IO Ninja Hardware Manual

Table of Contents

Introduction

Last update: 09DEC2019

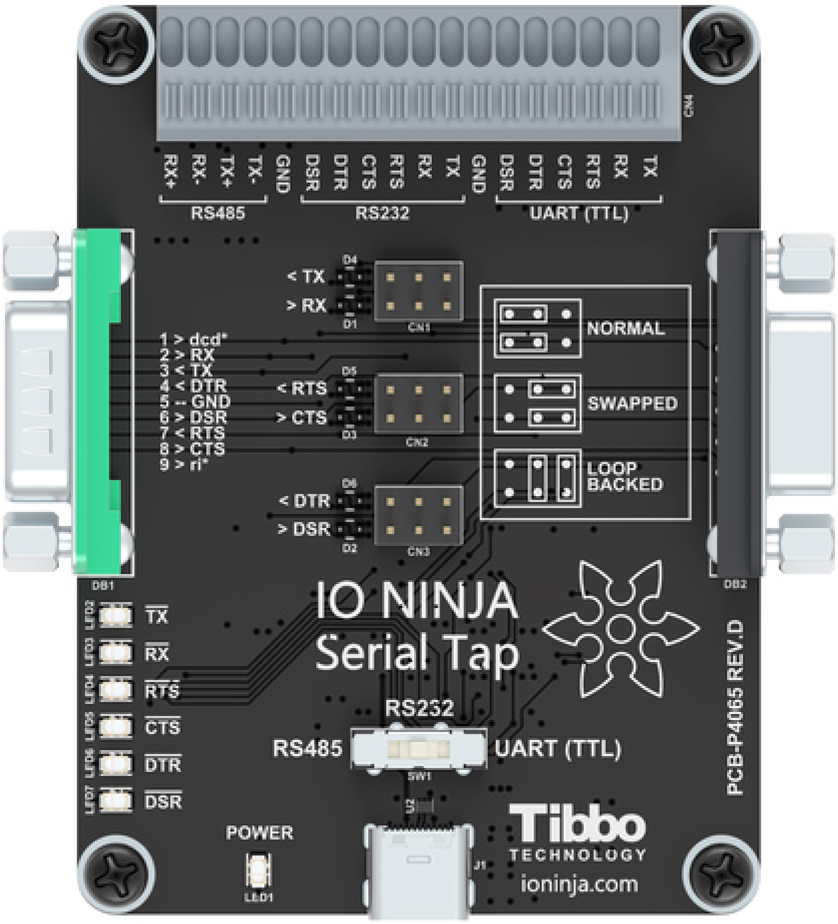
[Manual Update History](#)

This Manual covers IO Ninja-compatible hardware:

- [Serial Tap](#)
- [I2C/SPI Tap](#)
- [Ethernet Tap](#)

Other types of hardware sniffers will be available in the near future.

Serial Tap



Serial Tap is an affordable serial sniffer that can monitor RS232, RS485, and TTL-level UART communications. The Tap connects to your PC via a USB cable and is compatible with IO Ninja terminal/sniffer software.

Features

- The Serial Tap is designed for use with IO Ninja software (http://ioninja.com/).
- Monitors TX, RX, RTS, CTS, DTR, and DSR lines [Note 1] of serial ports.
- All inputs are connected to an 18-position, quick-release terminal block.
- Has three operating modes: RS232, RS485 [Note 2], and UART (TTL [Note 3]), selectable via a slide switch:
  - RS232 mode:
    - Two onboard DB9 connectors — male and female — offer true "cable wedge" sniffing [Note 4];
    - Six jumpers for swapping and loopbacking signals within RX/TX, RTS/CTS, and DTR/DSR pairs [Note 5];
    - Six bi-color LEDs indicate the status of monitored lines and allow distinguishing between the positive voltage (red), negative voltage (green), and zero voltage (off) states [Note 5];
    - DB9 connectors additionally pass through DCD and RI signals (without monitoring them).
  - RS485 mode allows to monitor TX and RX signal pairs.
  - UART (TTL) mode allows to monitor UARTs of IC chips with logic levels from 3.3V to 5V.
- Full-speed (12Mbps) USB2.0 interface on a USB-C connector.
- Six yellow LEDs indicate the state of six monitored lines as they enter the USB controller. LEDs turn on when the lines go LOW [Note 6].
- USB-powered, no additional external power necessary.
- Supplied with a USB-C cable and two DB9 gender changers.
- Compact, outside dimensions only 82 x 74 x 30 mm.

Note 1

A more accurate version of this statement would be: "Monitors two TX, two RTS, and two DTR lines," but that would be somewhat confusing to the reader. For clarity, we chose to list the lines as they are commonly known: TX, RX, RTS, CTS, DTR, and DSR. Keep in mind, however, that when it comes to sniffers, these names are relative and depend on the view. For two interconnected serial devices, a TX line on one end is an RX line on the other end. If so, which line is a "TX" and which line is an "RX" from the sniffer's point of view? Only you can answer this question. The Serial Tap itself has two inputs capable of monitoring serial data lines. Although they are both identical inputs, one is marked "TX" and another one is marked "RX." The same goes for other signals — RTS, CTS, DTR, and DSR. All these lines are inputs of the Tap. We gave them "regular" names to make the use of the Serial Tap intuitively easier. The same goes for RS485 lines. RX+/- and TX+/- pairs are both inputs of the Serial Tap.

This said, the Serial Tap incorporates two DB9 connectors, for which the distinction between TX and RX lines, RTS and CTS lines, and DTR and DSR lines is real. To give our terminology an anchored point of view, we have decided that the DB9-M connector on the left of the Tap is our "primary side" RS232 connector. For the entire board, all signal names are consistent with the standard pin assignment of this primary DB9-M connector. Of course, all "anchoring" disappears as soon as you abandon DB9s and plug into the terminal block lines.

Note 2

These inputs can also be used for monitoring RS422 communications.

Note 3

"TTL" here means "digital signals of microcontrollers, microprocessors, and other ICs." TTL inputs of the Serial Tap are also compatible with CMOS circuitry. Most importantly, "TTL" serial signals are usually inverted with respect to RS232 signals: a LOW (negative voltage) on an RS232 line corresponds to a HIGH level on a serial TTL (CMOS) line, and HIGH (positive voltage) on an RS232 line corresponds to a LOW level on a serial TTL (CMOS) line.

Note 4

TX, RX, RTS, CTS, DTR, and DSR lines of the primary DB9 connector located on the left of the Serial Tap are connected in parallel with RS232 inputs on the terminal block. This can be seen on the Block Diagram of the Serial Tap.

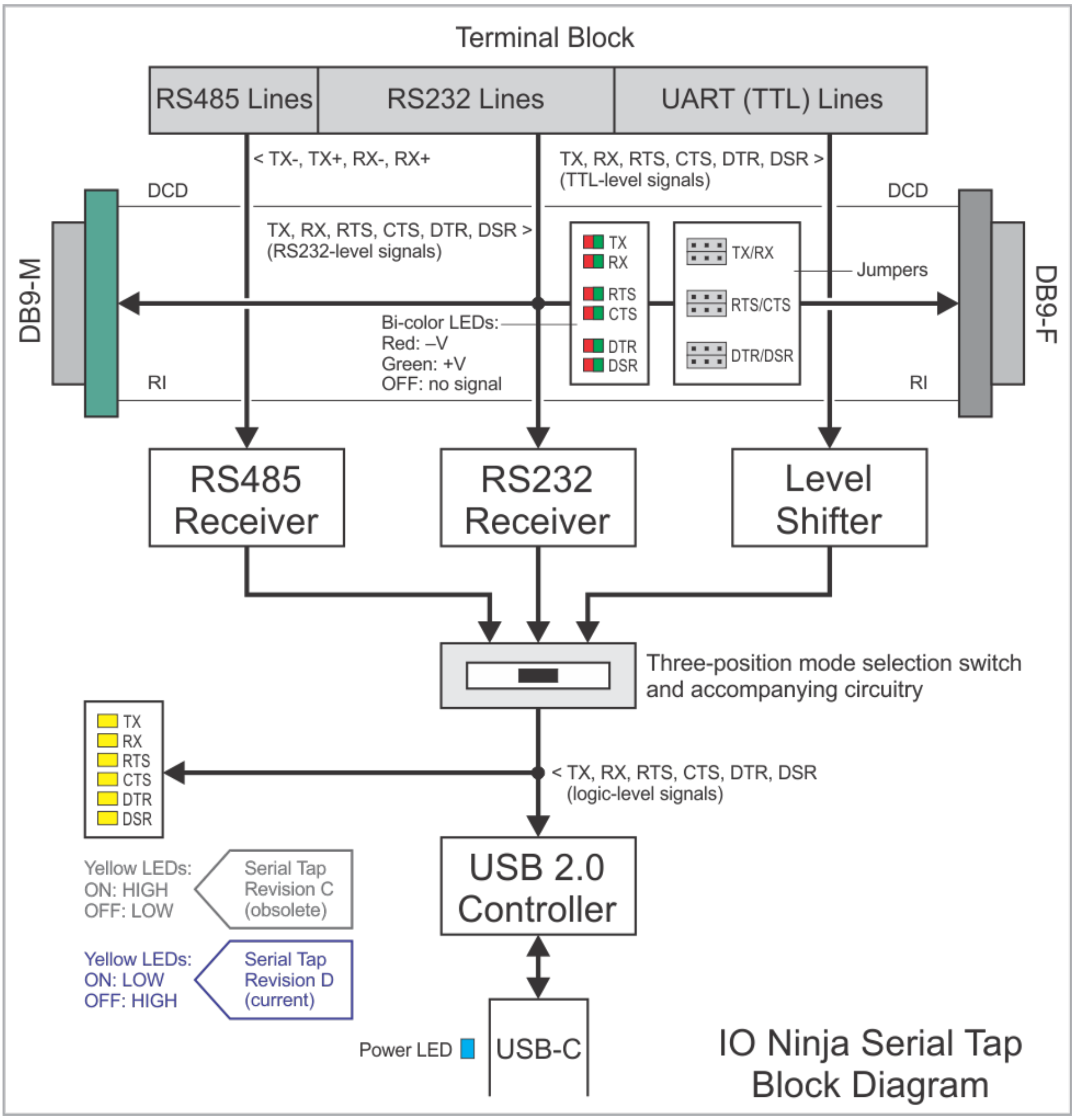
Note 5

Jumpers and bi-color LEDs on RS232 lines give the Serial Tap the second use as a simple mating and signal checking tool for serial devices.

Note 6

This statement pertains to the latest revision D of the product. On the earlier (and now obsolete) revision C, yellow LEDs used to turn on when the lines went HIGH.

Block Diagram

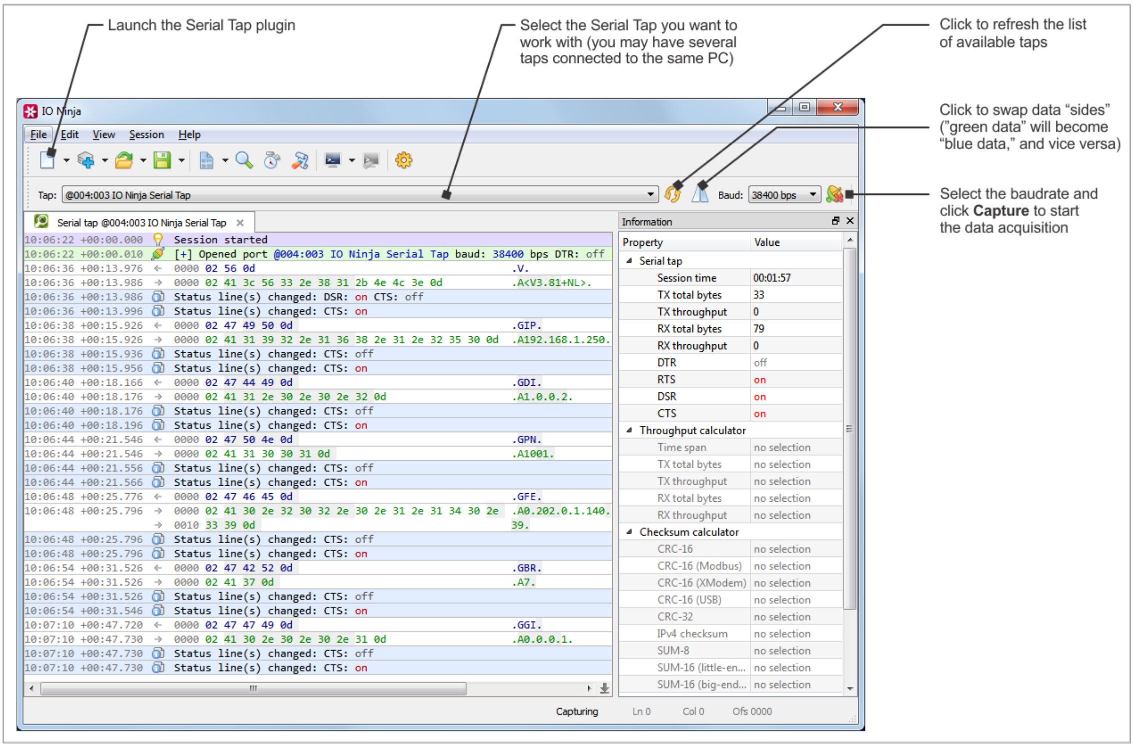


Using the Serial Tap

To start using the Serial Tap:

- Install and run IO Ninja software (http://ioninja.com/).
- On the Serial Tap, slide the mode selection switch into the desired position depending on what interface you will be using — RS485, RS232, or UART (TTL).
- Connect into the corresponding group of terminals on the terminal block. When in the RS232 mode, you can also employ wedge RS232 monitoring method.
- Plug the Serial Tap into the USB port of your PC.
- Launch the **Serial Tap** plugin.

- Click on the **Tap** drop-down and select the Serial Tap. Note: you can connect several Serial Taps to one PC. If necessary, refresh the list of available Taps.
- Set the baudrate and click the **Listen** button to start the data acquisition.



A note on the Serial Tap limitations

Please note that the Tap monitors serial traffic through two independent serial-over-USB channels. The first channel monitors TX, RTS, and DTR lines, while the second channel deals with RX, CTS, and DSR lines. Because these serial-over-USB channels are independent of each other, respective timing errors are always introduced on Windows and IO Ninja levels when receiving and recording the serial data and signal state changes.

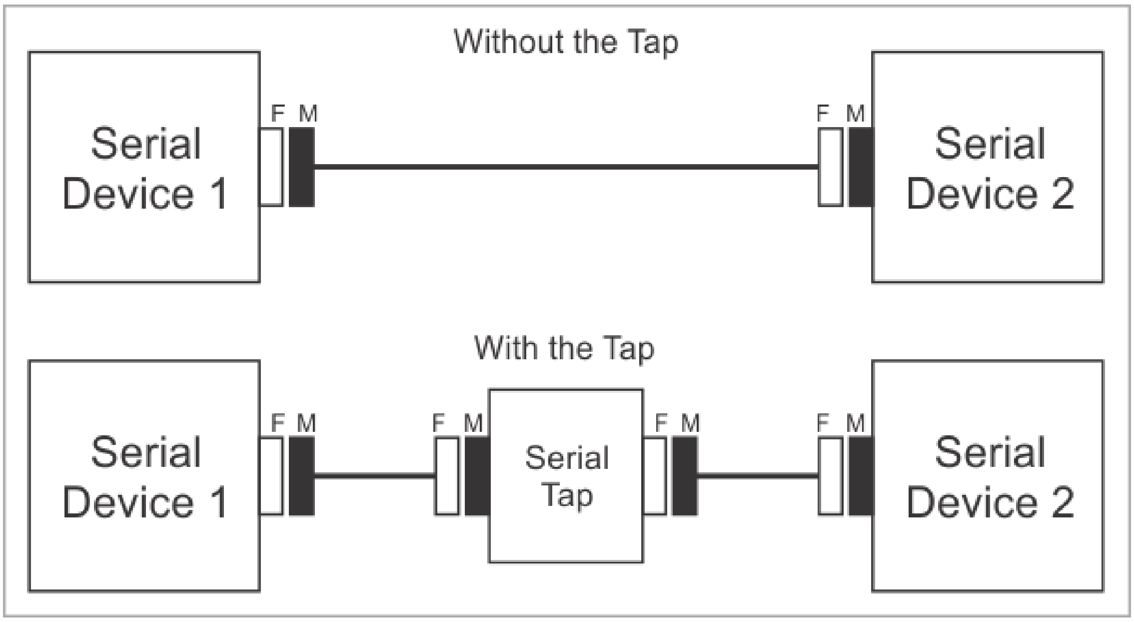
For example, let's suppose that you are monitoring serial communications between two interconnected serial devices and both devices have sent out some data at the same time. One of the USB channels will be luckier and get service first, while the other USB channel will experience a slight delay in service. IO Ninja, therefore, will show the data from one of the devices as having arrived first, and the data from the second device as having arrived second... while in reality, both devices have transmitted at the same time.

Wedge RS232 monitoring

The Serial Tap allows you to insert (wedge) it in between two RS232 devices. Here is how this is done.

Let's suppose that two serial devices are interconnected by a serial cable. Let's also suppose that the first device has a DB9-F connector, while the second device has a DB9-M connector. The serial cable is, therefore, of the M-to-F type.

To wedge the Serial Tap between these two devices, you will need the second M-to-F cable:



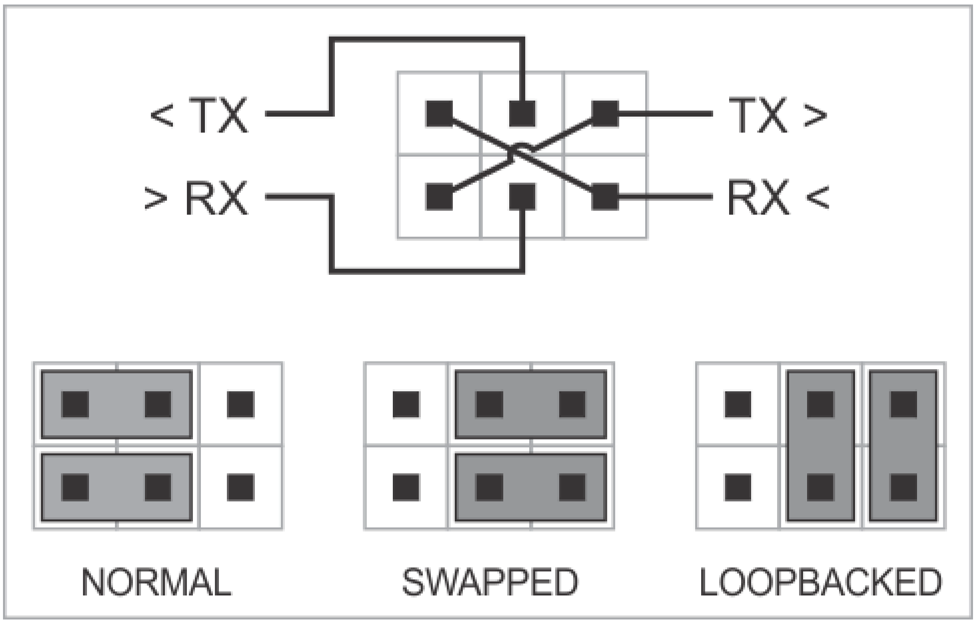
Of course, DB9 genders on the serial devices may differ from the above example, and so your serial arrangement may be different. To aid you in the "wedging process," each Serial Tap comes with two DB9 gender changers.

Jumper pairs

The Serial Tap has an additional useful feature allowing you to swap and loopback the signals in TX/RX, RTS/CTS, and DTR/DSR signal pairs. To achieve this, two jumpers are provided for each of the three pairs. There are three standard jumper configurations:

- **Normal** — In this position, the lines are arranged in such a way that wedging the Tap between the serial devices does not change anything. Meaning, TX on one end goes to RX on another end, and vice versa.
- **Swapped** — This swaps signals in a pair. Meaning, TX goes to TX, and RX goes to RX.
- **Loopbacked** — Both serial devices "receive back" their own signals. Meaning, the TX line on each side "comes back" through the RX line.

The following diagram illustrates the jumper arrangements. The diagram shows the jumpers for the TX and RX signal pair. RTS/CTS and DTR/DSR jumpers work in the same way.



Specifications

The Serial Tap is supplied with a USB cable and two DB9 gender changers.

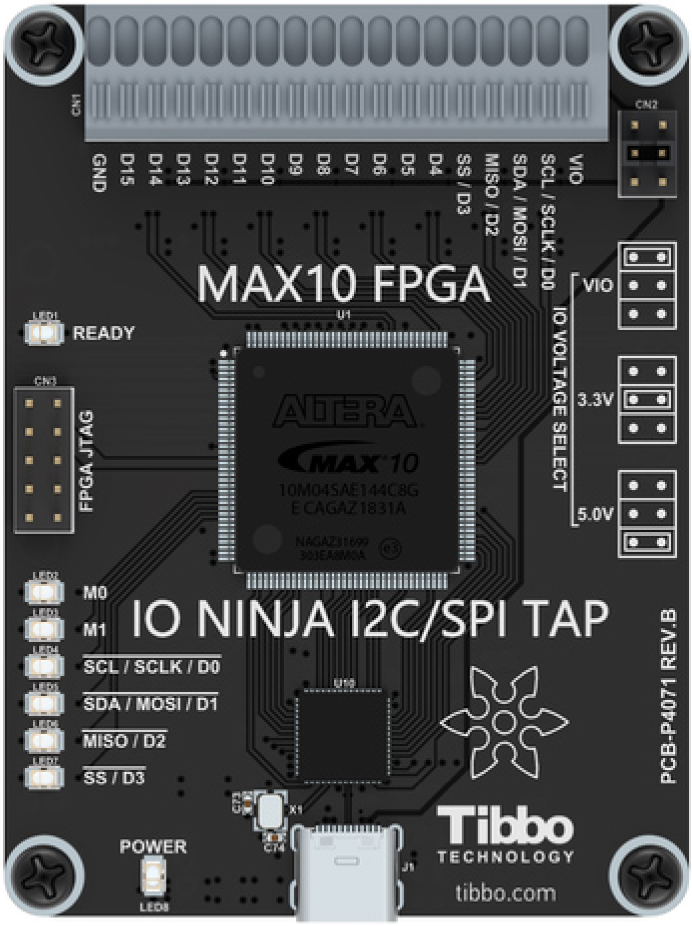
Hardware specifications

USB port	USB 2.0, full-speed (12MHz), USB-C connector
Maximum baudrate	2,764,800 bps
Operating temperature	0 to +60 degrees C
Operating relative humidity	10-90%
Mechanical dimensions	82 x 74 x 30 mm

All specifications are subject to change without notice and are for reference only. Tibbo assumes no responsibility for any errors in this Manual and does not make any commitment to update the information contained herein.

Is the data you are looking for missing from the above table? Do not just assume that you know the answer — talk to Tibbo! Remember, that the ultimate responsibility for all decisions you make regarding the use and the mode of use of Tibbo products lies with you, our Customer.

I2C/SPI Tap



I2C/SPI Tap is an affordable sniffer that can monitor I2C and SPI communications. The Tap connects to your PC via a USB cable and is compatible with IO Ninja terminal/sniffer software.

Features

- Designed for use with IO Ninja software (http://ioninja.com/).
- Reliably captures I2C traffic at clock speeds of up to 700KHz and SPI traffic at clock speeds of up to 24MHz [Note 1]; all four SPI modes are supported.
- Incorporates an I2C and SPI test sequence generator for self-testing the Tap.
- Based on Intel MAX10 FPGA.
- Has 16 buffered IO lines D0~15 connected to a quick-release terminal block.
- All buffers act as level shifters allowing the Tap to work with logical signals in the 1.8V-5V range.
- Eight LEDs onboard:
  - Green Ready LED;

- ☐ Two general-purpose green LEDs M0 and M1 for mode indication;
- ☐ Four yellow LEDs for indicating the state of IO lines D0~3 (sufficient for monitoring I2C and SPI communications);
- ☐ Blue Power LED.

- Built-in test generator outputs fixed I2C and SPI data sequences.
- High-speed (480Mbps) USB2.0 interface on a USB-C connector.
- Supplied with a USB-C cable.
- USB-powered, no additional external power necessary.
- Compact, outside dimensions only 82 x 74 x 30 mm.
- Product functionality may be expanded in the future:
  - ☐ Onboard FPGA is reconfigurable via the USB interface;
  - ☐ Sixteen IO lines provide ample room to grow;
  - ☐ It is envisioned that the product's abilities may be expanded to handle such protocols as...
    - Quad-SPI;
    - SDIO;
    - Single-wire bus;
    - and more!

Note 1

This number depends on the host PC's performance and its "workload." Captured data may be lost if a PC's USB channel is unable to receive it in a timely manner. Our tests have shown that an average i7-based desktop PC (that is not performing other time-critical tasks) easily captures sustained SPI traffic with clock speeds up to 24MHz.

Using I2C/SPI Tap

Connecting I2C and SPI signals to the Tap

To monitor I2C traffic, connect your I2C bus to terminals D0 (SCL) and D1 (SDA) terminals.

To monitor SPI traffic, connect your SPI bus to terminals D0 (SCLK), D1 (MOSI), D2 (MISO), and D3 (SS).

Note that the Tap has a built-in generator of test signals.

Selecting the logic voltage levels

I2C/SPI Tap can work with logic voltage levels from 1.8V to 5V. For correct operation of the Tap, you must select correct reference voltage for the Tap's level shifters.

This is done using the IO VOLTAGE SELECT jumper. Two reference voltages — 3.3V and 5V — are provided by the Tap itself. Other logic levels require an external reference voltage applied to the VIO terminal. To select the VIO input, put the jumper into the VIO position and connect the VIO terminal to the external voltage source (typically, the VCC line of the device you are monitoring). For example, if you are tapping into the I2C bus of a 1.8V device, then connect the VIO terminal to the 1.8V power used by this device.

No SDA pull-up resistor on the Tap

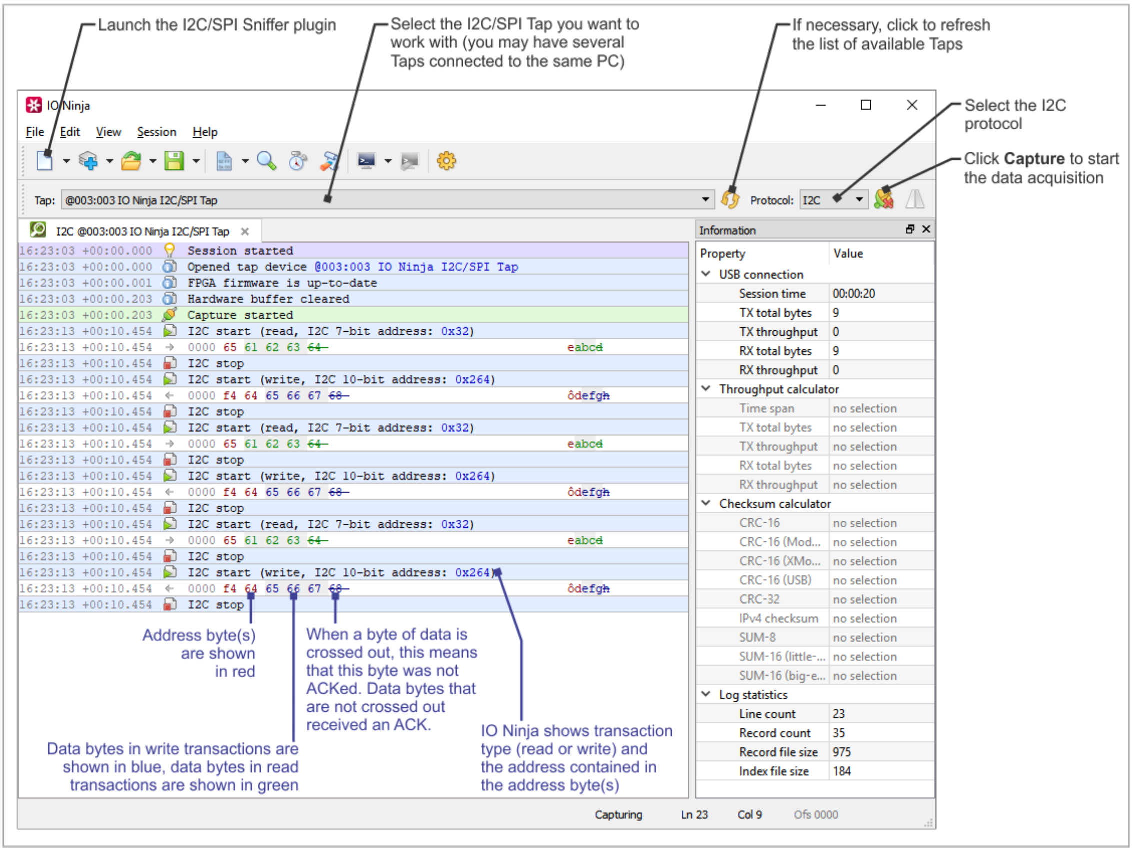
The SDA line of I2C interface requires a pull-up resistor. This Tap does not contain such a resistor and expects that the monitored device will have it.

Setting up IO Ninja

- Install and run IO Ninja (http://ioninja.com/).
- Plug the Serial Tap into the USB port of your PC.
- Launch the **I2C/SPI Tap** plugin.
- Click on the **Tap** drop-down and select the target I2C/SPI Tap. Note: you can connect several I2C/SPI Taps to one PC. If necessary, refresh the list of available Taps.
- Select the I2C or SPI protocol in the **Protocol** dropdown.
- In the case of SPI protocol, set the SPI mode, word length (SPI data bits), and endianness in the **Settings** dialog.
- Click the **Capture** button to start the data acquisition. At this moment, IO Ninja will check the Tap's configuration. If the Tap is configured for a different protocol (for example, for SPI while you need I2C), IO Ninja will reconfigure the Tap. This will only take a couple of seconds. Note that the Tap has two green LEDs M0 and M1. These LEDs indicate the Tap's configuration:
  - In the I2C mode, M0 is ON, and M1 is OFF;
  - In the SPI mode, M0 is OFF, and M1 is ON.
- When monitoring SPI traffic, you can also flip (swap) MISO and MOSI lines' data. From the Tap's point of view, MOSI and MISO are abstract labels. This is because the Tap is neither SPI slave nor SPI master. The tap is a passive observer, and so it is up to you to decide which of the two SPI data lines is MISO, and which is MOSI. Rather than physically swapping the wires connected to the Tap's terminal block, you can achieve the same result by clicking **Flip SPI MISO/MOSI**.

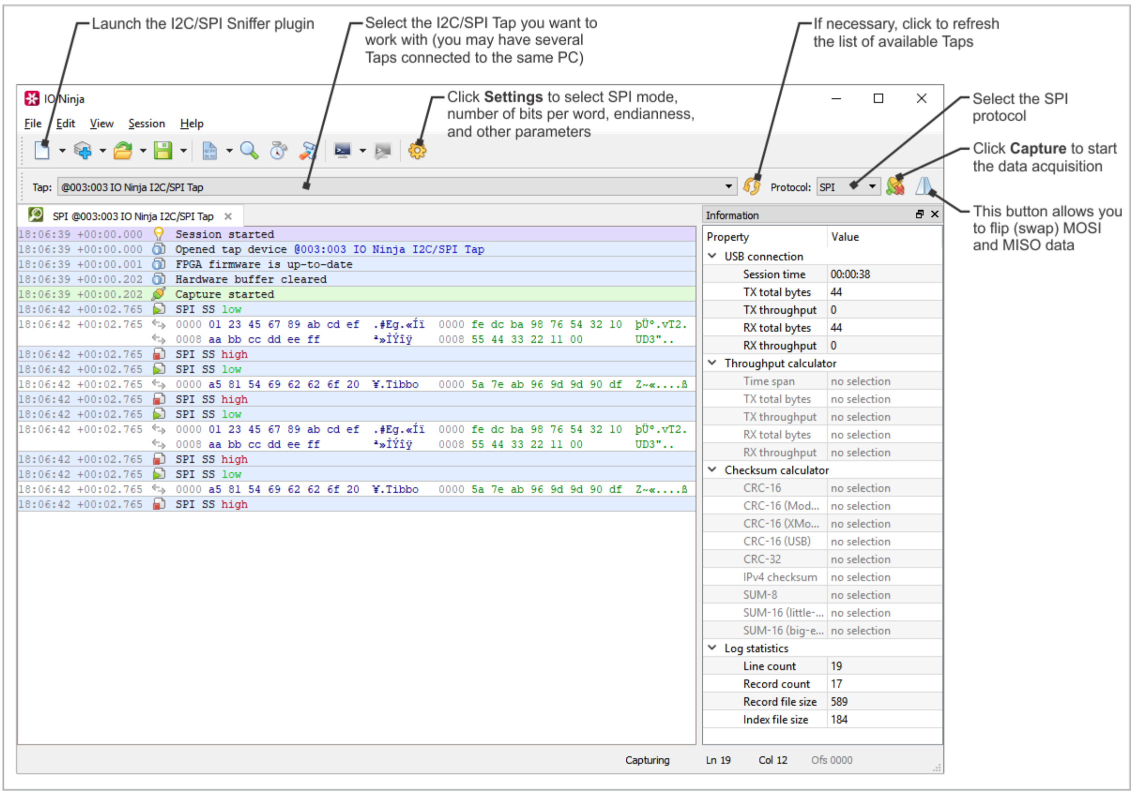
The above is illustrated by screenshots in I2C Mode Screenshot and SPI Mode Screenshots. The screenshot depicting IO Ninja in I2C mode also contains important comments on Ninja's representation of I2C traffic, specifically, how unacknowledged (unACKed) bytes are shown.

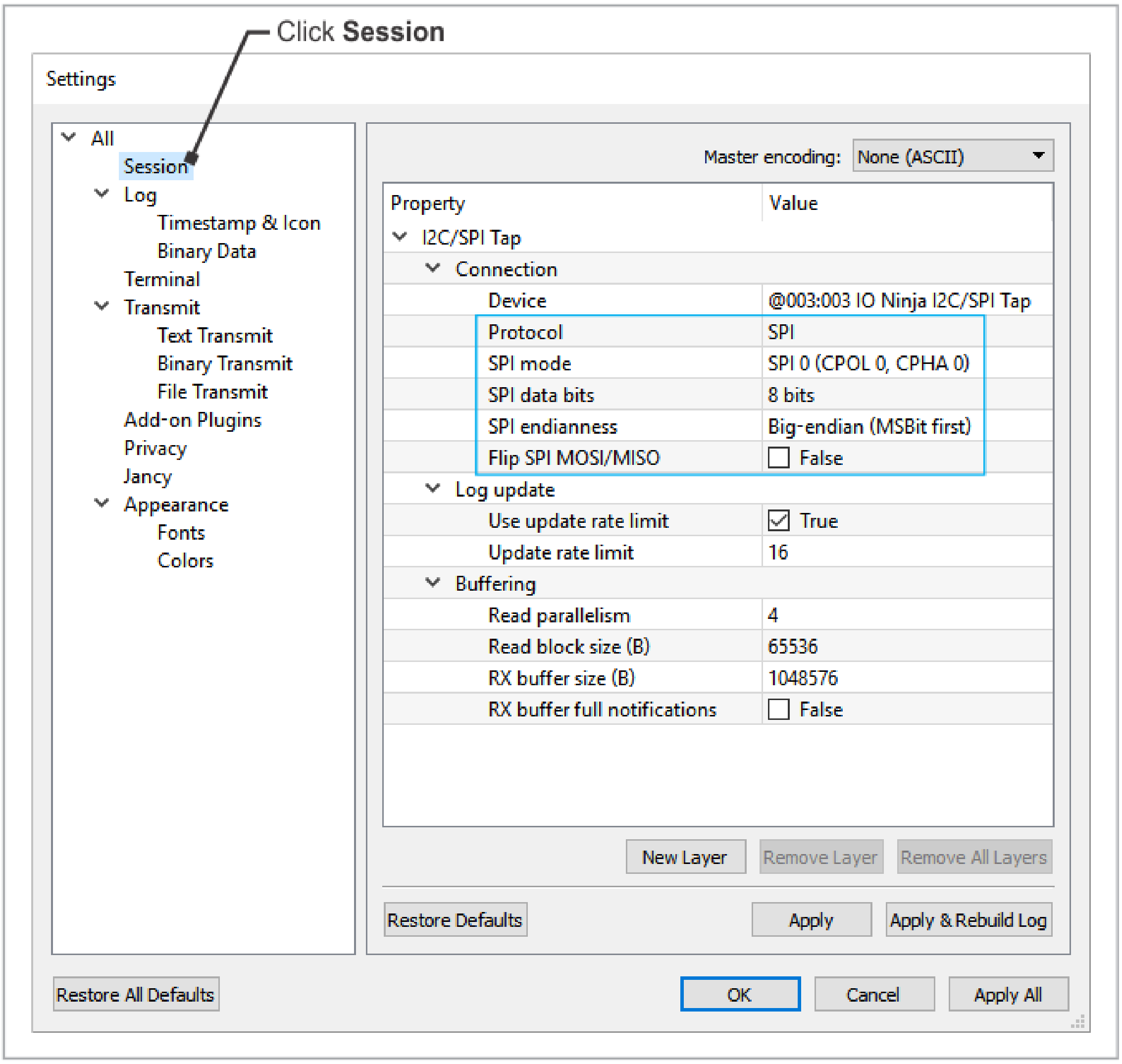
I2C Mode Screenshot



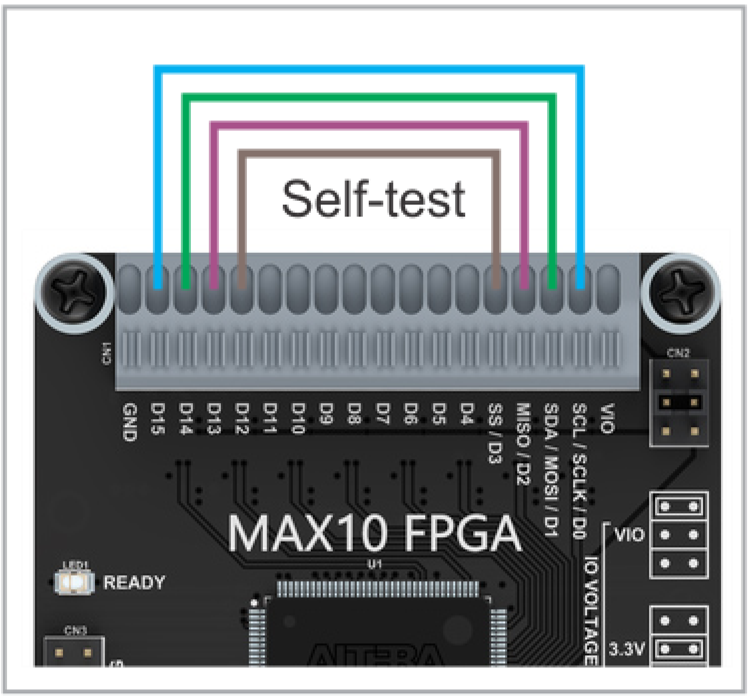
SPI Mode Screenshots

There are two screenshots in this section.




Click Session
Settings
All
Session
Log
Timestamp & Icon
Binary Data
Terminal
Transmit
Text Transmit
Binary Transmit
File Transmit
Add-on Plugins
Privacy
Jancy
Appearance
Fonts
Colors
Master encoding: None (ASCII)
Property Value
I2C/SPI Tap
Connection
Device @003:003 IO Ninja I2C/SPI Tap
Protocol SPI
SPI mode SPI 0 (CPOL 0, CPHA 0)
SPI data bits 8 bits
SPI endianness Big-endian (MSBit first)
Flip SPI MOSI/MISO False
Log update
Use update rate limit True
Update rate limit 16
Buffering
Read parallelism 4
Read block size (B) 65536
RX buffer size (B) 1048576
RX buffer full notifications False
New Layer
Remove Layer
Remove All Layers
Restore Defaults
Apply
Apply & Rebuild Log
Restore All Defaults
OK
Cancel
Apply All

Using Build-in Test Generator



I2C and SPI firmware (FPGA configurations) for this Tap include test generators producing sustained test traffic at 400KHz and 24MHz respectively.

Test I2C signals are generated on D14 (SDA) and D15 (SCL).

Test SPI signals are generated on D12 (SS), D13 (MISO), D14 (MOSI), and S15 (SCLK).

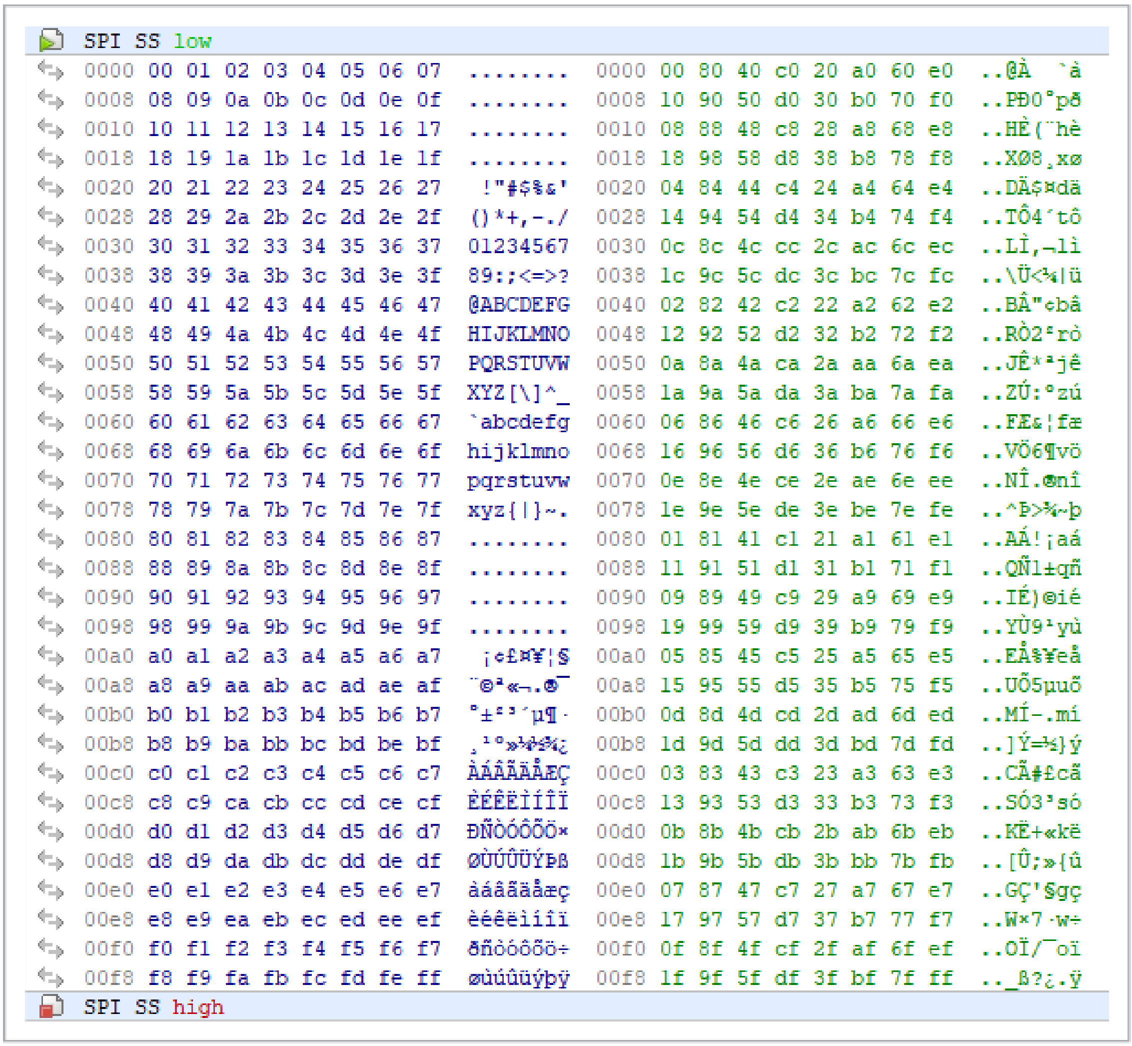
To receive the test traffic, connect two or four wires as shown on the diagram above.

Here is the data you will receive from the test generators.

I2C Test Sequence

```
I2C start (read, I2C 7-bit address: 0x08)
→ 0000 11 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e ................
→ 0010 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e ................
→ 0020 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e . !"#$%&'()*+,-.
→ 0030 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e /0123456789:;<=>
→ 0040 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e ?@ABCDEFGHIJKLMN
→ 0050 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e OPQRSTUVWXYZ[\]^
→ 0060 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e _`abcdefghijklmn
→ 0070 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e opqrstuvwxyz{|}~
→ 0080 7f 80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e ................
→ 0090 8f 90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e ................
→ 00a0 9f a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae . ¡¢£¤¥¦§¨©ª«¬.®
→ 00b0 af b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be ¯°±²³´µ¶·¸¹º»¼½¾
→ 00c0 bf c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce ¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎ
→ 00d0 cf d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de ÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞ
→ 00e0 df e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ßàáâãäåæçèéêëìíî
→ 00f0 ef f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ïðñòóôõö÷øùúûüýþ
→ 0100 ff 79 7a                                        ÿyz
I2C stop
I2C start (write, I2C 7-bit address: 0x08)
← 0000 10 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e ................
← 0010 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e ................
← 0020 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e . !"#$%&'()*+,-.
← 0030 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e /0123456789:;<=>
← 0040 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e ?@ABCDEFGHIJKLMN
← 0050 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e OPQRSTUVWXYZ[\]^
← 0060 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e _`abcdefghijklmn
← 0070 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e opqrstuvwxyz{|}~
← 0080 7f 80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e ................
← 0090 8f 90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e ................
← 00a0 9f a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae . ¡¢£¤¥¦§¨©ª«¬.®
← 00b0 af b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be ¯°±²³´µ¶·¸¹º»¼½¾
← 00c0 bf c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce ¿ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎ
← 00d0 cf d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de ÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞ
← 00e0 df e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ßàáâãäåæçèéêëìíî
← 00f0 ef f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ïðñòóôõö÷øùúûüýþ
← 0100 ff 79 7a                                        ÿyz
I2C stop
```

SPI Test Sequence



Specifications

I2C/SPI Tap is supplied with a USB cable.

Hardware specifications

USB port	USB 2.0, high-speed (480MHz), USB-C connector
Supported I2C clock speeds	Up to 700KHz
Supported SPI clock speeds	Up to 24MHz [Note 1]
Operating temperature	0 to +60 degrees C
Operating relative humidity	10-90%
Mechanical dimensions	82 x 74 x 30 mm

Note 1

This number depends on the host PC's performance and its "workload." Captured data may be lost if PC's USB channel is unable to receive it in a timely manner. Our tests have shown that an average i7-based desktop PC (that is not performing other time-critical tasks) easily captures sustained SPI traffic with clock speeds up to 24MHz.

All specifications are subject to change without notice and are for reference only. Tibbo assumes no responsibility for any errors in this Manual and does not make any commitment to update the information contained herein.

Is the data you are looking for missing from the above table? Do not just assume that you know the answer — talk to Tibbo! Remember, that the ultimate responsibility for all decisions you make regarding the use and the mode of use of Tibbo products lies with you, our Customer.

Ethernet Tap

Ethernet Tap is an affordable sniffer for monitoring Ethernet traffic flowing through a single Ethernet connection. To monitor Ethernet traffic, an Ethernet cable that was initially connecting a switch (hub) to an Ethernet device is replaced by two Ethernet cables and the Ethernet Tap is inserted between them, in what is known as a "wedge" fashion. The Tap passes the Ethernet traffic through with a delay equal to about 5.2uS.

The Tap connects to your PC via a USB cable and is compatible with IO Ninja terminal/sniffer software.

Features

- Designed for use with IO Ninja software (http://ioninja.com/).
- Reliably captures 10Base-T and 100Base-T Ethernet traffic[Note 1][Note 2].
- Based on Intel MAX10 FPGA.
- Six LEDs onboard:
  - Green Ready LED;
  - Green 100Mbps and yellow Link/Activity LEDs for the left RJ45 jack;
  - Green 100Mbps and yellow Link/Activity LEDs for the right RJ45 jack;
  - Blue Power LED.
- High-speed (480Mbps) USB2.0 interface on a USB-C connector.
- Supplied with a USB-C cable.
- USB-powered, no additional external power necessary.
- Compact, outside dimensions only 82 x 74 x 30 mm.
- The product is field-upgradeable via the USB interface.

Note 1

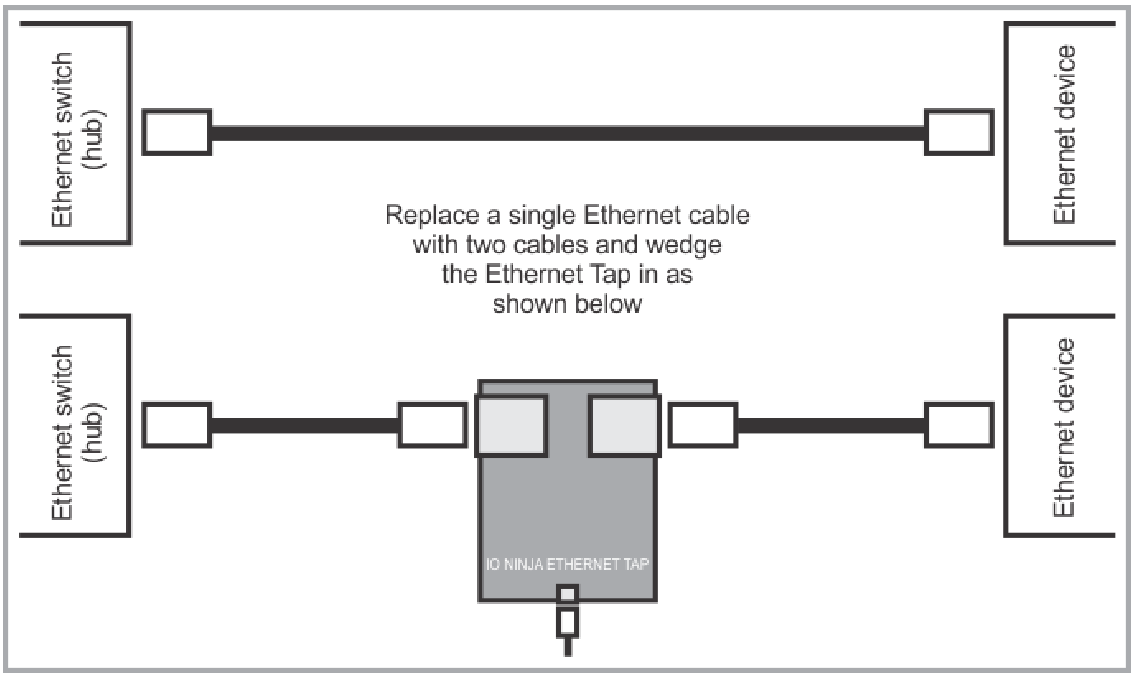
As long as your PC is fast enough to receive all the captured data. Some data may be lost if a PC's USB channel is unable to receive it on time. Our tests have shown that an average i7-based desktop PC (that is not performing other time-critical tasks) easily captures 100BaseT traffic even under high Ethernet load conditions.

Note 2

The Tap does not allow mixing 10BaseT and 100BaseT connections. Both sides of the Tap must have the same connection speed. That means that you may not connect one side of the Tap to a 100Base-T device, and another side—to a 10Base-T device.

Using Ethernet Tap

Connecting Ethernet Tap



Setting up IO Ninja

- Install and run IO Ninja (http://ioninja.com/).
- Plug the Ethernet Tap into the USB port of your PC—the blue **Power** LED shall turn on. If the Tap is loaded with a valid configuration file, the green **Ready** LED will also turn on.
- Launch the **Ethernet Tap** plugin (we have a screenshot here).
- Click on the **Tap** drop-down and select the target Ethernet Tap (you can connect several Ethernet Taps to one PC, so selecting the right one may be necessary). You can refresh the list of available Ethernet Taps by clicking the **Refresh** button to the right of the Tap drop-down.
- Click the **Capture** button to start the data acquisition. At this moment, IO Ninja will check the Tap's internal firmware. If the Tap firmware is outdated, IO Ninja will upload the new firmware into the Tap. This will only take a couple of seconds.

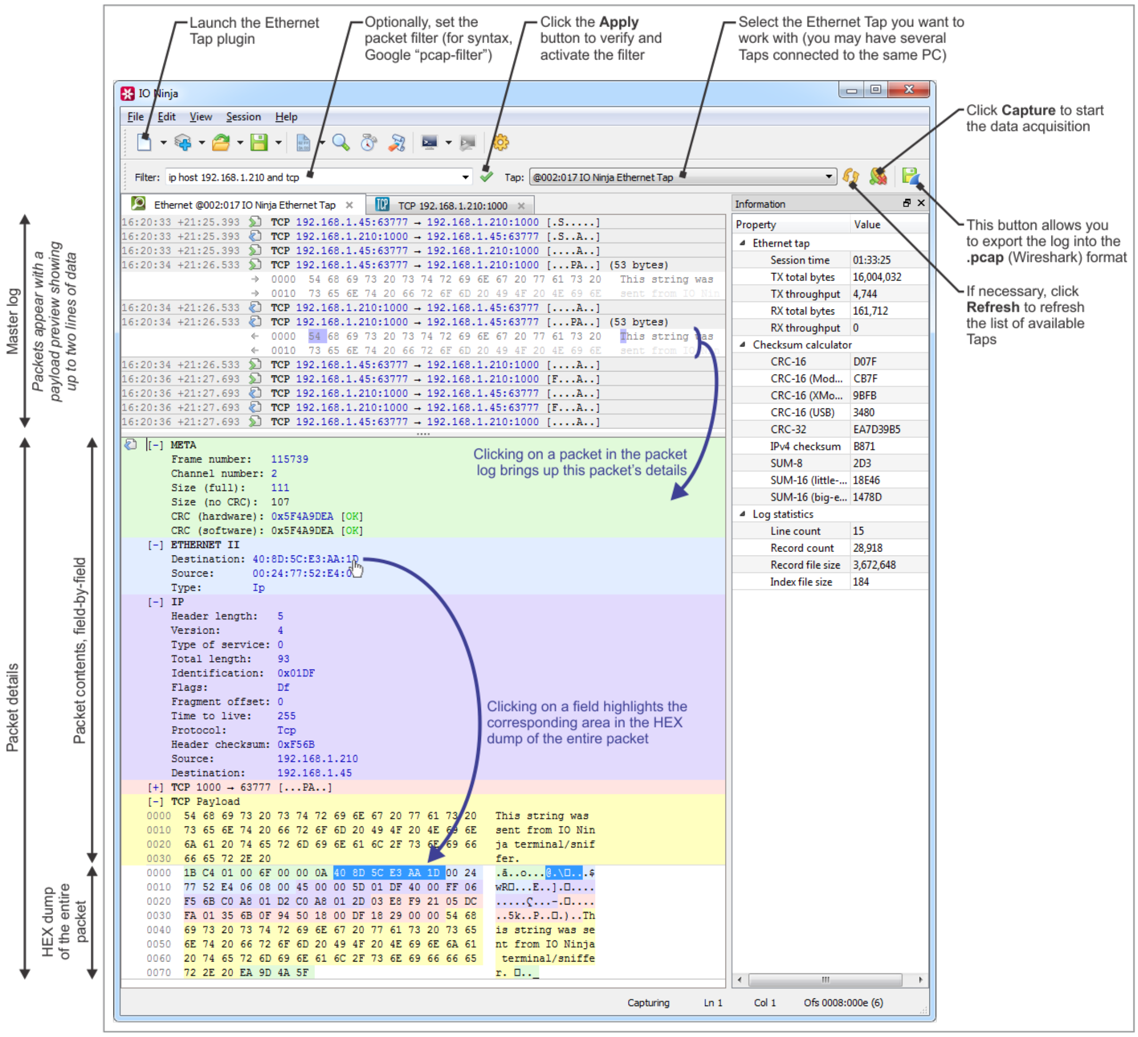
Filtering captured data

Most networks experience constant and varied traffic, while your attention is usually focused on something particular. Filtering allows you to only display the packets that satisfy the defined criteria. The filter is set by entering a filter string into a **Filter** textbox and pressing the **Apply** button to the right of the field. When setting a filter, follow the "pcap filter" format, and if you don't know what that is, Google "pcap-filter."

Clicking the **Apply** button verifies the filter string correctness and enables that filter if no problems are found. If the filter string is invalid, the **Filter** textbox will turn red.

Note that the Ethernet Tap captures all data flowing through it, and the filtering is performed on the already recorded data. Changing the filter affects what packets are displayed on the screen and is retroactive, meaning that setting a new filter causes IO Ninja to re-output the entire log with the new filter setting applied.

Ethernet Tap Plugin Screenshot



Specifications

Ethernet Tap is supplied with a USB cable.

Hardware specifications

USB port	USB 2.0, high-speed (480MHz), USB-C connector
Throughput	Up to 100BaseT Ethernet's real-life throughput [Note 1] [Note 2]
Operating temperature	0 to +60 degrees C
Operating relative humidity	10-90%
Mechanical dimensions	82 x 74 x 30 mm

Note 1

As long as your PC is fast enough to receive all the captured data. Some data may be lost if a PC's USB channel is unable to receive it on time. Our tests have shown that an average i7-based desktop PC (that is not performing other time-critical tasks) easily captures 100BaseT traffic even under high Ethernet load conditions.

Note 2

The Tap does not allow mixing 10BaseT and 100BaseT cables. Both sides of the Tap must have the same connection speed. That means that you may not connect one side of the Tap to a 100Base-T device, and another side—to a 10Base-T device.

All specifications are subject to change without notice and are for reference only. Tibbo assumes no responsibility for any errors in this Manual and does not make any commitment to update the information contained herein.

Is the data you are looking for missing from the above table? Do not just assume that you know the answer — talk to Tibbo! Remember, that the ultimate responsibility for all decisions you make regarding the use and the mode of use of Tibbo products lies with you, our Customer.

Update history

09DEC2019

- Documented the Ethernet Tap.

09SEP2019

- Corrected an error in the Serial Tap documentation (UART (TTL) mode allows to monitor UARTs of IC chips with logic levels from 3.3V, not 1.8V as was previously stated).

26MAR2019

- Documented the I2C/SPI Tap.

18SEP2018

- Updated Serial Tap documentation to reflect the changes made in revision D.

03JUL2018

- Documented the Serial Tap.