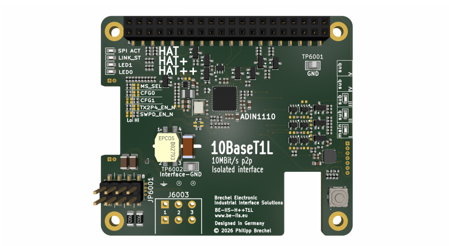
Isolated 10BASE-T1L Industrial Ethernet HAT++ for Raspberry Pi



Brechel Electronic
Industrial Interface Solutions
Designed by Philipp Brechel (Germany)
www.be-iis.eu | www.github.com/be-iis

Document ID: BE-IIS-HPP-T1L_RevB_Datasheet
Revision: Rev. B
Date: May 20, 2026
Document Status: Released
Product Status: Final Testing

10BASE-T1L (IEEE 802.3cg) enables 10 Mbit/s Ethernet communication over a single twisted pair for long-reach point-to-point links.

This HAT enables 10BASE-T1L communication based on the Analog Devices ADIN1110 Ethernet MAC-PHY and, as part of the BE-IIS HAT++ ecosystem, is stackable with other HATs from the portfolio for seamless system expansion and simplified system integration.

Key Features

- 10BASE-T1L compliant with IEEE 802.3cg
- ADIN1110 10BASE-T1L MAC-PHY
- Galvanically separated MDI
- Raspberry Pi HAT+ compliant (2024)
- Stackable HAT (BE-IIS-HAT++)
- Configurable CS and IRQ routing
- Optional PoSPE support
- Connector J6000 updated to new connector option
- RoHS compliant
- Quality component suppliers

Product Description

The 10BASE-T1L Industrial Ethernet HAT is a Raspberry Pi HAT+ compliant Single Pair Ethernet interface board in accordance with IEEE 802.3cg. It integrates an ADIN1110 10BASE-T1L MAC-PHY and provides galvanic separation between the logic side and the field-side interface.

Applications

- Technology evaluation
- Industrial network evaluation
- Prototyping
- Education and laboratory use
- Long-reach Ethernet node evaluation

Contents

1 Introduction

The BE-IIS HAT++ 10BASE-T1L Industrial HAT is a Raspberry Pi HAT+ compliant interface board for 10BASE-T1L communication in industrial and laboratory environments.

The board integrates an ADIN1110 10BASE-T1L MAC-PHY [1], providing direct Ethernet connectivity over a single twisted pair. The MDI is galvanically isolated from the logic domain, enabling robust operation in electrically harsh environments and across long cable distances.

10BASE-T1L enables reliable long-distance Ethernet communication with data rates of 10 Mbit/s over single-pair cabling, making it well suited for industrial field-level connectivity and sensor/actuator networks.

The design is fully aligned with the **HAT++ ecosystem**, enabling **stackable operation** of multiple interface boards on a single Raspberry Pi. The HAT++ concept ensures **conflict-free resource allocation** and allows flexible combinations of different communication interfaces within one system.

The HAT++ ecosystem includes a growing portfolio of industrial interface modules such as **10BASE-T1S**, **Ethernet (SPI-based)**, **CAN/CAN-FD**, **Modbus/RS-485**, and additional communication and measurement interfaces.

The HAT is currently compatible with a wide range of **64-bit Raspberry Pi platforms**, including **Raspberry Pi 3, 4, and 5**, as well as **Raspberry Pi Zero 2 W**. Support for **32-bit Raspberry Pi operating systems** is currently under development.

All required software components, including drivers and system configuration, are provided through the **BE-IIS installer**, enabling a simple and reproducible setup process.

In line with the BE-IIS design philosophy of transparency, **schematics**, **PCB layout data**, and **3D models** are available, allowing full insight into the design and enabling users to build upon it. (see Section 2).

Optionally, an add-on HAT is available to provide **Power over Single Pair Ethernet (PoSPE)** (see 11).

The HAT can be used as a standard single HAT, as a fully compliant HAT+ device with automatic configuration, or as part of the BE-IIS HAT++ ecosystem for scalable industrial setups.

The HAT supports 10BASE-T1L operation and can be used for evaluation, prototyping, industrial network testing, and educational purposes.

2 Design Resources

All design files and software resources are publicly available.

HTML **Product Page**
https://www.be-iis.eu/products/BE-IIS-HPP-T1L_B/

PDF **Datasheet (PDF)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1L_B/datasheet.pdf

PDF **Schematic (PDF)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1L_B/schematic.pdf

HTML **Layout & BOM (Interactive)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1L_B/ibom.html

STL **3D Model (STEP/STL)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1L_B/model.zip

GIT **GitHub Repository**
https://github.com/be-iis/be-iis-installer

GIT **Installer Script**
https://github.com/be-iis/be-iis-installer/blob/main/scripts/install/install-all.sh

3 Hardware Configuration

3.1 Main Features

The BE-IIS-HAT++T1L enables 10BASE-T1L communication on Raspberry Pi platforms. It allows a standard Raspberry Pi platform (e.g. Raspberry Pi Zero or Raspberry Pi 3/4/5, excluding Compute Module variants) to operate as a 10BASE-T1L node. Communication between the Raspberry Pi and the onboard MAC-PHY is implemented via the SPI interface with an interrupt line for event handling.

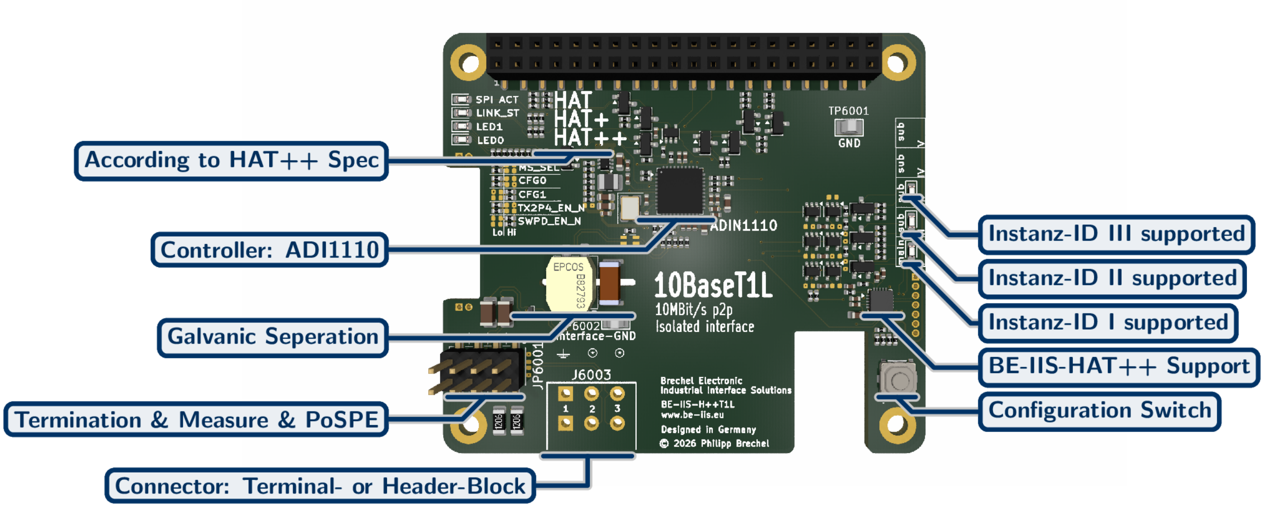


Figure 1: BE-IIS 10BASE-T1L top view with annotations

MAC-PHY

- Supplier: Analog Devices
- ADIN1110 10BASE-T1L MAC-PHY
- Crystal 25MHz

SPI Interface

- SPI0.0, SPI0.1, and SPI0.2 supported
- Persistent selection via push button

Isolation

- Galvanically separated MDI
- Implemented by transformer coupling
- Transformer dielectric test rating: 2250 V DC / 60 s

Power over Dataline

- Combination with a PoSPE HAT possible
- Supports currents up to 1 A
- Supports input voltages from 9 V to 33 V
- Includes an integrated 5 V main supply

Protocol Support

- 10BASE-T1L according to IEEE 802.3cg

Transmit Amplitude

- 1.0 $V_{p\text{-}p}$
- 2.4 $V_{p\text{-}p}$
- Controlled by autonegotiation

Master / Slave

- Prefer Slave (default)
- Role determined by autonegotiation

SPI Protocol

- Generic SPI
- 8-bit CRC

3.2 Block Diagram

The block diagram shown in Figure 2 is simplified. It illustrates the main functional blocks, power domains, separation barriers, and the principal data paths of the system.

The interrupt signal routing is not shown. It is configured using the same scheme as the chip-select (CS) routing. Reset and auxiliary control signals are omitted for clarity.

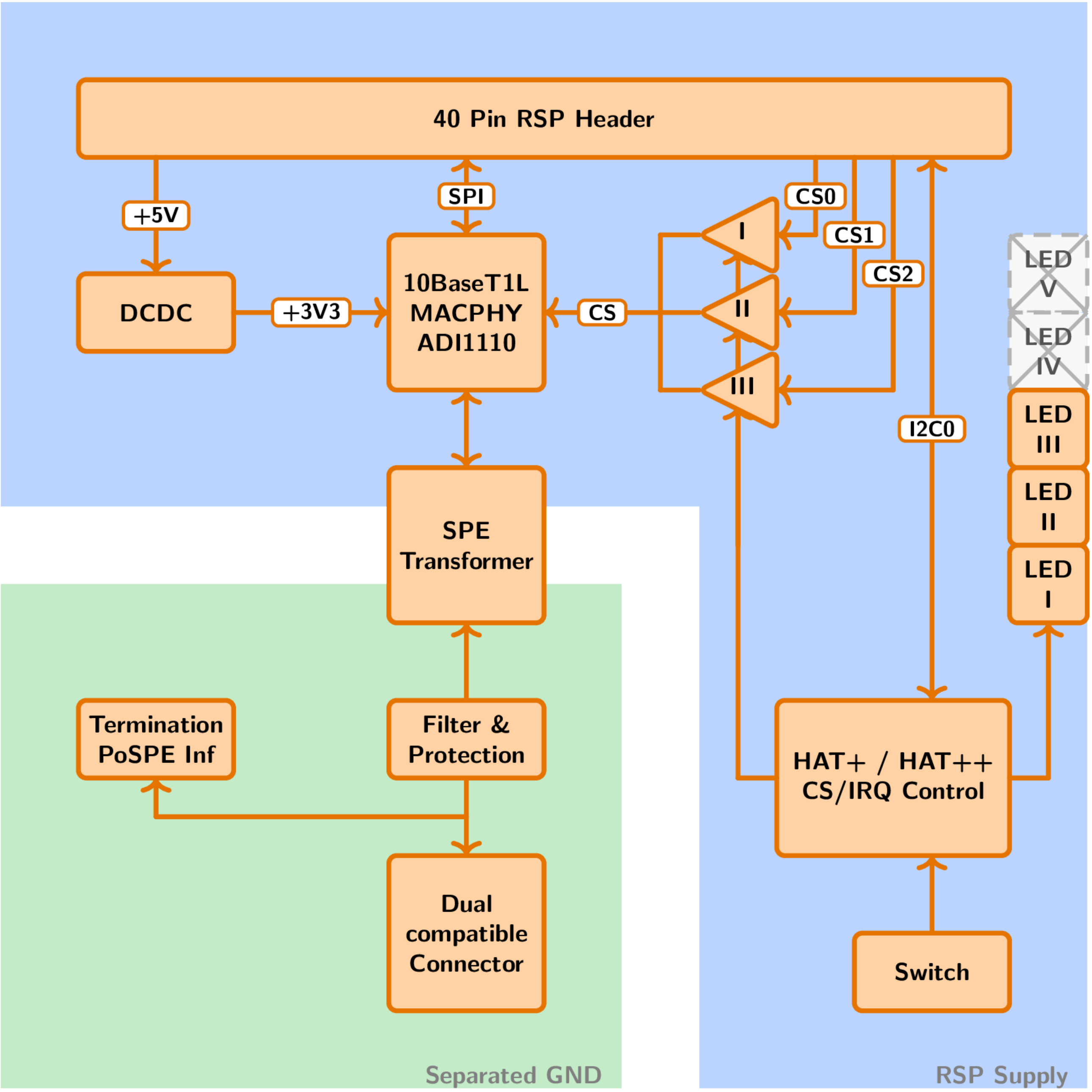


Figure 2: Simplified block diagram

3.3 Hardware Components

The selected hardware components and their interconnection are optimized to support reliable **10BASE-T1L Ethernet communication** at **10 Mbit/s** over **long cable distances** using single-pair cabling, including **galvanic isolation** of the MDI interface. The design is fully aligned with the **HAT++ ecosystem**, enabling modular integration and conflict-free operation in stacked configurations.

3.3.1 ADIN1110 10BaseT1L MACPHY

The **ADIN1110** by *Analog Devices* is a stand-alone **10BASE-T1L Ethernet MAC-PHY** with an SPI interface, designed to add long-distance Ethernet connectivity to systems without native Ethernet support.

It supports **10BASE-T1L** communication according to IEEE 802.3cg, enabling data rates of **10 Mbit/s** over single-pair cabling across distances of up to **1000 m**. Communication with the host is performed via a high-speed **SPI interface**, while the integrated PHY provides direct connection to the Ethernet medium via the MDI.

The device integrates essential networking functions including **frame buffering**, **filtering**, and **timestamping**, allowing efficient handling of Ethernet traffic while offloading processing tasks from the host CPU.

In Linux-based systems, the ADIN1110 is supported by dedicated drivers and can be integrated into the standard **network stack**, appearing as a conventional Ethernet interface (e.g. `eth0`). For further details, refer to the supplier's product documentation (see 1).

3.3.2 HAT++ Control Logic

The control logic manages HAT+, HAT++, and application-specific functionality. A **Texas Instruments microcontroller** is used to implement the control logic. On the **I^{2}C0 bus**, it behaves like an **AT28-compatible EEPROM**, thereby enabling the **HAT++ functionality** while maintaining compatibility with the standard HAT+ detection mechanism.

Tri-state buffers are used to switch the control paths and to selectively connect the required control electronics depending on the active operating mode.

Functions:

- HAT+ detection (EEPROM interface)
- HAT++ Instance ID detection
- LED control for status and mode indication
- Instance mode selection via push button
- Board resource management based on the selected Instance ID

Board Resource Management:

- Control of I^{2}C addresses for I^{2}C-based HATs
- Selection of SPI chip select (CS) for SPI-based HATs
- IRQ routing and handling
- Enable I^{2}C pull-ups in Instance ID I
- HAT-specific control functions

For details on the HAT++ system, see section 4.

3.4 Isolation

Galvanic isolation is implemented between the field-side interface and the Raspberry Pi domain.

The interface and the Raspberry Pi are galvanically isolated. The isolation barrier provides a minimum clearance distance of $\geq 5\,\mathrm{mm}$. In areas where this spacing cannot be maintained, isolation slots are implemented to ensure a creepage distance of at least **5 mm**.

All components bridging the isolation barrier are specifically designed and specified for isolation applications. A detailed list of these components is provided in 1.

The isolation implemented on the board provides **functional galvanic isolation**. The boards are delivered without labeling or certified testing and must therefore be considered as providing **functional insulation only**.

Higher isolation ratings can be achieved by using an alternative BOM, application-specific validation, testing, labeling, and certification. This can be provided upon request (see 7.3).

RefDes	Supplier	MPN	Description
L6002	TDK	ICI70CGI-222	Signal Transformer
C6013	PSK	FK21X102K502EGG	X1/Y2 Capacitor

Table 1: Isolation Components

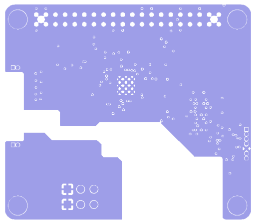
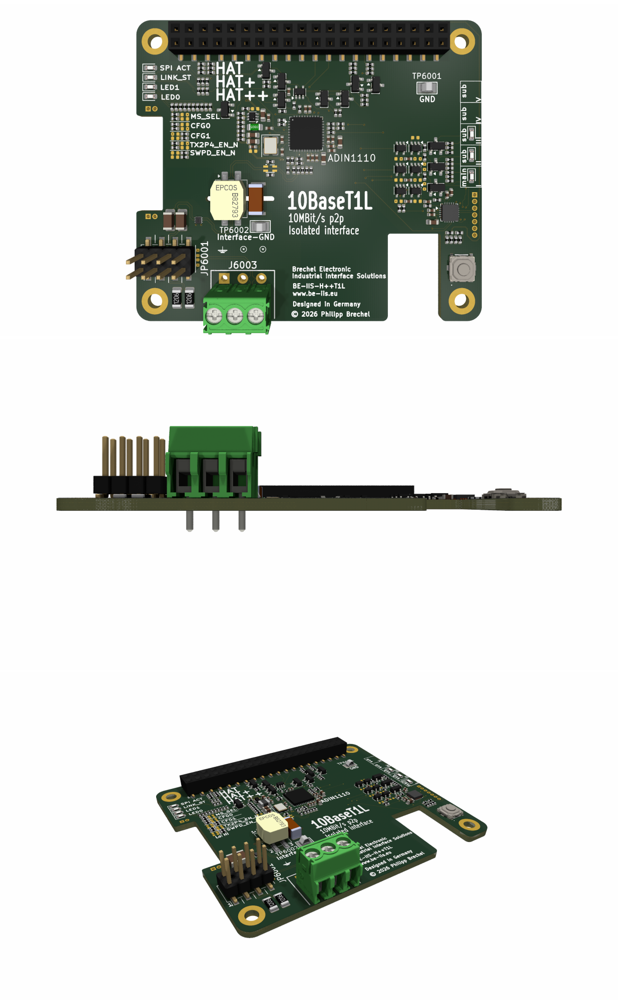


Figure 3: Isolation Barrier

3.5 Connectors

Connectors J4000 and J4001 provides the physical MDI Bus interface. Both Connectors are placed on the same BUS for ease Multipdrop usage. Two commonly used connector options can be populated. on both connector lands

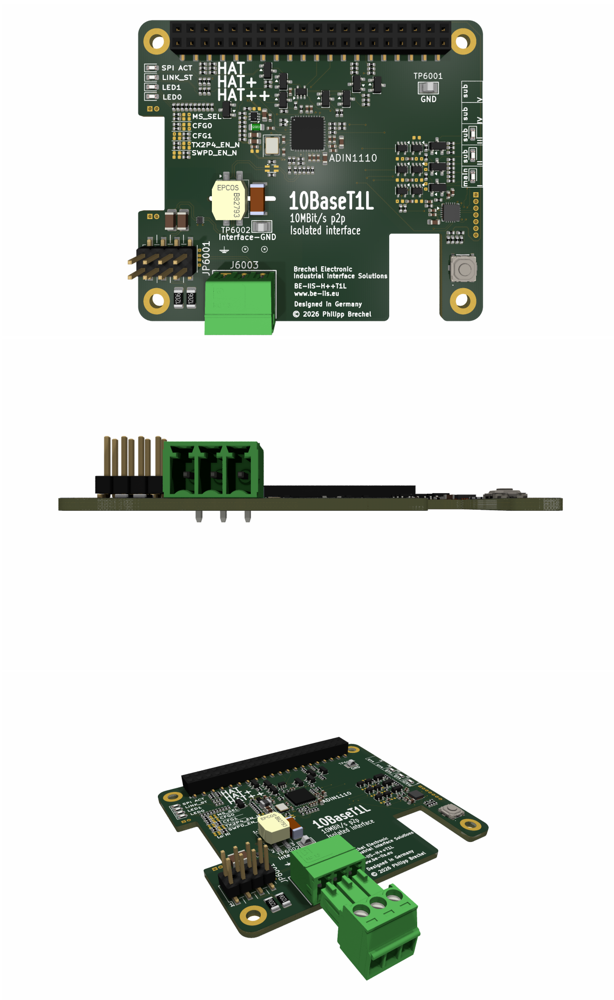


Description:
3-pin 3.5 mm rising cage screw terminal block
Optimized for One-HAT-Operation

Option1:
Phoenix Contact PT 1,5/3-3,5-H
Order Code: 1984620 [3]

Option2:
Wuerth Elektronik WR-TBL 300VAC 10A 3P
Order Code: 691214110003 [4]

Note: or equivalent component with compatible mechanical and electrical specifications
Pinning: 1: SHIELD (optional, cable shield) 2: SPE 3: SPE



Description:
3-pin 3.5m mm PCB header and connector
Optimized for Stacked-HAT-Operation

Option1:
PCB Header: Phoenix Contact - MC 1,5/3-G-3,5 [5]
PCB Connector: Phoenix Contact - MC 1,5/3-ST-3,5 [6]
Order Code: 1844223(Header), 1840379(Connector)

Option2:
PCB Header: Wurth Electronic - WR-TBL 300VAC 12A 3P [7]
PCB Connector: Wurth Elecztronic - WR-TBL 300VAC 10.5Am Vertical 26-16AWG [8]
Order Code: 691305140003(Header), 691361100003 (Connector)

Pinning: 1: SHIELD (optional, cable shield) 2: SPE 3: SPE
Note: This connector is included in the delivery

3.6 Jumper and Configuration

Jumper JP6001 provides configuration of the bus termination and, in combination with an optional PoSPE board, enables connection to the SPE signal line.
The jumper shall be installed by default. The intended use for removing the jumper is connection of a PoSPE board from the BE-IIS-HAT+ portfolio.
In 10BASE-T1L applications, termination is required in standard use cases.
Figure 4 shows the jumper configuration.

Pin	Signal	Pin	Signal	Note
1	Shield	2	Shield	
3	Termination	4	SPE1	Shorted by default
5	Termination	6	SPE2	Shorted by default
7	Shield	8	Shield	

Table 2: JP6001 pin assignment

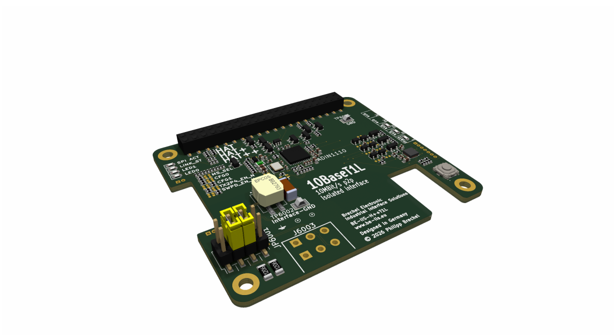


Figure 4: JP6001 jumper configuration

3.7 Indicators (LEDs)

In addition to the LED status bar on the side, which indicates the selected instance mode, several indicator LEDs are placed on the HAT. Their functionality is described in Table 3.

Reference	Name	Function	Default Behavior
D6004	SPI ACT	SPI bus activity indicator	Active during SPI communication
D6003	LINK_ST	Link status indicator	On when valid link is established
D6001	LED1	User-configurable LED	Disabled
D6002	LED0	User-configurable LED	On when valid link is established

Table 3: LED functionality

3.8 Signal Polarity and Wiring Orientation

The two signal wires of the 10BASE-T1L interface are not polarity-sensitive. Either wire may be connected to either signal terminal.

Both wiring orientations are valid and supported by the interface.

This applies to normal data communication as well as to operation with power over the data line [TODO].

4 HAT++

HAT++ is designed to enable conflict-free stacking of multiple HATs while keeping hardware and software integration simple.

The concept is based on transparency rather than a black-box design. All hardware resources, configurations, and software components are openly accessible following the BE-IIS transparency principles.

HAT++ is not limited to multi-board systems. The same concept can also be used with a single HAT, providing a consistent and scalable approach from simple setups to complex systems.

4.1 HAT++ Compatibility Concept

The board is designed according to the BE-IIS HAT++ design principles.

HAT++ is fully backward compatible with:

- Raspberry Pi HAT+
- Standard Raspberry Pi HAT

The HAT can be operated in three different modes:

- **HAT (manual)** – full manual configuration
- **HAT+ (autodetect)** – automatic detection via EEPROM
- **HAT++ (autodetect + stackable)** – extended functionality, support for stacking multiple HATs

4.2 Instance ID

Each BE-IIS HAT++ provides multiple **Instance ID**.
An **Instance ID** defines how the HAT is connected to the Raspberry Pi in terms of hardware resources, including:

- Chip-Select (SPI)
- Interfaces
- Interrupt signals
- I^{2}C target address

Each **Instance ID** represents a unique hardware configuration, allowing multiple HATs to operate in parallel without resource conflicts.

Selection:

- The active **Instance ID** is selected using the on-board push button
- The selected mode is indicated by LEDs on the right side of the PCB
- The selected mode is stored permanently 2 s after being set
- The selected mode becomes active after a power cycle

Purpose:

- Enables conflict-free stacking of multiple HATs
- Allows flexible system configuration
- Provides deterministic hardware resource mapping

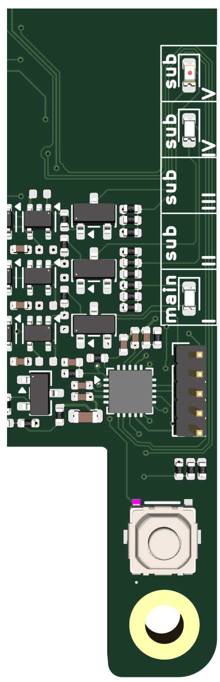


Figure 5: Instance ID indication on the PCB

Instance ID HW resources:

This board uses the following interfaces, either exclusively or shared with other HATs in the system:

Instance Mode	Signal	Pinname
I	CSN0	GP8
I	IRQ0	GP6
II	CSN1	GP7
II	IRQ1	GP5
III	CSN2	GP16
III	IRQ2	GP12

Table 4: Exclusive HW resources

Instance Mode	Signal	Pin
I & II & III	SCLK	GP11
I & II & III	MISO	GP9
I & II & III	MOSI	GP10
I & II & III & IV & V	RESET	GP13
I & II & III & IV & V	SCL0	GP1
I & II & III & IV & V	SDA0	GP0

Table 5: Shared HW resources

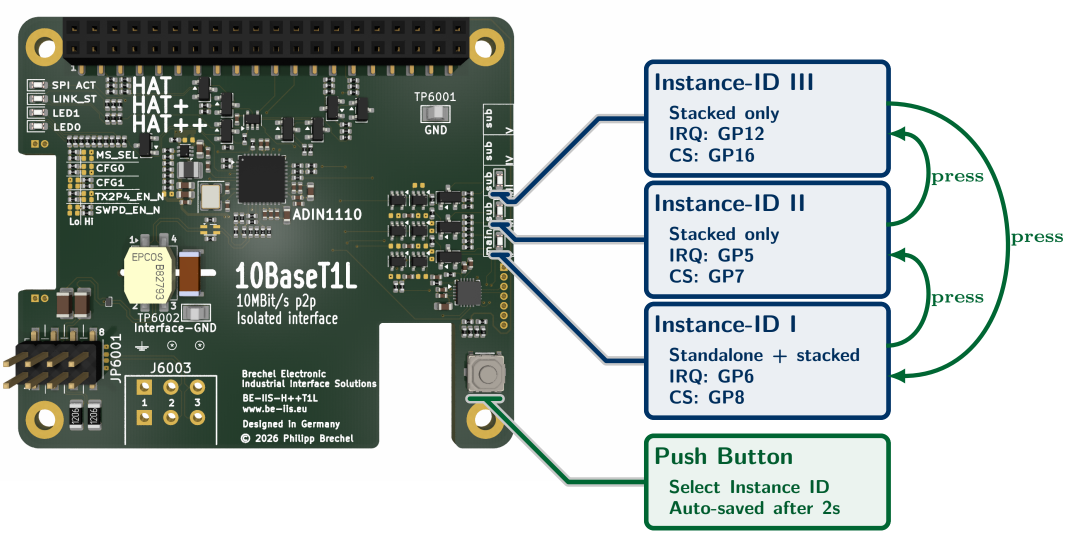


Figure 6: HAT Instance ID HW resources

4.3 Standalone and Stacked Operation

The board is designed to operate as part of the **HAT++ ecosystem**, enabling seamless combination with other HAT++ boards in a stacked configuration.
However, it can also be used as a standalone board, fully compatible with standard **Raspberry Pi HAT** / **HAT+** usage.
The following section provides a comparison of both operation modes.

Manual	Auto-detect (HAT++)
Configuration: • At least one HAT must be configured to **Instance ID I** to provide the required **I^2C pull-up resistors** in accordance with the **Raspberry Pi HAT specification** • The BE-IIS installer is not used, or • BE-IIS overlays are removed from `/boot/firmware/BE-IIS*` if the installer has been used previously • If the **HAT+ auto-detection mechanism** is used, the overlay name must match `BE-IIS-HPP-T1L-I` and the Instance ID must be I • Otherwise, a custom overlay must be provided and applied manually (e.g., via `config.txt` or systemd)	**Configuration:** • At least one HAT must be configured to **Instance ID I** to provide the required **I^2C pull-up resistors** according to the **Raspberry Pi HAT specification** and to enable the **auto-detection mechanism** • Each HAT must use a unique **Instance ID**
Software: • Device Tree overlays must be created and applied manually • Kernel modules must be built and installed manually • Practical examples and references can be found at [2]	**Software:** • Run the BE-IIS installer from GitHub [2] • Automatic overlay deployment and driver setup • All devices are enumerated and available after boot
Stacking: • Up to 5 HAT++ boards can be stacked • Each HAT must use a unique **Instance ID** • Alternatively, a custom **resource management** scheme can be implemented	**Stacking:** • Up to 5 HAT++ boards can be stacked • Each HAT must use a different **Instance ID** • **Instance ID I** must be used at least once
Result: • Full system control • Maximum flexibility • Highest integration effort	**Result:** • Automatic system integration • Scalable from single to multi-board setups • Minimal integration effort

5 Software and System Configuration

The BE-IIS-HAT++ system provides a unified platform for fast system integration.

- Predefined drivers and kernel modules
- Support for prebuilt modules and custom kernel builds
- Ready-to-use build and configuration scripts
- Centralized software repository [TODO]
- Typical setup time below a few minutes

After installation, the system can be used without further software modification.

5.1 System Support

The hardware is designed for use with Raspberry Pi platforms running a standard Raspberry Pi OS. All listed configurations have been validated or are expected to operate reliably with mainline Linux drivers.

- Supported Raspberry Pi platforms:
  - Raspberry Pi 3B, 3B+, 3A+
  - Raspberry Pi 4B
  - Raspberry Pi 5
  - Raspberry Pi Zero 2 W
- Supported Linux kernel versions:
  - $\geq$ **6.12**
  - Older kernel versions may work but are not officially supported
- Supported operating systems:
  - Raspberry Pi OS (64-bit)
  - Raspberry Pi OS Full and Lite
- Other Linux distributions:
  - Debian, Ubuntu, and other Linux distributions may work
  - Not tested or officially supported at this time

5.2 Driver & Integration

A standard Raspberry Pi OS installation is used as the base system. The provided installer configures all required components automatically, including kernel modules, Device Tree overlays, and systemd services.

- Prepare a Raspberry Pi hardware platform and operating system from the System Support list
- Run the provided Git-based installer
- Reboot the system to apply all configurations
- System supports the full BE-IIS-HAT++ portfolio

```
# install git
$ sudo apt install -y git

# clone installer
$ git clone https://github.com/be-iis/be-iis-installer.git

# enter directory
$ cd be-iis-installer

# run installer
$ ./scripts/install/install-all.sh
```

Drop-in CMD

```
sudo apt install -y git && cd  /Downloads && git clone https://github.com/be-iis/be-iis-installer.git && cd be-iis-installer && ./scripts/install/install-all.sh
```

After running the installer, a summary is printed to indicate the installation status and applied system changes.

Example output (shortened):

```
[INFO] Installation complete.
[INFO] Total scripts : 6
[INFO] Successful    : 6
[INFO] Failed        : 0

[INFO] Changes active after reboot:
[INFO]   - systemd service
[INFO]   - udev rules
[INFO]   - module autoload / runtime setup
Press ENTER to reboot now or CTRL+C to cancel...
```

5.3 Hardware–Software Interaction

The Instance ID can be changed during normal operation using the on-board control interface. The selected Instance ID is stored persistently after a short delay.

- Instance ID can be changed during runtime
- The selected Instance ID is stored persistently after approximately 2 seconds
- After changing the Instance ID, the interface becomes temporarily unavailable
- A system reboot restores full functionality with the updated configuration
- Instance ID **0** must be present at least once in the system
- In stacked configurations, each board must use a unique Instance ID

5.4 System Inspection

The system status and integration process can be inspected using standard Linux tools. All BE-IIS related services provide detailed runtime information via the system journal.

- View system integration logs:

```
# show BE-IIS system integration log
$ journalctl -b | grep BE-IIS
```

Example output (shortened):

```
BE-IIS Instance I   (0-0050): HAT detected -> BE-IIS-HPP-T1S-I
BE-IIS Instance II  (0-0060): HAT detected -> BE-IIS-HPP-CAN-SIC-II
BE-IIS Instance III (0-0070): HAT detected -> BE-IIS-HPP-LAN-III
BE-IIS Instance IV  (0-0074): HAT detected -> BE-IIS-HPP-UART-II
BE-IIS Instance V   (0-0076): HAT detected -> BE-IIS-HPP-MODBUS-III
BE-IIS HAT++ system integration complete.
```

Drop-in CMD

```
journalctl -b | grep BE-IIS
```

5.5 Interface Naming

All interfaces are assigned deterministic and persistent names using udev rules. This ensures stable device identification across reboots and different hardware configurations.

- Network interfaces are named based on function and instance index
- UART interfaces are exposed via symbolic links
- Naming is independent of kernel enumeration order
- udev rules location: `/etc/udev/rules.d/70-beiis-names.rules`

Example:

```
beiis-t1s0   # 10BASE-T1S interface
beiis-t1l0   # 10BASE-T1L interface
beiis-lan0   # Ethernet interface
beiis-can0   # CAN interface
beiis-uart0a # UART and MODBUS channel A
beiis-uart0b # UART and MODBUS channel B
```

```
$ cat /etc/udev/rules.d/70-beiis-names.rules
```

Drop-in CMD

`cat /etc/udev/rules.d/70-beiis-names.rules`

6 Electrical Characteristics

6.1 Supply Voltage

Parameter	Min	Typ	Max
3.3 V Input [V]	3.135	3.3	3.65
5 V Input [V]	4.5	5	5.5

Table 6: Voltage supply

6.2 Current Consumption

Parameter	Typ	Unit
Current @ 5 V	25	mA
Current @ 3.3 V	5	mA

Table 7: Current consumption

7 Environmental Conditions

7.1 Conditions

Condition	Min	Max
Operating Temperature [°C]	-40	+85
Storage Temperature [°C]	-40	+105
Relative humidity [%]	5	95

Table 8: Operating conditions

7.2 Usage

Condition	Parameter
Usage	indoor
Pollution degree	2
Operating altitude	up to 2000 m

Table 9: Operating usage

7.3 EMC and Environmental Compliance (Preliminary)

The standard version of the board is provided without formal EMC or safety certification.
The hardware design is developed with consideration of commonly applied IEC standards, including:

- **ESD immunity:** IEC 61000-4-2
- **Electrical fast transient (EFT/Burst):** IEC 61000-4-4
- **Surge immunity:** IEC 61000-4-5
- **Conducted RF immunity:** IEC 61000-4-6
- **Radiated RF immunity:** IEC 61000-4-3
- **EMC immunity (industrial):** IEC 61000-6-2
- **EMC emission (industrial):** IEC 61000-6-4
- **Safety / isolation reference:** IEC 62368-1

These standards are not verified for the standard product variant.
Compliance with specific standards, test levels, or safety requirements is not guaranteed unless explicitly specified.
If defined EMC or isolation requirements are provided, application-specific validation, testing, and certification can be supported. Upon request, product variants with validated performance, including labeling, certification, and test reports (e.g. Hi-Pot testing), can be delivered.

8 Delivery

The product is delivered as a partially assembled kit intended for final user assembly. Mechanical accessories and connector components required for standard evaluation and stacked operation are included.

Order Code	BE-IIS-HPP-T1L
Condition	Assembly kit
Status	Partially assembled
Included Items	1× BE-IIS-HPP-T1L-PCBA-FT 4× M2.5x16 mm spacers 1× 2×20 pin stackable header 2x Jumper 1x PCB Header (see 3.5) 1x PCB Connector (see 3.5)
REACH & RoHS	Compliant with EU Directive 2011/65/EU and REACH Regulation (EC) No 1907/2006

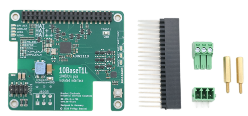


Figure 7: Delivery condition

9 Mechanical

9.1 Board Format

- Form factor: Raspberry Pi HAT+
- Mechanical dimensions: Raspberry Pi HAT compatible [9]
- Mounting hole pattern: Raspberry Pi HAT compatible [9]
- Stackszize: 16mm

9.2 3D Data

- Available on the BE-IIS product page [10]

10 Assembly

This product is delivered as a kit and requires basic soldering and mechanical assembly.

10.1 Assemble 2x20-Pin Main Connector

The 2x20-pin connector provides the interface to the Raspberry Pi. For proper HAT functionality, the connector must be assembled carefully.
A stackable 2x20-pin header is included in the delivery and is recommended for most applications, especially when using the BE-IIS HAT++ stacking system.
The header must be inserted between the Raspberry Pi and the HAT: the header is first mounted onto the Raspberry Pi, and the HAT is then plugged onto the header.

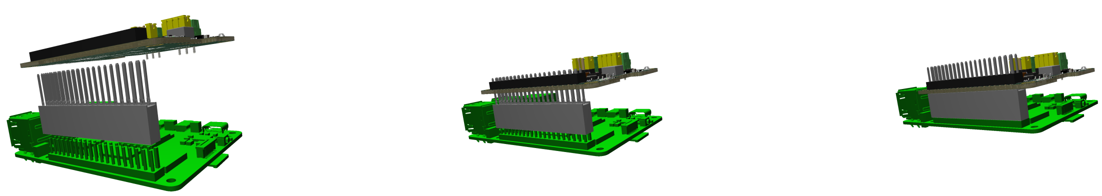


Figure 8: Assembly of the 2x20-pin stackable header between Raspberry Pi and HAT

10.2 Assemble Spacer

To ensure mechanical stability and correct stacking height, spacers must be installed.

- Recommended spacer height: see Section 9.1
- Fix the PCB using appropriate screws and spacers
- Ensure stable mechanical mounting to avoid stress on the connector

The spacers define the stacking distance and provide mechanical fixation of the HAT.

10.3 3.5 mm Terminal Block Connector

A suitable screw terminal block is typically included in the delivery. Alternatively, a compatible PCB header (plug or socket variant) may be used, depending on the application. Refer to the corresponding product section for supported connector types.
Assembly instructions:

- Ensure correct orientation before soldering: the cable entry openings must face outwards from the PCB edge
- Insert the connector fully into the PCB to ensure proper mechanical alignment
- Solder all pins carefully with sufficient wetting

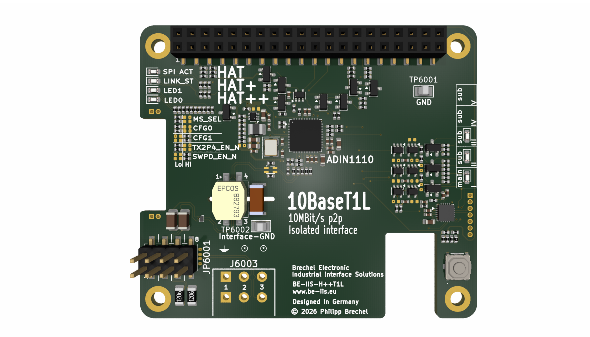
Important notes:

- Avoid direct contact between the soldering iron and the plastic housing of the connector, as this may cause visible damage or deformation
- Ensure clean solder joints without excessive solder to maintain proper mechanical fit

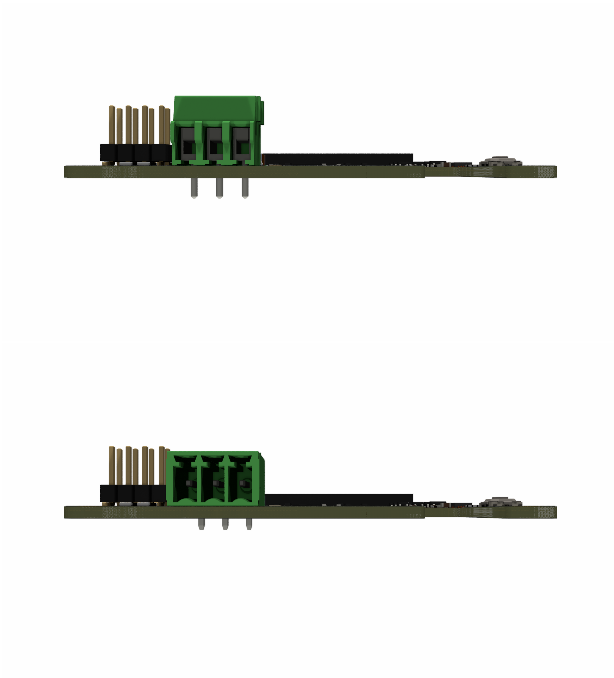
Correct assembly ensures reliable electrical contact and proper usability of the terminal interface.

10.4 Board Overview

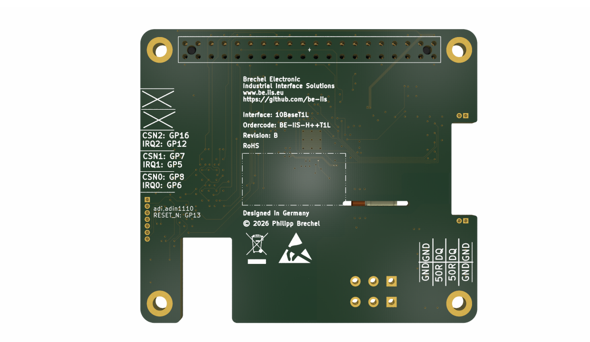
11 References



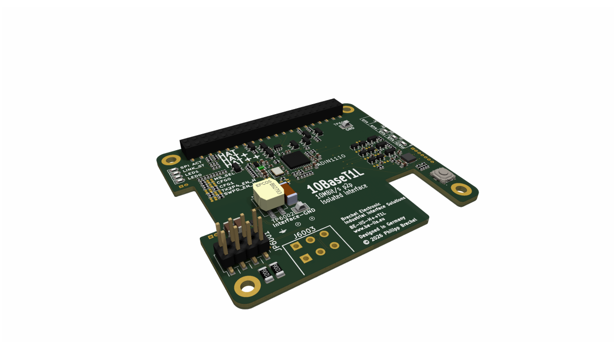
Top view



Front view



Bottom view



Side view

Figure 9: BE-IIS-HPP-T1L – mechanical overview

1. Analog Devices ADIN1110 product page
2. BE-IIS Installer (Software and Setup Tools)
3. PhoenixContact PT 1,5/ 3-3,5-H - PCB terminal block
4. Wurth Electronic - WR-TBL Series 2141 - 3.50 mm Horiz. Entry Modular
5. PhoenixContact - MC 1,5/ 3-G-3,5 - PCB header
6. PhoenixContact - MC 1,5/ 3-ST-3,5 - PCB connector
7. Wurth Electronic - WR-TBL 300VAC 12A 3P - PCB header
8. Wurth Electronic - WR-TBL 300VAC 10.5Am Vertical 26-16AWG
9. Raspberry Pi HAT+ Specification
10. Schematic, PCB-Viewer, BOM, 3D-Model
11. Power over Dataline extension board

12 Revision History

Revision	Date	Description
A.00	2026-02-12	Initial draft
B.00	2026-05-05	First version of B

Company Information

Manufacturer

BE-IIS welcomes technical feedback, suggestions and improvement ideas. Business inquiries, cooperation requests and distribution opportunities are welcome.

Name	Philipp Brechel
Company	Brechel Electronic
Business	Industrial Interface Systems
Address	Hindenburgstraße 100/1
	73207 Plochingen, Germany

Contact

Email	contact@be-iis.eu
Web	www.be-iis.eu
GitHub	github.com/be-iis
LinkedIn	linkedin.com/in/philipp-brechel
Languages	German and English

Brechel Electronic – Industrial Interface Systems

Designed by Philipp Brechel (Germany)