- off
- on

## Keywords

A keyword is similar to a string, except that it is not put into quotes, and only certain values are allowed. The accepted set of keywords is different for each command.

When keywords are required as arguments, they can be given in either the short form or long form, just like mnemonics (e.g. `STATic` could also be abbreviated as `STAT`). When keywords are returned by a query, they are returned in the long form.

## Error Handling

Every command or query might cause an error. Reasons might be:

- the command is not valid (e.g. a typo)
- arguments are incorrect (e.g. missing arguments, too many arguments, invalid argument type)
- the operation is not valid (e.g. a command that refers to a non-existing module)
- hardware failure (e.g. relay is stuck)

Any errors are pushed into a queue, which can be queried with the `:SYST:ERR?` query (see page 26). Upon an error, a query might return an invalid response, but a command (which has no response) silently fails. Therefore, it is strongly recommended to query errors after every command or query. See also page for 36 best practices.

## Module Numbering

A mainframe has five slots where modules can be mounted. The left-most slot is the 1<sup>st</sup> slot, the right-most slot (next to the display and power-switch) is the 5<sup>th</sup> slot.

Note that the SCPI commands use a 0-based scheme, so the left-most slot is identified with index 0.

When a secondary mainframe is connected, it can be accessed as if it were the 6<sup>th</sup> through 10<sup>th</sup> slot (i.e. 0-based indexes 5 through 9).



## IEEE-488 Commands

The following list explains all available IEEE-488 commands. Note that these commands must start with an asterisk (unlike regular SCPI commands), as shown in the syntax descriptions and in the examples.

### **\*IDN?**

Syntax: `*IDN?`

Returns the instrument's manufacturer, product number, serial number and firmware version. The values are separated by commas.

The manufacturer is "BitifEye", the product number is "BIT-2100B". The serial number consists of letters and digits. The firmware version consists of the major revision, a dot, and the minor revision.

Example query: `*IDN?`

Example response: `BitifEye,BIT-2100B,DE0000001,0.10`

### **\*RST**

Syntax: `*RST`

Resets the device into default state (the same state as after power-up).

In a primary/secondary configuration, the secondary frame will also be reset.

Example command: `*RST`

### **\*TST?**

Syntax: `*TST?`

Conducts a self-test of all testable components of the instrument. The tests are the same tests that are conducted when the `:SYST:SELF?` Query (see page 26) is issued.

In a primary/secondary configuration, the secondary frame will also be tested.

After the self-tests are complete, which might take a while, a numeric value is returned. If all self-tests pass, `0` is returned; otherwise, the number of errors is returned.

Note that the self-tests will take a while to complete, depending on how many and which modules are installed. You'll probably receive a time-out from your VISA client, unless the VISA timeout is programmed to 30 seconds or more.

If errors were detected, it is recommended to use the `:SYST:SELF?` query to get a human-readable report.

Example query: `*TST?`

Example response: `0`

## Common Commands

### **:SYSTem:ERRor?**

Syntax: `:SYSTem:ERRor?`

Returns the first error in the error queue. The error will be removed from the error queue.

The error message consists of an error code (integer) and an error message. When no error is in the queue, the error code is zero and the message is "No Error". The code and the message are separated by a comma.

Example query: `SYST:ERR?`

Example response: `0, "No Error"`

### **:SYSTem:ERRor:COUNT?**

Syntax: `:SYSTem:ERRor:COUNT?`

Returns the number of errors in the error queue. The error queue will not be modified.

Example query: `SYST:ERR:COUN?`

Example response: `0`

### **:SYSTem:HELP:HEADers?**

Syntax: `:SYSTem:HELP:HEADers?`

Returns a list of all supported SCPI headers, separated by carriage-return characters ("`\r`", see also page 23).

Example query: `YST:HELP:HEAD?`

Example response: `"*IDN\r*RST\r:SYSTem:ERRor?\r:SYSTem:HELP:HEADers?"`

The actual response will contain many more headers. Depending on the software that you use, the "`\r`" character might be displayed as a new line or not.

### **:SYSTem:SELFtest?**

Syntax: `:SYSTem:SELFtest?`

Conducts a self-test of all testable components of the instrument. The tests are the same tests that are conducted when the `*TST?` Query (see page 25) is issued.

In a primary/secondary configuration, the secondary frame will also be tested.

After the self-tests are complete, which might take a while, a string is returned. If all self-tests pass, the string is `"pass"`; otherwise, the string starts with `"fail"`, followed by a human-readable description of the detected problems.

Note that the self-tests will take a while to complete, depending on how many and which modules are installed. You'll probably receive a time-out from your VISA client, unless the VISA timeout is programmed to 30 seconds or more.

Example query: `SYST:SELFtest?`

Example response: `"pass"`

### **:SYSTem:CONFiguration?**

Syntax: `:SYSTem:CONFiguration?`

Returns a list of modules that are currently mounted into the mainframe. If a secondary mainframe (page 13) is connected, the modules of the secondary mainframe are included as well.

The returned string contains a series of module descriptors, separated by “;”. Each module descriptor is formatted as follows:

- the slot number (0-based), followed by “=”
- the number of relays in the module, followed by “x”
- the number of paths of each relay, followed by “:1”
- if the relay supports path 0 as well (all terminals disconnected), a “\*” follows
- the termination of each relay is indicated as “-T” (terminated) or “-UT” (unterminated); see also page 9

Example query: `:SYST:CONF?`

Example response: `"0 = 1x6:1*-T; 3 = 4x2:1-UT"`

In this example, in the first slot there is a module with a single 6:1 relays (which can also do path 0, i.e. disconnect all terminals from the common terminal), which is terminated (i.e. all disconnected terminals, except for the common terminal, are terminated); and another module in the 4<sup>th</sup> slot, which contains four relays, all 2:1, unterminated.

## Relay Control

These SCPI commands and queries allow to obtain information about modules and to control their relays.

In a primary/secondary configuration, all commands listed in this section affect both the primary and the secondary mainframe.

Access to modules with relays is provided with the `:RELAy` branch of the SCPI tree. Access to the individual relays of such a module is provided in the sub-branch `:SWITCh` (to avoid confusion with the existing "relay" branch).

### Module Identifier Strings

Some of the SCPI commands act on a module, and require a *module identifier string* to identify that module.

A module can either be identified by its index, or by its slot index:

- `"<index>"`: the index of the module (0-based), regardless in which slot it is mounted (i.e. empty slots are not counted)
- `"<slot>!"`: the index of the slot (0-based) in which the module is mounted; secondary frames start at 5

For some examples of module identifier strings, see figure 9.

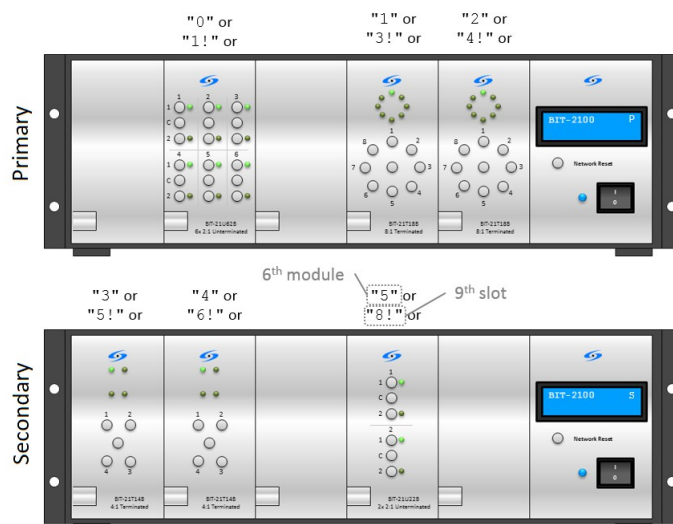


Figure 9: Module identifier string examples

`"5!"`: the module in the 1<sup>st</sup> slot of the secondary frame

Example for the usage of module identifier strings:

- `:REL:SER? "2!"` returns the serial number of the module in the 3<sup>rd</sup> slot (see also page 30)

If an identifier is provided that does not match any module, an error is caused.

## Relay Identifier Strings

Some of the SCPI commands act on a module, and require a *module identifier string* to identify that module.

A relay can either be identified by its index, or by a module identifier, followed its index within a module:

- "`<relayindex>`": the index of the relay (0-based), regardless in which module or slot it is mounted (i.e. empty slots are not counted)
- "`<moduleindex>.<relayindex>`": the index of the module (0-based), regardless in which slot it is mounted (i.e. empty slots are not counted), followed by a period, and the index of the relay (0-based) on that particular module
- "`<slotindex>!.<relayindex>`": the index of the slot (0-based) in which the module is mounted; secondary frames start at 5, followed by a period, and the index of the relay (0-based) on that particular module

For some examples of relay identifier strings, see figure 10.

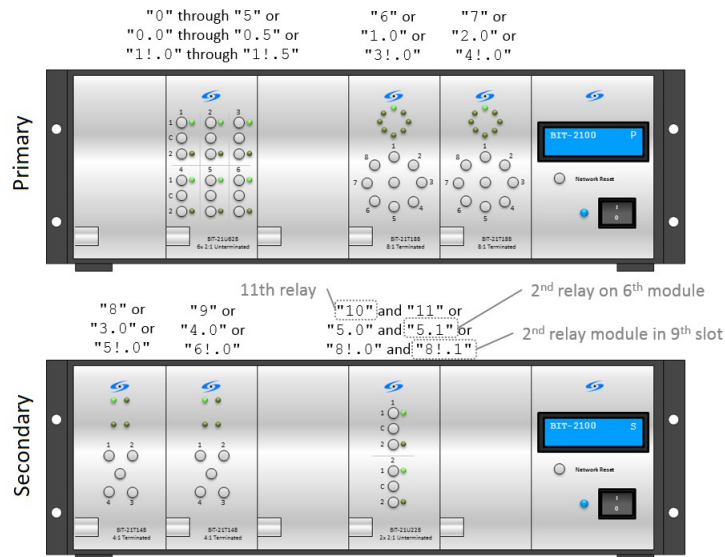


Figure 10: Relay identifier string examples

Example for the usage of relay identifier strings:

- `:REL:SWIT:NCYC? "2.2"` returns the switch cycle counter of the 3<sup>rd</sup> relay in the 3<sup>rd</sup> module (see also page 32)

If an identifier is provided that does not match any relay, an error is caused.

### **:RElay:COUNT?**

Syntax: `:RElay:COUNT?`

Returns the total number of modules, including the modules in secondary frames.

Example query: `REL:COUNT?`

Example response: `3`

### **:RElay:SLOT?**

Syntax: `:RElay:SLOT? <module_id>`

Returns the slot index (0-based) in which a module (identified by a module identifier string, see page 28) is mounted.

Example query: `REL:SLOT? "0"`

Example response: `1`

In this example, there is no module in the 1<sup>st</sup> slot, but one in the 2<sup>nd</sup> slot. The response of this query tells you that the first available relay switch module is in the 2<sup>nd</sup> slot (slot index 1).

### **:RElay:TYPE?**

Syntax: `:RElay:TYPE? <module_id>`

Returns the product number string of a module (identified by a module identifier string, see page 28) is mounted.

Example query: `REL:TYPE? "0"`

Example response: `"BIT-21T16B"`

### **:RElay:SERial?**

Syntax: `:RElay:SERial? <module_id>`

Returns the serial number string of a module (identified by a module identifier string, see page 28).

Example query: `REL:SER? "0"`

Example response: `"DE000042"`

### **:RElay:TERMinated?**

Syntax: `:RElay:TERMinated? <module_id>`

Returns a Boolean indicating whether the relays in the module (identified by a module identifier string, see page 28) are terminated (1) or unterminated (0).

Example query: `REL:TERM? "2!"`

Example response: `1`

### **:RElay:LATChing?**

Syntax: `:RElay:LATChing? <module_id>`

Returns a Boolean indicating whether the relays in the module (identified by a module identifier string, see page 28) are latching (1) or non-latching (0); see also page 9.

Example query: `REL:LATC? "1"`

Example response: `0`

### **:RElAy:PATH**

Syntax: `:RElAy:PATH <module_id>, <path>`

Changes the physical path of one or more relay in the module (identified by a module identifier string, see page 28).

If the module only has a single relay, the 2<sup>nd</sup> argument indicates the path the relay is put into. Path 1 is indicated by `1`, and so on. To disconnect all terminals from the common terminal (if supported), use `0`.

If the module has multiple relays (which must not support disconnecting all, the 2<sup>nd</sup> argument is a bit mask. The first bit is for the first relay, where a 0-bit is for path 1, and a 1-bit is for path 2.

For finer-granular access to the individual relays, use `:RElAy:SWITCh:PATH` (page 32).

Example command: `:REL:PATH "1", 3`

### **:RElAy:PATH?**

Syntax: `:RElAy:PATH? <module_id>`

Returns the currently applied path of the relays in the module (identified by a module identifier string, see page 28). The return value has the same semantics as explained in the `:RElAy:PATH` command (page 31).

For finer-granular access to the individual relays, use `:RElAy:SWITCh:PATH?` (page 32).

Example query: `:REL:PATH? "1"`

Example response: `3`

### **:RElAy:SWITCh:COUNT?**

Syntax: `:RElAy:SWITCh:COUNT? <module_id>`

Returns the number of relays in the module (identified by a module identifier string, see page 28).

Example query: `:REL:SWIT:COUNT? "1"`

Example response: `1`

### **:RElAy:SWITCh:TERMinated?**

Syntax: `:RElAy:SWITCh:TERMinated? <relay_id>`

Returns a Boolean indicating whether the relay (identified by a relay identifier string, see page 29) is terminated or unterminated. The semantics are the same as for `:RElAy:TERMinated?` (page 30). Since all relays in a module always have the same characteristics, there are no differences to that query.

Example query: `:REL:SWIT:TERM? "1!.0"`

Example response: `1`

### **:RElAy:SWITCh:LATChing?**

Syntax: `:RElAy:SWITCh:LATChing? <relay_id>`

Returns a Boolean indicating whether the relay (identified by a relay identifier string, see page 29) is latching or not. The semantics are the same as for `:RElAy:LATChing?` (page 30). Since all relays in a module always have the same characteristics, there are no differences to that query.

Example query: `:REL:SWITCh:LATC? "1.0"`

Example response: `0`

### ***:RElay:SWITCh:PATH***

Syntax: `:RElay:SWITCh:PATH <relay_id>, <path>`

Changes the physical relay path.

The 2<sup>nd</sup> argument indicates the path the relay (identified by a relay identifier string, see page 29) is put into. Path 1 is indicated by `1`, and so on. To disconnect all terminals from the common terminal (if supported), use `0`.

Example command: `:REL:SWIT:PATH "1.0", 3`

### ***:RElay:SWITCh:PATH?***

Syntax: `:RElay:SWITCh:PATH? <relay_id>`

Returns the currently applied path of the relay (identified by a relay identifier string, see page 29). The return value has the same semantics as explained in the `:RElay:SWITCh:PATH` command (page 32).

Example query: `:REL:SWIT:PATH? "1.0"`

Example response: `3`

### ***:RElay:SWITCh:SERial?***

Syntax: `:RElay:SWITCh:SERial? <relay_id>`

Returns the serial number of the relay (identified by a relay identifier string, see page 29) as a string. This is the serial number that the manufacturer assigned to the relay itself, not to be confused with the serial number that BitifEye assigned to the module as in `:RElay:SERial?` (page 30).

Example query: `:REL:SWIT:SER? "1.0"`

Example response: `"DE12345678"`

### ***:RElay:SWITCh:NCYCles?***

Syntax: `:RElay:SWITCh:NCYCles? <relay_id>`

Returns the value of the switch cycle counter of a relay (identified by a relay identifier string, see page 29).

Example query: `:REL:SWIT:NCYC? "1.0"`

Example response: `1234`



## Legacy Commands

The SCPI commands and queries `:CONF?`, `:PATH[?]` and `[[:SEL]:FRAM[?]]` of the BIT-2100A are kept for backward compatibility. Please refer to the user manual of the BIT-2100A for the description of these commands.

In a primary/secondary configuration, the legacy commands also affect the secondary frame.

## Network Configuration

The following SCPI commands control the two network interfaces of the instrument.

Note that there are two interfaces at the rear side of the instrument; the left connector is addressed with index 1, the right connector is addressed with index 2.

All addresses are IPv4.

### **:NETWork:RESet**

Syntax: `:NETWork:RESet`

Resets both network interfaces into their default configurations (see also page 15).

Note that when you issue this command over a network connection, the connection will obviously drop.

Example command: `:NETW:RES`

### **:NETWork:INTerface#:CONFigure**

Syntax: `:NETWork:INTerface#:CONFigure <STATic|DHCP>[, <ip>, <subnet>, ]<gateway>]`

Configures a specified network interface to either DHCP or a static IP.

If using a static IP, the IP, a subnet mask and a gateway must be specified. Either the IP and a subnet mask are specified separately, or an IP with a subnet suffix must be specified.

Note that when you configure the network connection that you use to issue this command, the connection will obviously drop.

Example command:

`:NETW:INT1:CONF STAT, "192.168.0.10/24", "192.168.0.1"`

### **:NETWork:INTerface#:CONFigure:MODE?**

Syntax: `:NETWork:INTerface#:CONFigure:MODE?`

Returns the configured mode (`STATic` or `DHCP`) of the network port.

Example query: `:NETW:INT1:CONF:MODE?`

Example response: `STATic`

### **:NETWork:INTerface#:CONFigure:IP?**

Syntax: `:NETWork:INTerface#:CONFigure:IP?`

Returns the configured IP of the network port.

Note that this IP is not necessarily the IP under which the instrument can be reached (e.g. when no DHCP address is assigned). To determine the actual IP, use the `:NETW:INT#:IP?` query.

Example query: `:NETW:INT1:CONF:IP?`

Example response: `"192.168.0.10"`

### ***:NETWork:INTerface#:CONFigure:SUBNetmask?***

Syntax: `:NETWork:INTerface#:CONFigure:SUBNetmask?`

Returns the configured subnet mask of the network port.

Example query: `:NETW:INT1:CONF:SUBN?`

Example response: `"255.255.255.0"`

### ***:NETWork:INTerface#:CONFigure:GATeway?***

Syntax: `:NETWork:INTerface#:CONFigure:GATeway?`

Returns the configured gateway IP of the network port.

Example query: `:NETW:INT1:CONF:GAT?`

Example response: `"192.168.0.1"`

### ***:NETWork:INTerface#:IP?***

Syntax: `:NETWork:INTerface#:IP?`

Returns the actual IP of the network port. If the network interface is inactive, the IP 0.0.0.0 is returned.

You can use this query for instance when you connect via USB and want to determine the IP address of the instrument (if you have physical access to the instrument, you can also just read the IP from the LC display).

Example query: `:NETW:INT1:IP?`

Example response: `"192.168.0.1"`

### ***:NETWork:IP?***

Syntax: `:NETWork:IP?`

Returns the actual IP of the network in general. If both network ports have a valid IP, the address of the first port is returned. If none of the network interfaces are active, the IP 0.0.0.0 is returned.

You can use this query for instance when you connect via USB and want to determine the IP address of the instrument (if you have physical access to the instrument, you can also just read the IP from the LC display).

Example query: `:NETW:IP?`

Example response: `"192.168.0.1"`

## Programming Examples

### ***Best Practice***

It is generally recommended that you query the error queue after every command or query. The advantages of this approach are:

- when an error occurs, it's easier to find out which command or query triggered the error
- it can be verified that all commands are actually executed (which is not the case with syntax or argument errors, for example)
- the remote control software is always in sync with the instrument's firmware, as the query delays the remote control software until the command is completely handled and checked

The following programming examples do not show any error queries, for the sake of brevity and readability. You are recommended to use the following pseudo-code to run the programming examples, or any remote programming:

- command:
  1. send command string plus termination character
  2. send error query string plus termination character
  3. read and handle response
  4. repeat steps 2–4 until no error is reported
- query:
  1. send query string plus termination character
  2. read and handle response
  3. send error query string plus termination character
  4. read and handle response
  5. repeat steps 3–5 until no error is reported

## Querying

Assume that you are using an instrument with the following configuration:

- 1<sup>st</sup> slot: 4:1, terminated
- 2<sup>nd</sup> slot: empty
- 3<sup>rd</sup> slot: 6:1, unterminated
- 4<sup>th</sup> slot: empty
- 5<sup>th</sup> slot: 2x 2:1, unterminated

If you query the configuration with the following command:

```
:SYST:CONF?
```

You will get the response `0 = 1x4:1*-T; 2 = 1x6:1*-UT; 4 = 2x2:1-UT.`

As noted earlier, it is recommended to query errors after every command or query, so the complete set of SCPI commands should be:

```
:SYST:CONF?  
:SYST:ERR?
```

For the sake of brevity, error queries will be omitted in subsequent examples.

## Switching

Assume you want to put the modules into following paths:

- 1<sup>st</sup> slot: path 2
- 3<sup>rd</sup> slot: path 0 (i.e. disconnect all terminals)
- 5<sup>th</sup> slot:
  - 1<sup>st</sup> relay: path 1
  - 2<sup>nd</sup> relay: path 2

In order to do so, you could use the following series of SCPI commands:

```
*RST  
:REL:SWIT:PATH "0!.0",2  
:REL:SWIT:PATH "2!.0",0  
:REL:SWIT:PATH "4!.0",1  
:REL:SWIT:PATH "4!.1",2
```

As mentioned earlier, querying errors after each command (i.e. 5 times total in this example) is recommended.

## 8. User Serviceable Parts

### Changing Fuses

**WARNING**

Before changing fuses, make sure the device is physically disconnected from AC power! Never operate the BIT-2100 with the fuses or the fuse carrier removed!

To change the fuses, follow these instructions:

1. Make sure the device is powered off and all cables are disconnected
2. Open the fuse carrier on the rear side of the instrument (see Figure 11)
  1. release the fuse carrier using a screwdriver
  2. pull the fuse carrier out
3. Replace both fuses with 1.6AT 250 V slow blow
4. Insert and close the fuse carrier properly (see Figure 12)

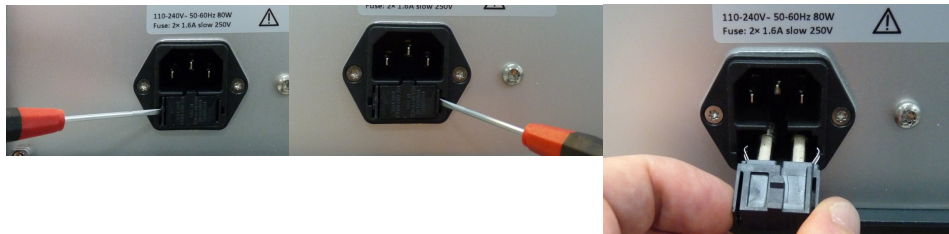


Figure 11: Removing Fuses



Figure 12: Inserting Fuses

## Dismounting Modules

### WARNING

Before dismounting modules, make sure the device is physically disconnected from AC power! Never operate the BIT-2100 with any module slot uncovered!

### CAUTION

Always use proper ESD (electrostatic discharge) protective equipment when preparing, mounting or dismounting modules. ESD can destroy the electronic components of the BIT-2100 Series Switch System.

To dismount a switch module or a filler panel, follow these instructions:

1. Make sure the device is powered off and all cables are disconnected
2. Dismount the two screws at the very top of the module and the two screws at the very bottom of the module (below the handle) using a Torx T8 screwdriver (see Figure 13)
3. Carefully pull the module out (see Figure 14)
  - Note that switch modules are in a red plastic rail, so you can only pull them out straight
4. Close the opened slot, either by a filler panel, or by another module

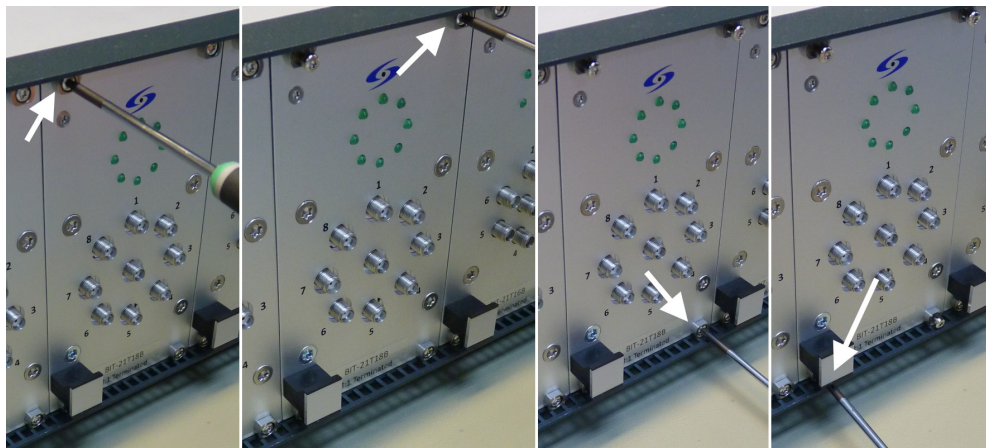


Figure 13: Removing screws from modules

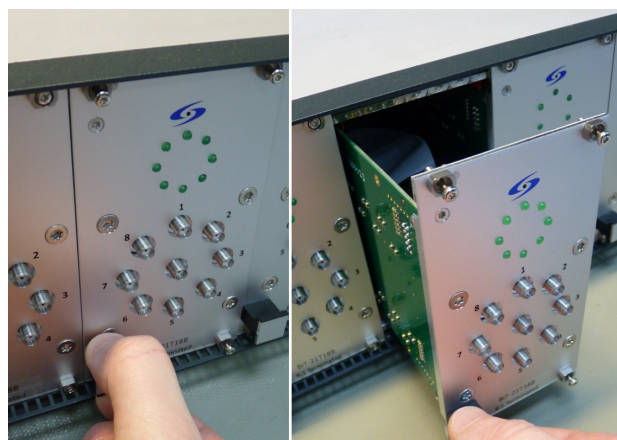


Figure 14: Removing a module

## Mounting Modules

### WARNING

Before mounting modules, make sure the device is physically disconnected from AC power! Never operate the BIT-2100 with any module slot uncovered!

### CAUTION

Always use proper ESD (electrostatic discharge) protective equipment when preparing, mounting or dismantling modules. ESD can destroy the electronic components of the BIT-2100 Series Switch System.

To mount a switch module or a filler panel, follow these instructions:

1. Make sure the device is powered off and all cables are disconnected
2. If a module or filler panel is still installed, remove it first (see page 39)
3. Carefully push the module into the slot (see Figure 15)
  - Make sure that the module is in the red plastic rail
  - Push in until the front panel of the module is flush with the others
4. Mount the two screws at the very top of the module and the two screws at the very bottom of the module (below the handle) using a Torx T8 screwdriver (reverse order of Figure 13)

Note that a firmware update might be required if the new module is not supported by the currently installed firmware.



Figure 15: Inserting a module



## 9. List of Acronyms

CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
ESD	Electrostatic Discharge
GUI	Graphical User Interface
IP	Internet Protocol
LAN	Local Area Network
LC(D)	Liquid Crystal (Display)
LED	Light-Emitting Diode
LSB	Least Significant Bit
LXI	LAN-based eXtensions for Instrumentation
MSB	Most-Significant Bit
PC	Personal Computer
RF	Radio Frequency
SCPI	Standard Commands for Programmable Instruments
SMA	Sub-Miniature A (Connector)
USB	Universal Serial Bus
VISA	Virtual Instrument Software Architecture