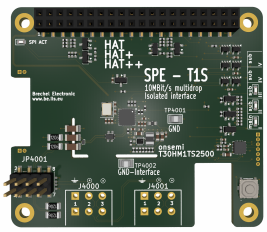
Isolated 10BASE-T1S Industrial Ethernet HAT++ for Raspberry Pi

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



Document ID: BE-IIS-HPP-T1S-ON_RevA_Datasheet
Revision: Rev. A
Date: October 19, 2026
Document Status: Released
Product Status: Active

10BASE-T1S Ethernet interface for Raspberry Pi with a data rate of 10 Mbit/s over a single twisted pair. The multidrop reference topology supports at least eight nodes over a bus length of at least 25 m.

The T30HM1TS2500 MAC-PHY from onsemi connects the SPI host interface to the capacitively coupled SPE bus. HAT++ instance selection assigns the chip-select and interrupt connections for use alongside other BE-IIS interface boards.

Key Features

- 10BASE-T1S compliant with IEEE 802.3cg
- T30HM1TS2500 10BASE-T1S MAC-PHY
- Capacitively coupled MDI
- Raspberry Pi 40-pin interface
- Stackable HAT (BE-IIS-HAT++)
- Configurable CS and IRQ routing
- Interface for an external PoSPE module
- Screw-terminal or pluggable connector
- 25 MHz reference crystal
- 3.3 V and 5 V host supply

Product Description

The HAT connects a Raspberry Pi to a shared 10BASE-T1S bus. Two parallel SPE connectors provide access to the bus, with jumper-selectable termination for end nodes.
The 40-pin header carries the SPI interface, control signals and power supply.

Applications

- Technology evaluation
- Industrial network evaluation
- Prototyping
- Education and laboratory use
- Switch and node evaluation

Contents

1 Introduction **4**
1.1 General . . . 4
1.2 Raspberry Pi Compatibility . . . 5
1.3 HAT Compatibility Level . . . 5
1.4 Introduction: Optional HAT++ Stacking . . . 6
1.5 Design Resources . . . 7

2 Hardware Configuration **8**
2.1 Main Features . . . 8
2.2 Block Diagram . . . 9
2.3 Hardware Components . . . 10
2.3.1 T30HM1TS2500 10BASE-T1S MAC-PHY . . . 10
2.3.2 HAT++ Control Logic . . . 10
2.4 Isolation . . . 11
2.5 Connectors . . . 12
2.6 Jumper and Configuration . . . 13
2.7 Indicators (LEDs) . . . 13
2.8 Signal Polarity and Wiring Orientation . . . 13
2.9 Power over Dataline / Power over SPE . . . 13

3 HAT++ Configuration **15**
3.1 Select the Board Configuration . . . 15
3.2 Combine Boards and Check Resources . . . 16
3.3 Install the Matching Software . . . 16

4 Software and System Configuration **17**
4.1 System Support . . . 17
4.2 Driver & Integration . . . 17
4.3 Hardware–Software Interaction . . . 18
4.4 System Inspection . . . 18
4.5 Interface Naming . . . 19
4.6 10BASE-T1S Driver Stack and Kernel Compatibility . . . 19

5 Delivery **21**

6 Mechanical **21**
6.1 Board Format . . . 21

6.2 3D Data 21

7 Assembly **21**

7.1 Assemble 2x20-Pin Main Connector 21

7.2 Assemble Spacer 22

7.3 3.5 mm Terminal Block Connector 22

7.4 Board Overview 23

8 References **23**

9 Revision History **24**

Company Information **25**

1 Introduction

1.1 General

The onsemi T30HM1TS2500 MAC-PHY [1] connects the Raspberry Pi SPI interface to a 10BASE-T1S network. The board supports 10 Mbit/s Ethernet communication using PLCA and CSMA/CD. The SPI interface handles data transfer and device configuration, while an interrupt signal provides event notification.

Two parallel SPE connectors allow integration into multidrop networks. Selectable termination enables operation at the physical ends of the bus segment. The HAT++ configuration assigns one of three available chip-select and interrupt pairs.

An optional Power over Single Pair Ethernet (PoSPE) daughterboard enables DC power injection into or extraction from the SPE bus. This allows Ethernet communication and power transmission over the same twisted-pair cable.

A terminal-based installer simplifies software integration by automatically configuring the required Linux drivers, Device Tree overlays and system services.

As part of the BE-IIS HAT++ family, the board can be combined with other interface boards in a single Raspberry Pi system. Supported combinations include CAN FD, Modbus, 10BASE-T1L and other industrial communication interfaces.

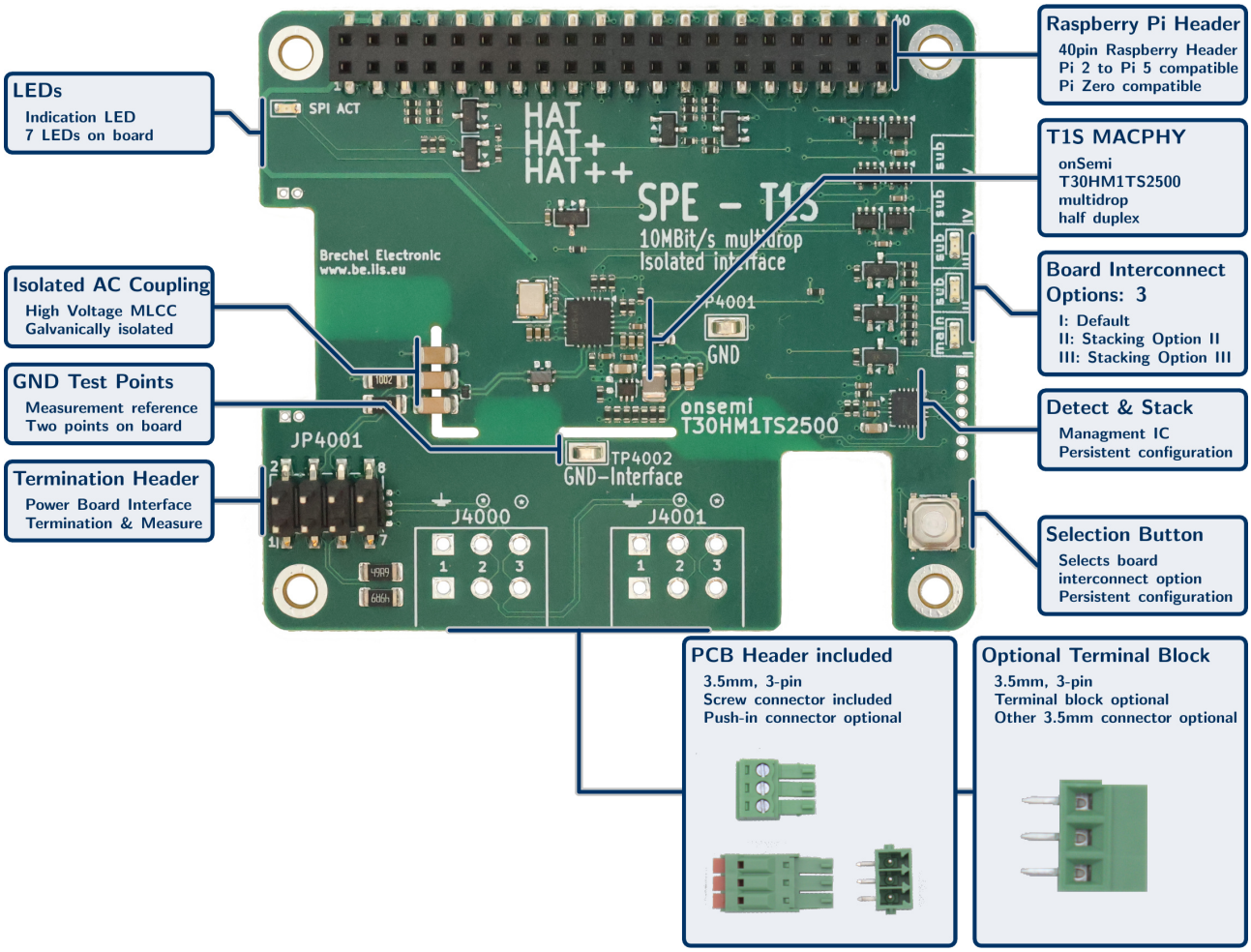


Figure 1: BE-IIS-HPP-T1S-ON – Annotated hardware features

1.2 Raspberry Pi Compatibility

The board follows the Raspberry Pi HAT+ mechanical and electrical interface concept. Compatibility depends on the selected Raspberry Pi model, operating system architecture, enabled interfaces, and the installed BE-IIS software configuration.

Model	32-bit	64-bit
Raspberry Pi 2	✓	
Raspberry Pi 3	✓	✓
Raspberry Pi 4	✓	✓
Raspberry Pi 5	✓	✓
Raspberry Pi Zero	✓	
Raspberry Pi Zero 2 W	✓	✓

Table 1: Raspberry Pi compatibility

Note: Compatibility depends on the installed operating system, kernel version and enabled interfaces.“‘

1.3 HAT Compatibility Level

Table 2 describes the HAT compatibility level of the board.

Level	Description	Support	Configuration
HAT	The board can be used as a standard Raspberry Pi HAT.	✓	3 configurable hardware configurations
HAT+	The board supports the Raspberry Pi HAT+ auto-detection mechanism.	✓	Instance ID 1 configurable
HAT++	The board supports detection and stackable operation according to the BE-IIS HAT++ specification.	✓	3 configurable hardware configurations

Table 2: HAT compatibility level

1.4 Introduction: Optional HAT++ Stacking

HAT++ is an extended function of the Raspberry Pi HAT+ concept. It enables stacking of up to five different HATs in one system. Depending on the supported instance modes, the same HAT type can be used up to three times in one stack.

Figure 2 gives a short overview of the required hardware configuration and the supported instance modes.

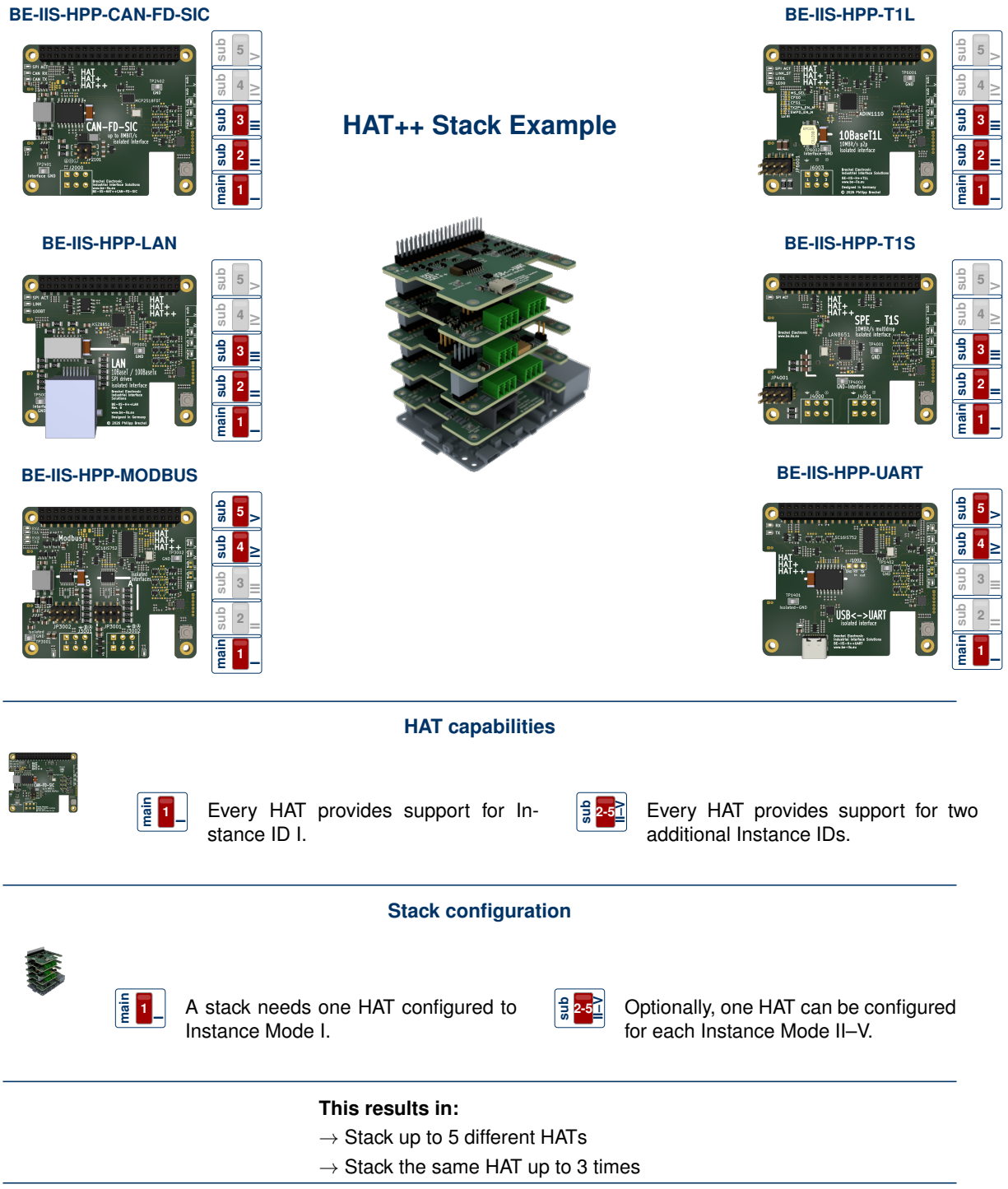


Figure 2: HAT++ stacking overview and instance mode configuration.

1.5 Design Resources

All design files and software resources are publicly available.

HTML **Product Page**
https://www.be-iis.eu/products/BE-IIS-HPP-T1S-ON_A/

PDF **Datasheet (PDF)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1S-ON_A/datasheet.pdf

PDF **Schematic (PDF)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1S-ON_A/schematic.pdf

HTML **Layout & BOM (Interactive)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1S-ON_A/ibom.html

STL **3D Model (STEP/STL)**
https://www.be-iis.eu/products/BE-IIS-HPP-T1S-ON_A/model.zip

GIT **GitHub Repository**
https://github.com/be-iis/be-iis-docs/tree/main/B-Sample/BE-IIS-HPP-T1S-ON

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

DigiKey
Coming soon

SELL ON TINDIE
Coming soon

BUY IT ON LECTRONZ
Coming soon

2 Hardware Configuration

2.1 Main Features

The T30HM1TS2500 integrates the Ethernet MAC and 10BASE-T1S PHY. It connects to SPI0 on the 40-pin host header and uses a 25 MHz crystal as its reference clock. The selected HAT++ instance determines the chip-select and interrupt connections.

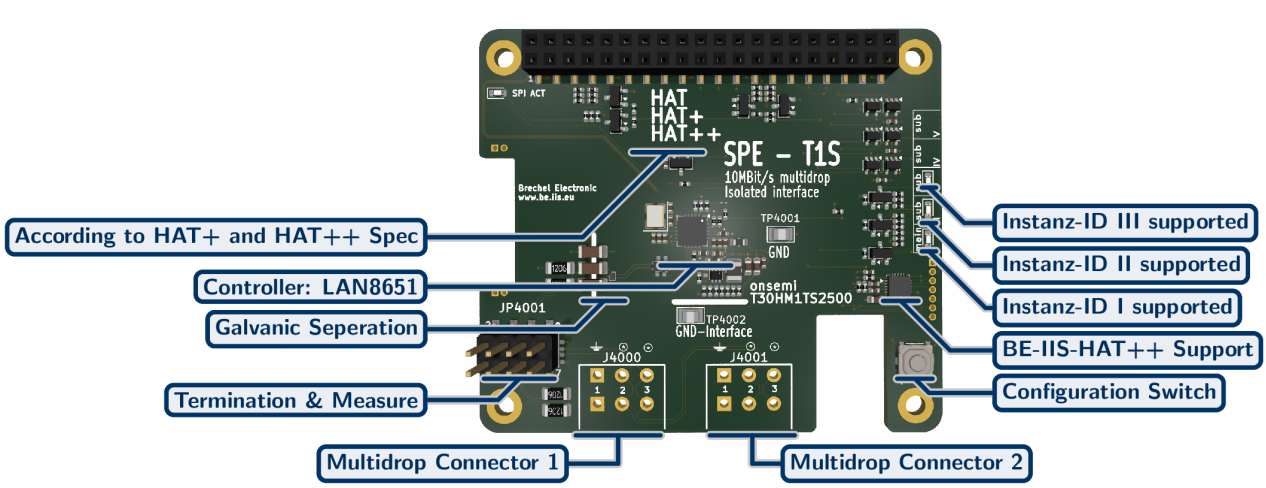


Figure 3: Annotated board overview (reference illustration)

MAC-PHY

- onsemi T30HM1TS2500
- 10BASE-T1S MAC-PHY
- 25 MHz crystal

SPI Interface

- Three selectable CS/IRQ pairs
- Persistent selection via push button

Isolation

- Capacitively coupled MDI
- Separate field-side shield domain

Power over Dataline

- External PoSPE module interface
- No onboard PoSPE power converter

Protocol Support

- 10BASE-T1S, 10 Mbit/s
- PLCA and CSMA/CD

Bus Topology

- Multidrop single-pair bus
- 8 nodes / 25 m reference topology
- Two parallel SPE connectors

Instance Configuration

- PLCA node ID and node count via software
- Termination at the two bus ends

2.2 Block Diagram

The board comprises the SPI-connected MAC-PHY, the SPE coupling and termination network, and the HAT++ control circuit. The SPE connectors share one bus connection; the HAT++ circuit routes chip-select and interrupt signals according to the selected instance.

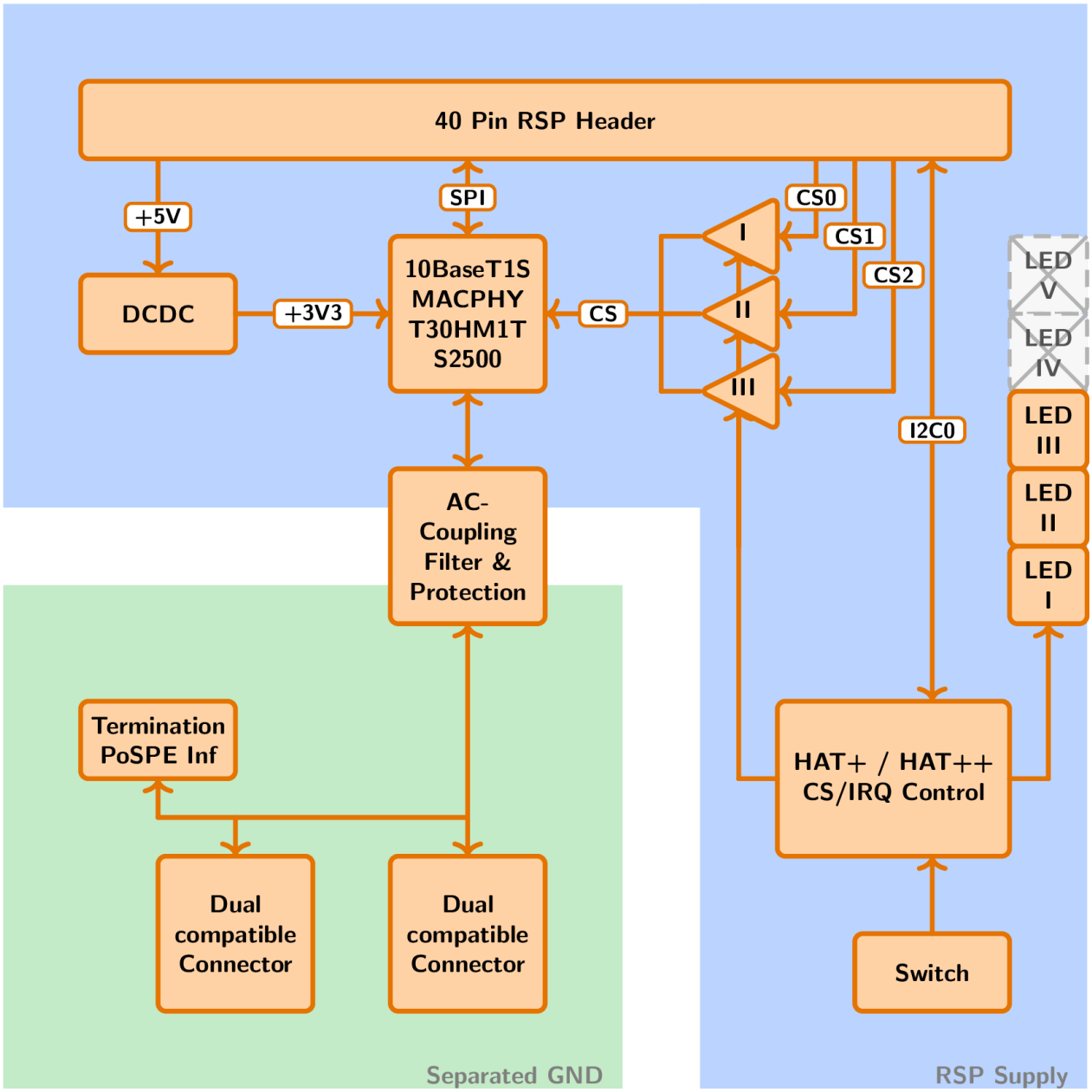


Figure 4: Block diagram (T1S reference design)

2.3 Hardware Components

The MAC-PHY handles Ethernet communication, while the HAT++ controller manages board identification and resource selection. J4000 and J4001 provide parallel bus connections. The termination network connects to JP4001 for configuration at the bus ends.

2.3.1 T30HM1TS2500 10BASE-T1S MAC-PHY

A key advantage of the T30HM1TS2500 is its **comprehensive network diagnostic and monitoring functionality**.

The device integrates several advanced physical-layer diagnostic capabilities, including **Signal Quality Indicator (SQI)**, **Harness Defect Detection (HDD)**, **Topology Discovery**, and **PLCA diagnostics**.

These features enable detailed monitoring of network conditions and support the identification of cabling faults, signal degradation, and network configuration problems.

onsemi places particular emphasis on **automotive-grade robustness, electromagnetic immunity, and advanced network management capabilities**.

These capabilities make the T30HM1TS2500 particularly attractive for industrial applications requiring advanced network diagnostics, reliable communication, and simplified maintenance.

The availability of individual diagnostic functions under Linux depends on the installed driver and kernel version (see section 4.6). Electrical characteristics and register descriptions are provided in the device datasheet [1].

2.3.2 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^2C0 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^2C addresses for I^2C-based HATs
- Selection of SPI chip select (CS) for SPI-based HATs
- IRQ routing and handling
- Enable I^2C pull-ups in Instance ID I
- HAT-specific control functions

For details on the HAT++ system, see section 3.

2.4 Isolation

The SPE pair is capacitively coupled through C4023 and C4026. C4019 couples the field-side shield ground, GNDSHD, to host ground. All three capacitors are rated at 100 nF / 450 V; the bypass positions R4020 and R4090 are unpopulated.

The 450 V value is the capacitor rating. It does not specify the withstand voltage of the assembled board.

Board-level withstand voltage and insulation certification are not specified for Rev. A.

RefDes	Supplier	MPN	Description
C4023, C4026, C4019	Various	Various	MLCC, 450 V, X7R

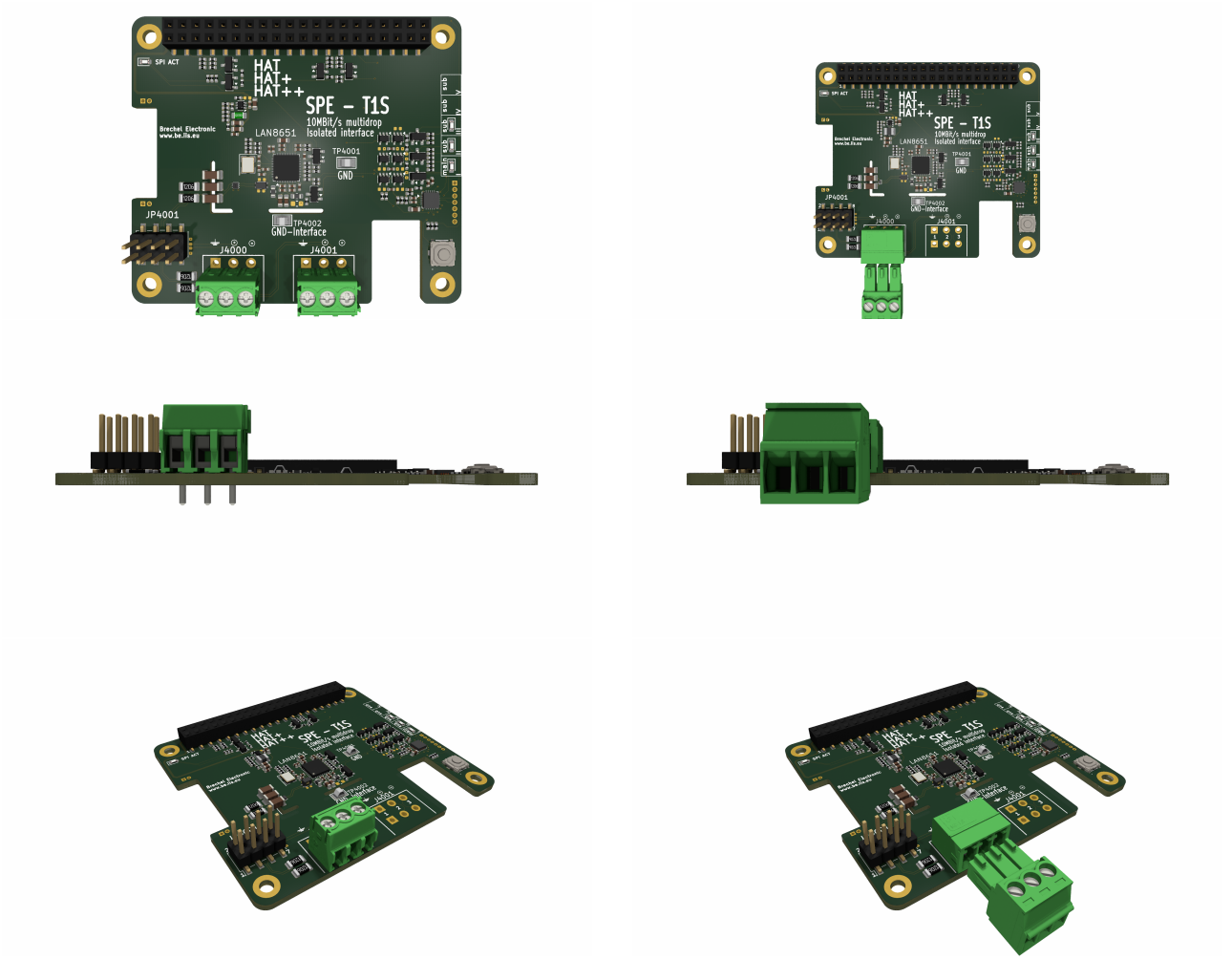
Table 3: Isolation Components



Figure 5: Isolation Barrier

2.5 Connectors

J4000 and J4001 provide parallel connections to the same SPE bus. Both are 3-pin connector positions with 3.5 mm pitch: pin 1 is SHIELD (GNDSHD), pin 2 is SPE– and pin 3 is SPE+. The fitted connector type and number of populated positions depend on the assembly option.



Screw-terminal option:
3-pin, 3.5 mm terminal block.
Würth Elektronik 691214110003 or mechanically compatible alternative.
Pinning: 1: SHIELD, 2: SPE–, 3: SPE+.

Pluggable option:
3-pin, 3.5 mm PCB header and mating plug.
Würth Elektronik 691305140003 / 691361100003 or compatible alternative.
Pinning: 1: SHIELD, 2: SPE–, 3: SPE+.

2.6 Jumper and Configuration

The bus termination is configured at JP4001 using two shunts.
Termination is required at the two physical ends of the bus segment. Intermediate nodes operate without termination.
End nodes: shunts 3–4 and 5–6 installed (termination enabled)
Intermediate nodes: both shunts removed (termination disabled)
In **point-to-point operation**, termination is enabled on both connected devices.
The two 49.9 Ω resistors, R4022 and R4023, provide a differential termination of 99.8 Ω.

Pin	Signal	Pin	Signal
1	Shield	2	Shield
3	Termination	4	SPE–
5	Termination	6	SPE+
7	Shield	8	Shield

Table 4: JP4001 pin assignment

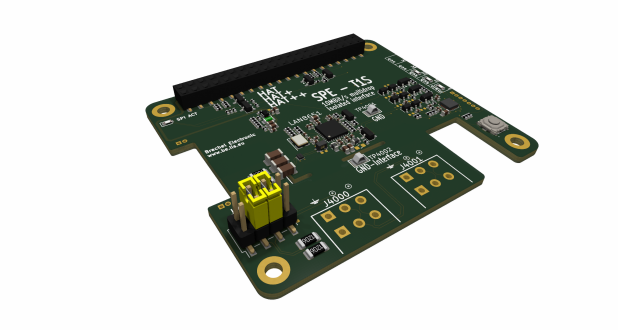


Figure 6: JP4001 termination jumper configuration with termination enabled

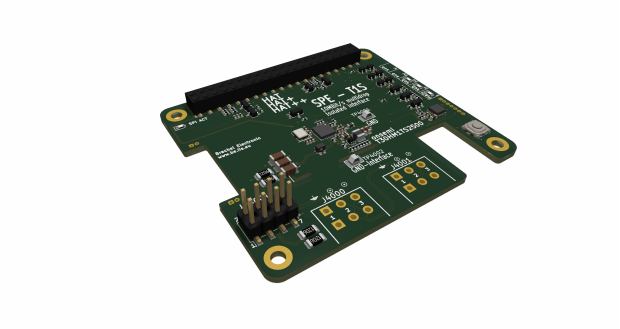


Figure 7: JP4001 termination jumper configuration with termination disabled

2.7 Indicators (LEDs)

The side LED bar displays the selected Instance ID. D4001 (SPI ACT) indicates chip-select activity on the host interface, independently of the Ethernet link state.

2.8 Signal Polarity and Wiring Orientation

SPE– is assigned to pin 2 and SPE+ to pin 3. Use a consistent assignment along the bus segment. Pin 1 provides the shield connection where required by the installation.

2.9 Power over Dataline / Power over SPE

The BE-IIS-HPP-T1S-ONSEMI supports an optional external Power over Single Pair Ethernet (PoSPE) module. The 10BASE-T1S HAT itself contains no power injection or power-conversion circuitry. Its JP4001 interface provides access to the SPE pair and shield for an external power interface; the Ethernet communication function remains on the base HAT.

The BE-IIS-HPP-PoSPE Rev. B accessory provides DC power coupling to the SPE wiring through a low-pass filtering network. This allows power to be transferred alongside the SPE data signals without requiring a separate supply cable. The accessory supports both power-source (PSE) and powered-device (PD) operation.

Power-source operation (PSE). In source mode, the PoSPE accessory couples power from an

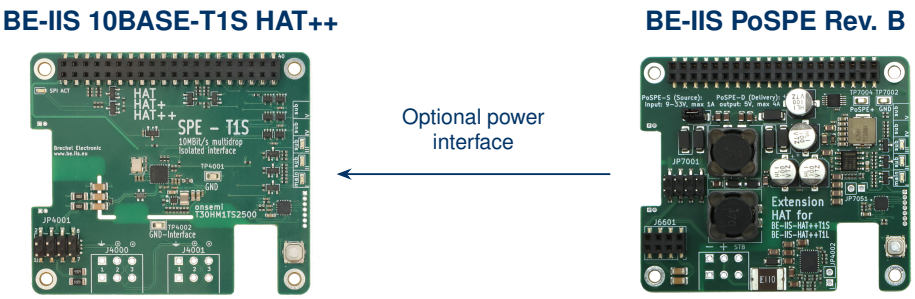


Figure 8: BE-IIS 10BASE-T1S ONSEMI HAT++ and optional BE-IIS-HPP-PoSPE Rev. B accessory. The accessory provides power coupling and conversion; the Ethernet HAT does not include a PoSPE converter.

external DC supply onto the SPE line. The specified operating voltage range of the accessory is 9–33 V. The published SPE current capability is up to 1 A. Available power at downstream nodes depends on the supply voltage, cable losses, connected loads and the selected topology.

Powered-device operation (PD). In delivery mode, the accessory extracts DC power from the SPE line and provides a regulated 5 V output for the Raspberry Pi supply path. The published output-current capability is up to 4 A. This output rating must not be interpreted as guaranteed power availability at an arbitrary cable length or at every node: the input-power budget and cable losses must be considered for each system.

Protection and monitoring. According to the accessory's product specification, the hardware includes overvoltage, undervoltage and overcurrent protection, reverse-polarity protection, and ESD/-transient protection. Current sensing is accessible through an I^2C interface. These functions belong to the optional PoSPE accessory, not to the onsemi S2500 Ethernet interface.

Integration note. Select the SPE termination according to the physical network topology as described in Section 2.6; PoSPE does not change the requirement for bus-end termination. Ensure that source and load configuration, power budget, wiring, and shielding follow the documentation of the selected accessory. BE-IIS-HPP-PoSPE Rev. B is currently listed as a development product. For current electrical specifications and documentation, see https://www.be-iis.eu/products/BE-IIS-HPP-PoSPE_B/.

3 HAT++ Configuration

HAT++ lets compatible expansion boards share one Raspberry Pi by assigning each board a defined set of hardware resources. Select the configuration with the on-board push button; the matching software uses that selection to configure the board. The board can operate individually or in a stack.

Quick start: one board

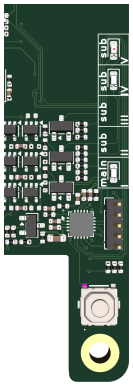
Select **Instance ID I**, leave the selection unchanged for at least **2 seconds**, then power-cycle the system. Use the BE-IIS installer to install the required drivers and overlays, and follow its reboot instructions.

3.1 Select the Board Configuration

An **Instance ID** selects the board's identification data and hardware routing, including the applicable chip-select, interrupt or peripheral address. In the HAT++ specification, the global position I–V is called the **Instance ID**. It describes resource use, not the board's physical position in the stack.

1. With power applied, press the button to cycle through the supported IDs.
2. Read the selected ID from the LED indication.
3. Wait at least **2 seconds** after the last change for it to be saved.
4. Switch the system off and on before using the new configuration.

The selection is retained without power. A button press does not reconfigure the running Linux system; a power cycle is required after a change.
The annotated image below shows this board's supported IDs, button sequence and resource assignments.



ID LEDs and selection button

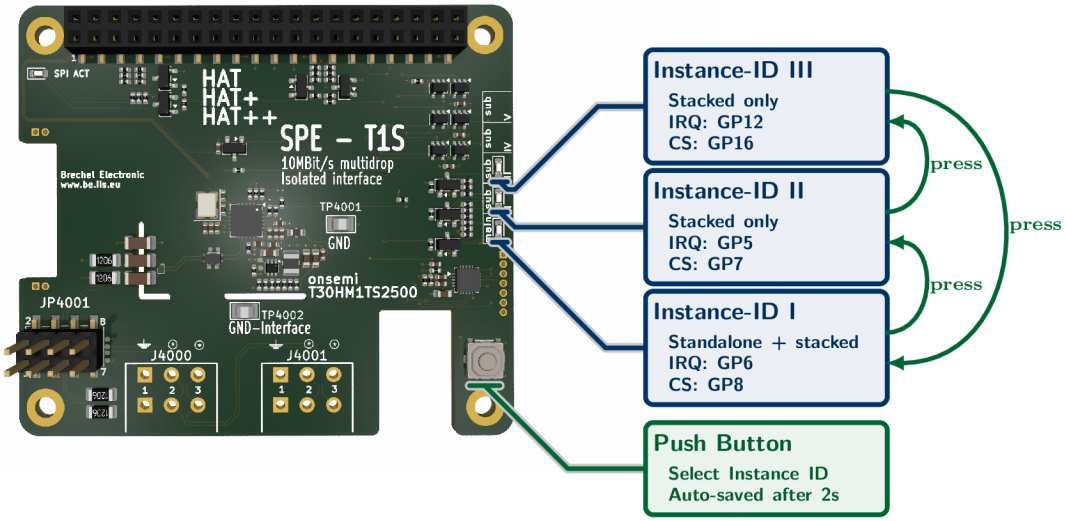


Figure 9: Board configuration: supported IDs, hardware routing and push button.

Overlay names: a suffix I/II/III may identify a board's local variant rather than its global Instance ID. For example, UART variant II uses Instance IV in the BE-IIS reference profile. Use the documented mapping or automatic detection; do not infer the Instance ID from the suffix alone.

3.2 Combine Boards and Check Resources

For automatic HAT++ operation, assign **Instance I to exactly one board**. It provides the identification-bus pull-ups and the primary configuration. Assign a different supported Instance ID to every additional board.

There are five global positions, I–V, and each board supports three of them, including I. A compatible stack can contain up to five boards; a product with three supported positions can be used at most three times. Check each product's supported positions before choosing a combination.

Resources used by this board

The tables below show the resources for each supported configuration. Exclusive signals must have only one owner. Shared signals require compatible electrical behaviour and coordinated use. GP/GPIO values are BCM GPIO numbers, not physical header-pin numbers.

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

Table 5: Exclusive HW resources

Instance Mode	Signal	Pin
I–III	SCLK	GP11
I–III	MISO	GP9
I–III	MOSI	GP10
All	RESET	GP13
All	SCL0	GP1
All	SDA0	GP0

Table 6: Shared HW resources

Unique Instance IDs coordinate the declared resource assignments; they do not replace checking additional GPIO use, supply requirements or mechanical clearance. Consult the other boards' datasheets when planning a stack.

3.3 Install the Matching Software

Automatic setup. Use the BE-IIS installer and the installation instructions in this datasheet. It deploys the supported overlays and driver setup. Raspberry Pi firmware handles Instance I; the BE-IIS Linux service applies the secondary-instance overlays. After boot, check that the required interfaces are present before starting the application.

Manual integration. Use the resource tables to select the matching Device Tree overlay and install any required kernel drivers. In Instance I, standard Raspberry Pi EEPROM-based detection can load `BE-IIS-HPP-T1S-ON-I` when the overlay is installed. Secondary instances require HAT++ host integration or explicit manual configuration. When changing from automatic to manual setup, disable the automatic loader and remove duplicate overlay configuration.

HAT++ extends the Raspberry Pi HAT+ concept. Instance I is the HAT+-compatible configuration; secondary-instance discovery is provided by HAT++ host software.

Reference: HAT++ Specification

HAT++ Specification – PDF Version 1.0, Rev. A, 30 September 2026 (technical review draft). See Sections 2–4 for Instance IDs, selection and resources, and Sections 7–8 for boot integration and diagnostics. The specification defines the common architecture; this datasheet defines this board's assignments and controls. Specification page and publication status.

4 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
- Typical setup time below a few minutes

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

4.1 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 2
  - Raspberry Pi 3B, 3B+, 3A+
  - Raspberry Pi 4B
  - Raspberry Pi 5
  - Raspberry Pi Zero, Zero W, Zero 2 W
- Supported Linux kernel versions:
  - $\geq$ **6.12**
  - BE-IIS aims to support Linux kernel 6.12 and newer across its product portfolio, provided that the required manufacturer drivers and kernel interfaces are available or can be adapted.
  - Older kernel versions may work but are not officially supported.
- Supported operating systems:
  - Raspberry Pi OS (32-bit and 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

4.2 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 ~ && 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...
```

4.3 Hardware–Software Interaction

The Instance ID can be selected during normal operation using the on-board control interface. The selected configuration is stored persistently after approximately 2 seconds.

The new hardware configuration becomes active after a power cycle. Changing the Instance ID does not automatically reconfigure the running Linux system.

- Instance ID selection is possible during runtime
- Configuration is stored persistently after approximately 2 seconds
- A power cycle is required to activate the new configuration
- Exactly one board must be configured as Instance I for automatic HAT++ operation
- Each additional board must use a unique, supported Instance ID

For detailed configuration instructions, see Section **??**.

4.4 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
```

4.5 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
```

4.6 10BASE-T1S Driver Stack and Kernel Compatibility

The Linux software stack for 10BASE-T1S is evolving rapidly. To support multiple Raspberry Pi platforms and Linux kernel versions, the BE-IIS installer selects a suitable OA-TC6 driver stack and, where necessary, applies compatibility adaptations before building the kernel modules.

Upstream driver sources Where the required functionality is not provided by the target kernel, the installer can prepare driver sources based on the upstream *onsemi S2500 MAC-PHY support v8* patch series (Message-ID: `20260928-s2500-mac-phy-support-v8-0-7e011aacc309@onsemi.com`). The resulting common OA-TC6 stack is used with the supported onsemi S2500, Analog Devices ADIN1140, and Microchip LAN865x drivers. If suitable drivers are already available in the running kernel, the installer may use those instead of rebuilding them.

Kernel-specific adaptations The installer selects the source baseline and required compatibility changes according to the detected kernel and its available APIs. These adaptations may include the OA-TC6 initialization interface, network-device receive-mode callbacks, PHY loopback interfaces, and Clause 45 management register access. On older kernels, asynchronous receive-mode handling may be replaced by a workqueue-based implementation.

Diagnostic limitations Not every diagnostic feature available in newer upstream drivers is supported by older kernel APIs. In particular, the NCN26000 PHY driver's Signal Quality Indicator (SQI) and cable-test callbacks may be omitted when the required kernel helpers are unavailable. Consequently, the available diagnostics depend on the installed driver and kernel combination. The installer reports applicable compatibility decisions and known limitations during source preparation and installation, where supported.

For current driver support and installation details, see the BE-IIS installer repository.

5 Delivery

The scope of delivery depends on the ordered assembly option. The kit configuration specifies the fitted connectors, host header and mechanical accessories.

Order Code	BE-IIS-HPP-T1S-ON
Condition	Assembly kit
Status	Partially assembled
Included Items	1× BE-IIS-HPP-T1S-ON-PCBA-FT 4× M2.5x16 mm spacers 1× 2×20 pin stackable header 2x Jumper 1x PCB Header (see 2.5) 1x PCB Connector (see 2.5)
REACH & RoHS	Compliant with EU Directive 2011/65/EU and REACH Regulation (EC) No 1907/2006

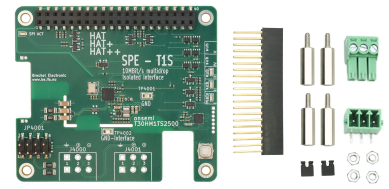


Figure 10: Delivery condition

DigiKey

Coming soon

SELL ON TINDIE

Coming soon

BUY IT ON LECTRONZ

Coming soon

6 Mechanical

6.1 Board Format

- Form factor: Raspberry Pi HAT+
- Mechanical dimensions: see the Rev. A PCB / 3D model
- Mounting-hole pattern: Raspberry Pi HAT interface
- Stack height: defined by the fitted header and spacers

6.2 3D Data

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

7 Assembly

Kit versions require assembly of the host header, spacers and connector before operation. The following instructions describe the standard assembly sequence.

7.1 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.

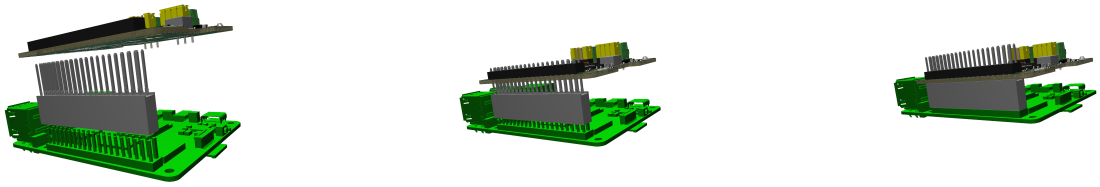


Figure 11: Assembly of the 2x20-pin stackable header between Raspberry Pi and HAT

7.2 Assemble Spacer

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

- Recommended spacer height: see Section 6.1
- 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.

7.3 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.

7.4 Board Overview

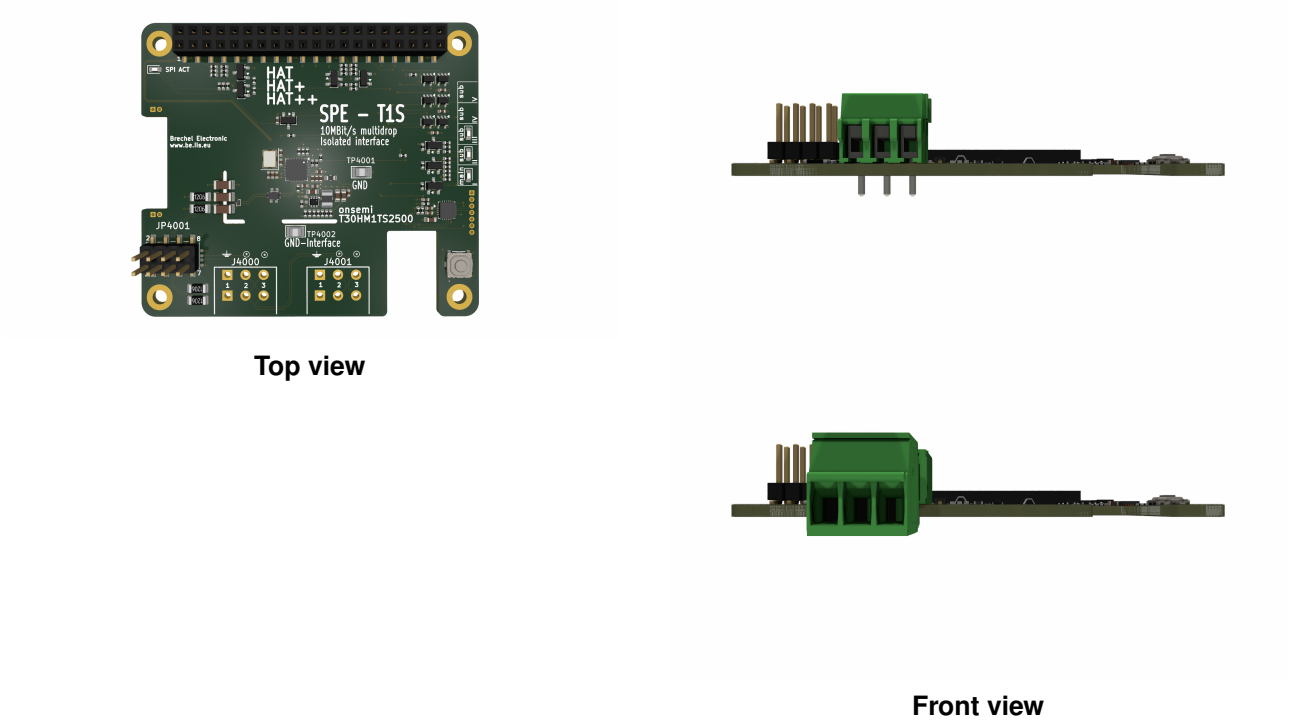


Figure 12: BE-IIS-HPP-T1S-ON – mechanical overview

8 References

1. onsemi T30HM1TS2500 datasheet
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. BE-IIS HPP Specification

9 Revision History

Revision	Date	Description
A.00	2026-09-30	Initial Rev. A draft
A.01	2026-09-30	Restore T1S template layout and figures
A.02	2026-09-30	Editorial revision; structure unchanged
A.1	2026-10-19	First Document Release

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 Solutions
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/brechel`
Digikey	`https://www.digikey.de/de/supplier-centers/industrial-interface-solutions`
Languages	`German and English`