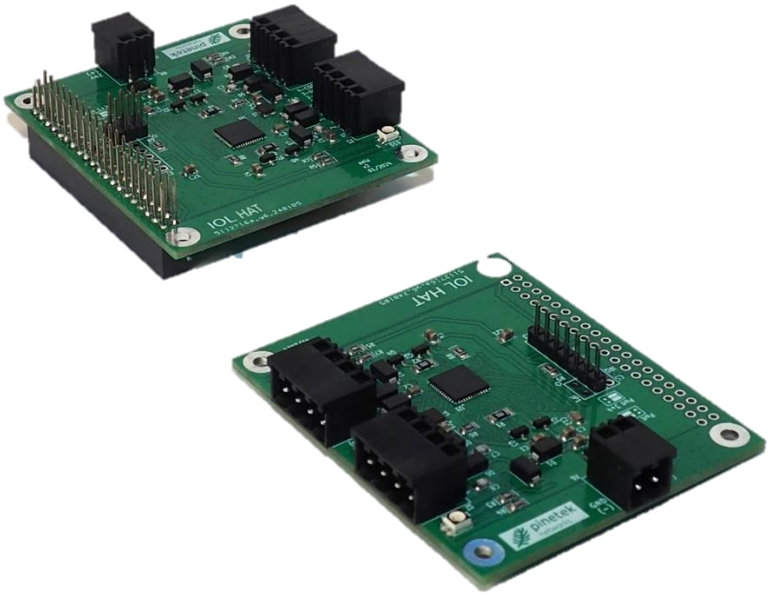
pinetek networks

DRAFT VERSION

Data sheet IOL HAT

2024.01

REV 1 - DRAFT



www.pinetek-networks.com

Introduction

Thanks for purchasing an IOL HAT module. Please read this information carefully to ensure safe and hassle-free operation of the device.

The following signs will guide you for dedicated areas of interest:

	This symbol is used to show areas of special attention that need to be considered for the safe operation of the IOL HAT.
	This symbol is used to show information relevant for the show areas of special attention that need to be considered for the safe operation of the IOL HAT.

Product Description

The IOL HT enables the exchange of data with smart SDCI sensors and actuators (IEC 61131-9, brand name IO-Link®). The IOL HAT works as a plug-in module for Raspberry Pi®[1] using a 40-pin GPIO connector or via a generic interface with other single-board computers. Two SDCI interfaces are available per IOL HAT, up to two IOL HATs can be connected to the Raspberry Pi via stack-through connectors.

The internal interface is implemented as a TCP socket interface so that integration is independent of the programming language of the user application. The software for the master functionality and the TCP interface (master application) are available as binary and source code.

With the IOL HAT, it is possible to integrate over 20,000 different types of IO-Link devices into Raspberry Pi applications.

The IOL HAT is available in two version:

- PT-1201 with Raspberry Pi specific 40-pin connector
- PT-1202 with a generic host connector

[1] Raspberry Pi is a trademark of Raspberry Pi Ltd: raspberrypi.com ↩

Scope of delivery

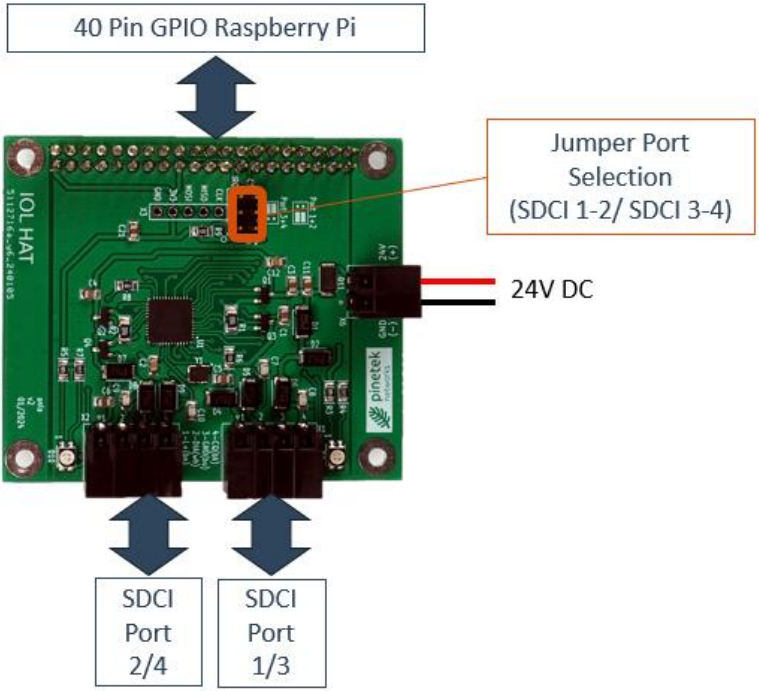
IOL HAT for Raspberry Pi (Item PT-1201)

- 1x IOL HAT with 40-pin GPIO connector incl. 2x jumpers for port selection
- 2x 4-pin connector for 2xSDCI port
- 4x standoffs incl. nuts
- 1x 2-pin connector for 24 Volt connection
- Quick start guide

IOL HAT for generic connection (Item PT-1202)

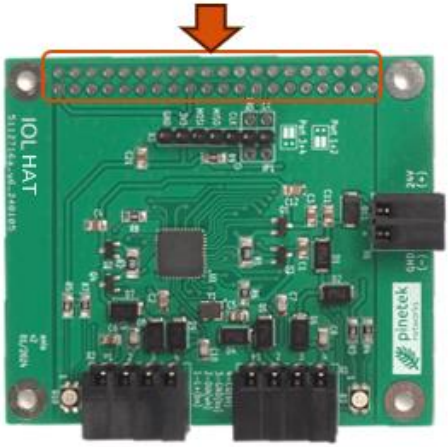
- 1x IOL HAT with generic connector
- 2x 4-pin connector for 2xSDCI port
- 4x standoffs incl. nuts
- 1x 2-pin connector for 24 Volt connection
- Quick start guide

Connections



ⓘ	All logic signals (SPI, interrupts) on the 40-pin GPIO and the generic interface X3 work on 3V3 logic levels.

40-pin GPIO to Raspberry PI and jumper (PT-1201 only)



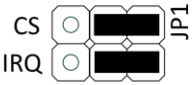
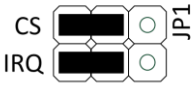
This connector is implemented as stack-through, i.e. up to two IOL HATs can be connected on top of each other. If the pins are not occupied by the connected IOL HAT, additional expansion boards can be connected via the stack-through connector.

⚠	Ensure the correct orientation of the GPIO connector. Misalignment or rotation can cause damage on the Raspberry Pi host and the IOL HAT.

The following pins are always in use when one or more IOL Hat modules are installed:

Usage	**Pin #**
SPI: MISO (Master-In-Slave-Out)	21 (SPI_MISO)
SPI: MOSI (Master-Out-Slave-In)	19 (SPI_MOSI)
SPI: CLK (Clock)	29 (SPI_CLK)
3V3	1,17
GND	6,9,14,20,25,30,34,39

The SDCI ports 1+2 or 3+4 are selected via jumper JP1. The following positions select the corresponding SDCI port:

SDCI Port 1+2	SDCI Port 3+4
	

Jumper positions other than those shown are not valid.

	If the devices are stacked, ensure that one device uses SDCI Port 1+2 and the other uses SDCI Port 3+4. Otherwise communication with the ports will not be possible.

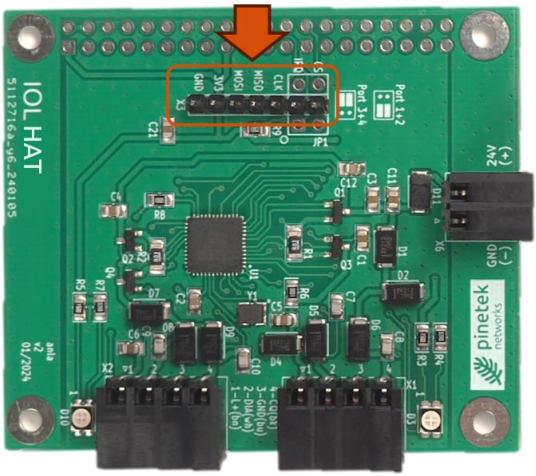
The following pins are used depending on the port selection:

Usage	SDCI Port 1+2 Pin #	SDCI Port 3+4 Pin #
SPI: CE (Chip Enable)	24 (SPI_CE0)	Pin 26 (SPI_CE1)
Interrupt	18 (GPIO24)	Pin 22 (GPIO25)

For the SPI interface, the Raspberry Pi acts as “master”, the IOL HAT as “slave”. The SPI mode is Mode 0 (CPOL=0; CPHA=0).

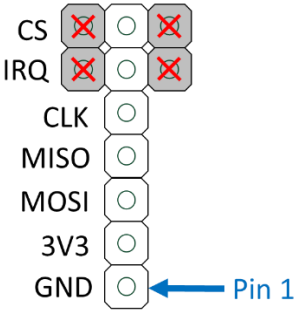
All pins in the GPIO connector are connected through without modification.

Generic Interface (PT-1202 only)



The interface to the host system is on X3 with the following pin assignments:

Pin #	Usage
1	GND
2	3V3
3	SPI: MOSI (Master-Out-Slave-In)
4	SPI: MISO (Master-In-Slave-Out)
5	SPI: CLK (Clock)
6	IRQ (Interrupt to host)
7	SPI: CS (Chip Select/Chip Enable)

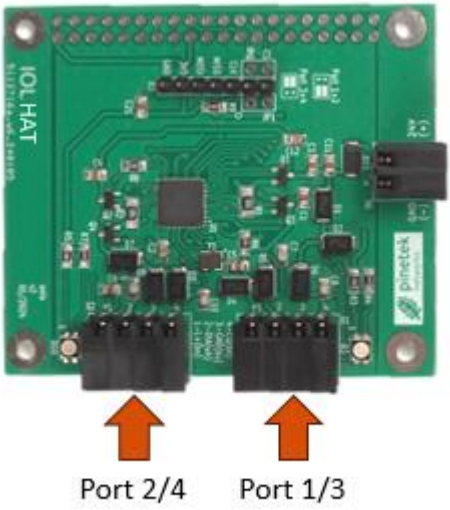


The crossed-out pins are not used in this configuration.

For the SPI interface, the Raspberry Pi acts as “master”, the IOL HAT as “slave”. The SPI mode is Mode 0 (CPOL=0; CPHA=0).

	Ensure the correct pin assignment of the GPIO connector. Misalignment or rotation can cause damage on the host module and the IOL HAT. Note that the 3V3 and GND pins are not reverse-polarity protected.

SDCI-Ports



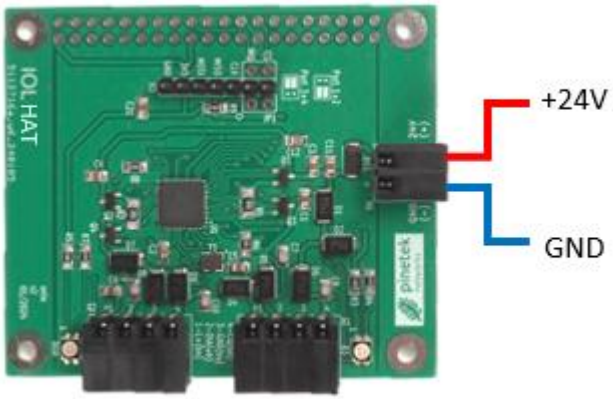
Both SDCI ports use the same pin-out. For industrial use, the signal lines on the SDCI ports have improved surge immunity. Each port features a bi-color LED for signalization of the operation mode. The typical M12 cable coloring scheme for pre-configured cables is listed.

Pin #	Usage	M12 cable color
1	L+ (24V)	brown
2	DIA (Input/Output line)	white
3	GND	blue
4	CQ (Serial line)	black

The connector is 4pin, 3,5mm MC-type.

⚠	Each SDCI Port is designed to supply max. 500mA on the L+ line and 150mA on the CQ line. The current supply for the L+ is limited by software. If a custom master application is used, the limits must be set in the driver configuration.

24V-Input



The 24V power supply input is used for the L+ supply of the SDCI ports. The logic part of the module is supplied over the 3V3 on the 40-pin GPIO connector or the generic connector.

⚠	Depending on the connected SDCI devices, the 24V input can consume up to 1,3A current. Please use adequate cabling. The port is not overcurrent protected, the L+ lines are current limited by the driver.

The 24V power input is reverse-current protected. The GND potential is a common potential with the host (Raspberry).

The connector is 2pin, 3,5mm MC-type.

API

Master application

To operate the IOL HAT on a Raspberry Pi, the master application binary from the IOL HAT repository is used:

https://github.com/Pinetek-Networks/iol-hat/bin/

The suitable master application file for the used Linux distribution (32bit/64bit) for the required port (SDCI 1+2 / SDCI 3+4) shall be used.

	To operate two IOL HATs on one Raspberry Pi, two instances of the master application need to be executed. Only run one instance of the same master application at once, otherwise the behavior will be undefined.

The Linux version can be determined with

```
uname -a
```

Upload the master application to the Raspberry Pi and run it. The master application will open a TCP server on the following ports:

- SDCI 1+2: Port 14001
- SDCI 3+4: Port 14002

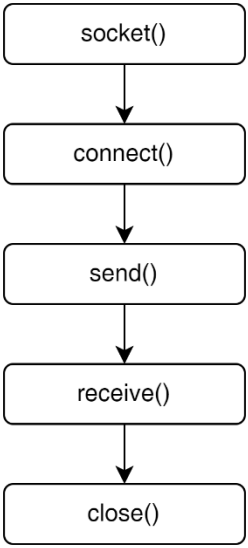
The client can connect to those ports and communicate through the binary prototcol as described below.

	The TCP connection is established for each communication with the master application.

Binary protocol

Sequence

The sequence for communication over TCP is as follows:



The commands and responses are exchanged in the send() and receive() part of the sequence.

The first octet in the command structure always defines the command ID.

	Some commands use a port ID. This refers to the first or second port on the IOL HAT. Which IOL HAT is addressed is determined through the TCP port. Port ID 1 can refer to IOL HAT port 1 or 3, Port ID 2 to IOL HAT port 2 or 4, depending if TCP port 14001 or 14002 is used.

The commands have a common structure in the first two octets:

Octet #	Usage
0	Command ID
1	Length

In case of a command success, the return message as described in the commands is returned by the TCP server.

Error message

In case of an error, the error message is returned:

Octet #	Usage
0	Error message ID = 0xFF
1	Error code

The following error codes are defined:

Error code	Usage
0x01	Message Length Error
0x02	Function ID unknown
0x03	Port power error, e.g. read while port disabled
0x04	Port ID error, i.e. port ID >2 called
0x05	Internal error, in this case octets 2..3 define the specific error code from the I-Link stack
0x06	Wrong status, e.g. data exchange when power off or no connection to the device

The binary protocol contains the following commands:

CMD_PWR

The power command switches the L+ power on the corresponding port. After switching on, the communication speed on the SDCI (COM1, COM2, COM3) is automatically detected and cyclic exchange of data is started.

Command:

Octet #	Usage
0	Command ID = 0x01
1	Port ID 0x01 or 0x02
3	Power status 0x00 = OFF 0x01..0xFF=ON

Return Success:

Octet #	Usage
0	Command ID = 0x01
1	Port ID 0x01 or 0x02
3	Power Status

CMD_LED

The LED command switches the LED on the corresponding port.

Command:

Octet #	Usage
0	Command ID = 0x02
1	Port ID 0x01 or 0x02
3	LED Status (see below)

The LED status is 0x01 for the green LED and 0x02 for the red LED. With LED status 0x03, both LEDs are activated.

Return Success:

Octet #	Usage
0	Command ID = 0x02
1	Port ID 0x01 or 0x02
3	LED Status

CMD_PD

This command exchanges the process data with the connected device. Process data is not validated in length or content.

Command:

Octet #	Usage
0	Command ID = 0x03
1	Port ID 0x01 or 0x02
2	Length Out
3	Length In
4..	Data Out

The Length Out field command is length for the output process data. The structure and length of the output data can be found in the SDCI device's documentation. The Length In field needs to be set according to the device's documentation. If the length fields are not correctly set, this may lead to invalid data or loss of communication.

Return Success:

Octet #	Usage
0	Command ID = 0x03
1	Port ID 0x01 or 0x02
2	Length Out (as given in the command)
3	Length In (as given in the command)
4..	Data In

CMD_READ

This command reads a parameter with the index and subindex as given on the addressed port.

Command:

Octet #	Usage
0	Command ID = 0x04
1	Port ID 0x01 or 0x02
2..3	Index, 16-bit value
4	Subindex
5	Length

The length field is the desired/maximum length for the attribute. The structure of the attribute can be found in the SDCI device's documentation.

Return Success:

Octet #	Usage
0	Command ID = 0x0
1	Port ID 0x01 or 0x02
2..3	Index, 16-bit value
4	Subindex
5	Read Length
6..	Read Data

The structure for the Read Data can be found in the SDCI device's documentation.

CMD_WRITE

This command writes a parameter with the index and subindex as given on the addressed port.

Command:

Octet #	Usage
0	Command ID = 0x05
1	Port ID 0x01 or 0x02
2..3	Index, 16-bit value
4	Subindex
5	Length
6..	Write Data

The structure of the write data can be found in the SDCI device's documentation.

Return Success:

Octet #	Usage
0	Command ID = 0x04
1	Port ID 0x01 or 0x02
2..3	Index, 16-bit value
4	Subindex
5	Length (as given in the command)

To verify the write operation, a read operation (Command ID 0x04) can be used on the same Index and Subindex.

CMD_STATUS

The status command returns the status of the addressed port.

Command:

Octet #	Usage
0	Command ID = 0x06
1	Port ID 0x01 or 0x02

Return Success:

Octet #	Usage	Comment
0	Command ID = 0x06	
1	Port ID 0x01 or 0x02	
2	Process data IN valid	0x00 = Not valid 0x01 = Valid
3	Process data OUT valid	
4	Transmission rate	0x00 = Detection Failure 0x01 = COM1 (4.8 kbps) 0x02 = COM2 (38.4 kbps) 0x03 = COM3 (230.4 kpbs)
5	Process data IN length	
6	Cycle time	
7	Process data OUT length	
8..9	Vendor ID	16-bit value
10..13	Device ID	32-bit value
14	Power	0x00 = Power OFF 0x01 = Power ON

Technical data

Mechanical

Dimensions

- 65,5 × 66 mm
- Height: 18 mm with stack-through

Mounting

- 4×2,5 mm Screwholes (Compatible with Raspberry Pi mounting holes)

Temperature range

- -25 °- 60 °C

Connections

Voltage supply

- 24V DC (+/-20 %), Current consumption max. 1300mA
- 3V3 DC (+/-5%) from Raspberry Pi < 50mA, over 40-Pin GPIO or generic interface (for IO-Link-Driver-IC logic and LEDs)

2 SDCI-Ports

- Screw plug-in terminal 3,5mm
- Class A (max. 500mA per port)
- Speeds COM1, COM2, COM3 (Auto-Detect)
- Operating mode: IOL
- 1 LED per port (red/green)

40 Pin GPIO Connector (for connection to Raspberry Pi, as stack-through)

- SPI 1 (MISO, MOSI, CLK)
- Port 1/2: SPI_CE0, GPIO24 (interrupt)
- Port 3/4: SPI_CE1, GPIO25 (interrupt)
- 3V3/GND

Generic Connector (for single-board computers other than Raspberry Pi)

- SPI (MISO, MOSI, CLK, CE)
- Interrupt (GPIO)
- 3V3, GND

LEDs

- Status per port (red/green)

Support

Support can be obtained from the website
www.pinetek-networks.com/en/support

Manufacturer information

Pintek Networks UG (haftungsbeschränkt)

Bachstelzenweg 8
DE-88677 Markdorf

www.pinetek-networks.com