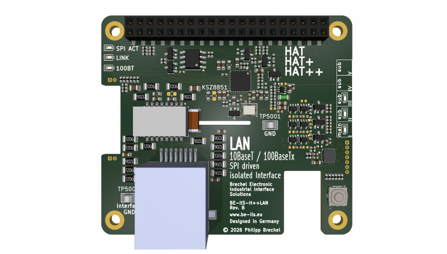
Isolated 100BASE-TX Industrial Ethernet HAT++ for Raspberry Pi



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

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

Standard Ethernet 10BASE-T / 100BASE-TX (10/100 Mbit/s) enables seamless integration into existing LAN infrastructures for industrial and embedded systems.

This HAT enables LAN communication based on the KSZ8851 Ethernet controller 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-T / 100BASE-TX Ethernet, up to 100 Mbit/s
- KSZ8851 Ethernet controller (SPI)
- Integrated MAC + PHY
- RJ45 interface
- EEPROM for MAC-ADR
- RSP HAT+ compliant (2024)
- Stackable (BE-IIS-HAT++)
- Configurable CS & IRQ
- Auto-Negotiation support
- Compatible with standard LAN infrastructure
- RoHS compliant
- Quality component suppliers

Product Description

The Ethernet Industrial HAT is a Raspberry Pi HAT+ compliant interface board providing standard 10BASE-T / 100BASE-TX connectivity according to IEEE 802.3.
It integrates a KSZ8851SNLI-TR Ethernet controller with SPI interface, combining MAC and PHY in a single device for direct connection to standard LAN infrastructure.

Applications

- Ethernet / LAN network evaluation
- Debug Interfaces
- Embedded Ethernet prototyping
- Industrial network integration
- Gateway and protocol converter development
- Education and laboratory use

Contents

1 Introduction

The BE-IIS HAT++ Ethernet Industrial HAT is a Raspberry Pi HAT+ compliant interface board providing Ethernet connectivity for industrial and laboratory environments.

The board integrates a KSZ8851 [1] SPI Ethernet controller, enabling a **10/100BASE-TX** Ethernet interface via the SPI bus. Galvanic isolation separates the logic domain from the Ethernet interface.

Ethernet enables high-bandwidth, packet-based communication and is widely used as a standard interface in industrial and IT network infrastructures. Its compatibility with existing network technologies allows seamless integration into established systems, while supporting flexible and scalable architectures.

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**, **10BASE-T1L**, **Ethernet (SPI-based)**, **Modbus/RS-485**, **CAN-FD**, and additional communication and measurement interfaces.

The HAT is compatible with a wide range of Raspberry Pi platforms, including **Raspberry Pi 2, 3, 4, and 5**, as well as **Raspberry Pi Zero and Zero 2 W**.

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.

The HAT can be used for evaluation, prototyping, industrial network integration, test setups, and educational purposes.

If you intend to use the HAT in a commercial product, please contact Brechel Electronic to adapt and optimize the design according to your specific requirements.

2 Design Resources

All design files and software resources are publicly available.

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

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

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

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

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

3.2 Main Features

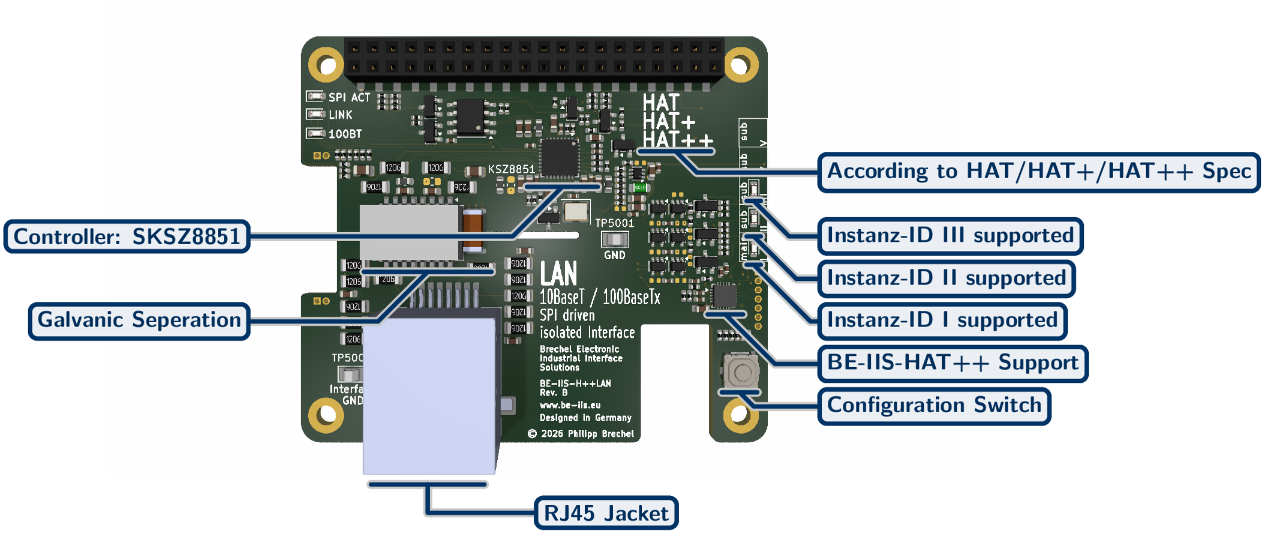


Figure 1: BE-IIS-LAN top view with annotations

Device

- Supplier: Microchip
- KSZ8851 10/100 Ethernet controller (MAC + PHY)
- Integrated MAC and PHY in single device (MACPhy)
- Mainline Linux kernel driver support

SPI Interface

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

Isolation

- Galvanic decoupled MDI
- Performed by YAGEO isolation Transfomer
- Spezified up to 1500 V DC for 60 s
- EMI-robust differential signaling

Protocol Support

- IEEE 802.3 compliant (10BASE-T / 100BASE-TX)
- Auto-Negotiation suppor
- Hardware checksum offload (IP/TCP/UDP)
- Full- and Half-Duplex operation

Bus Topology

- Star topology using Ethernet switches
- Scalable node count depending on network infrastructure
- Long cable lengths up to 100 m

EEPROM populated

- MAC address storage
- Programmable via `ethtool`

3.3 Block Diagram

The block diagram shown in Figure 2 is simplified. It illustrates the power domains, isolation barriers, main functional blocks, and principal signal paths.

The interrupt signal routing is not shown. It is configured using the same scheme as the chip-select (CS) routing.

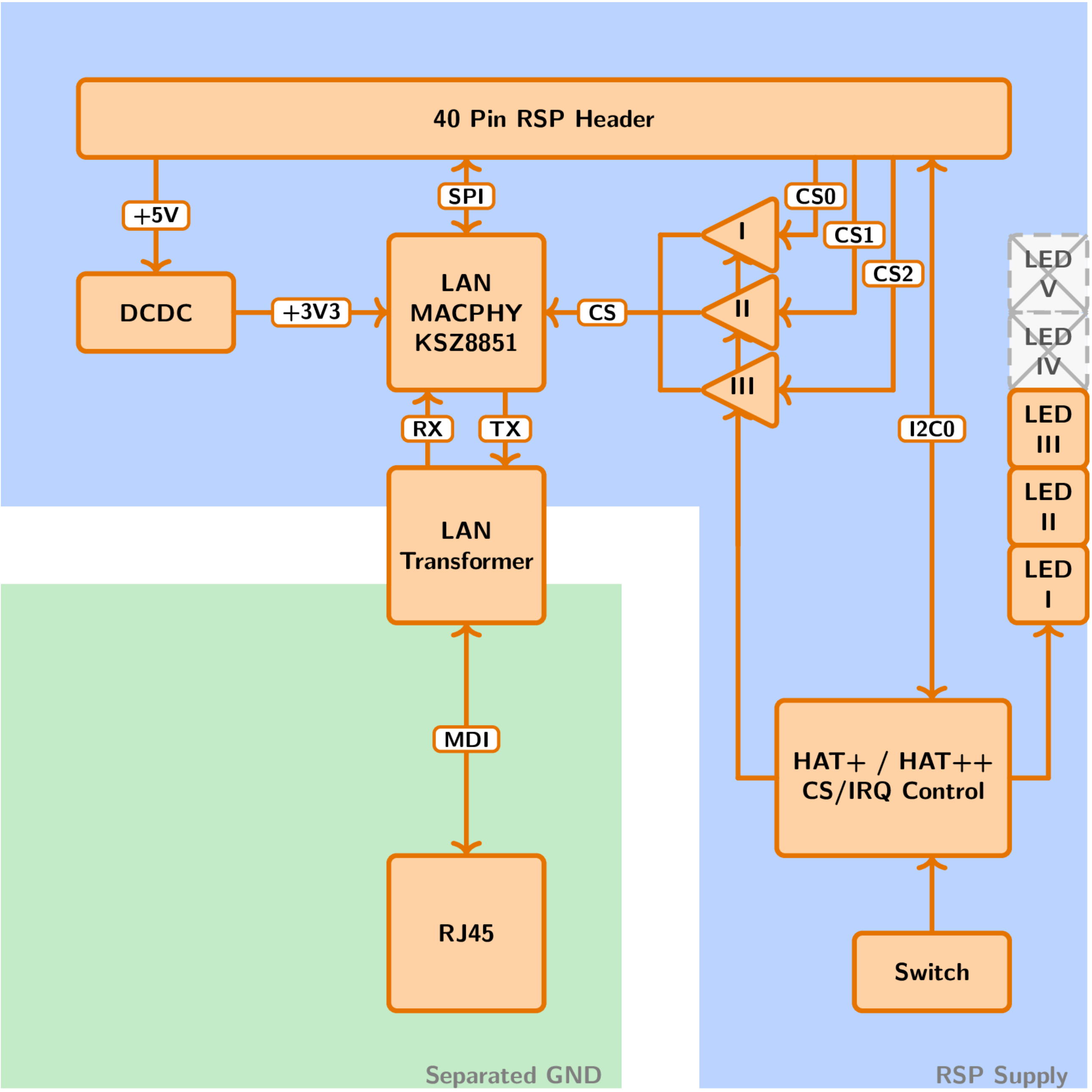


Figure 2: Simplified block diagram

3.4 Hardware Components

The selected hardware components and their interconnection are optimized to support **10/100BASE-TX Ethernet communication** via SPI, including **galvanic isolation** of the Ethernet interface. The design is fully aligned with the **HAT++ ecosystem**, enabling modular integration and conflict-free operation in stacked configurations.

3.4.1 KSZ8851 SPI Ethernet Controller

The **KSZ8851** by *Microchip Technology* is a stand-alone **10/100BASE-TX Ethernet controller** with an SPI interface, designed to provide Ethernet connectivity to systems without a native MAC.

It integrates a complete Ethernet MAC and PHY, enabling direct connection to standard Ethernet networks via an external transformer and RJ45 connector.

Communication with the host system is performed via a high-speed **SPI interface**, allowing integration with platforms such as the Raspberry Pi without requiring a native Ethernet interface.

The device supports standard Ethernet features including **auto-negotiation**, **full- and half-duplex operation**, and hardware-assisted frame handling. Internal buffering and DMA-like mechanisms reduce CPU load and enable efficient data transfer over the SPI bus.

In Linux-based systems, the KSZ8851 is supported by the `ks8851` driver and integrates into the standard **network stack**, allowing the interface to be used like a conventional Ethernet device.

The controller is optimized for embedded and industrial applications requiring robust and flexible Ethernet connectivity over SPI. For further details, refer to the supplier's product documentation (see 1).

3.4.2 HX1188 Ethernet Transformer

The **HX1188** by *Pulse Electronics* is an integrated **Ethernet transformer module** (magnetics) designed for **10/100BASE-TX** applications.

It provides **galvanic isolation** between the Ethernet cable interface and the system electronics, ensuring robust operation in environments with ground potential differences and electrical noise.

The device integrates the required transformer structures for signal coupling as well as impedance matching, enabling direct connection between the Ethernet PHY and the RJ45 connector.

In addition to isolation, the magnetics support proper signal integrity by matching the differential impedance of the Ethernet lines and suppressing common-mode disturbances.

The HX1188 is optimized for compact embedded designs and industrial applications requiring reliable Ethernet connectivity and standardized isolation behavior. The transformer provides galvanic isolation of the Ethernet interface. For details on system-level isolation, see section 3.5. For further details, refer to the supplier's product documentation (see **??**).

3.4.3 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^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 4.

3.5 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
TR5001	Pulse	HX1188NLT	LAN-Transformer, see 3.4.2
C5023	PSK	FK21X102K502EGG	X1/Y2 Capacitor

Table 1: Isolation Components

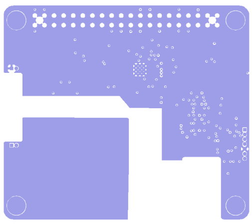


Figure 3: Isolation Barrier

3.6 Connectors

A shielded RJ45 connector is used for the Ethernet interface. The connector is pre-assembled and soldered on the board.

3.7 Jumper and Configuration

The board does not provide any jumpers or user-accessible configuration options. All functionality is defined by the hardware design and controlled via software.

3.8 Indicators (LEDs)

The board provides visual status indication for both the SPI interface and the Ethernet link. These indicators support quick diagnostics during operation, installation, and troubleshooting.

- **CS-ACT**: Bus activity of SPI interface
  - *on*: SPI transaction ongoing
  - *off*: SPI bus idle
- **LED1 - Speed**: Link speed indication of Ethernet interface
  - *on*: 100BASE-TX
  - *off*: 10BASE-T
- **LED0 - Link**: Link and activity indication of Ethernet interface
  - *blinking*: Data activity
  - *on*: Link established
  - *off*: No link

The Ethernet-related LEDs are directly driven by the PHY and reflect the actual link status independently of software configuration.

3.9 Signal Polarity and Wiring Orientation

The Ethernet interface uses a standard RJ45 connector with integrated auto-negotiation and automatic MDI/MDI-X crossover detection.

The PHY automatically adapts to the cable wiring, allowing the use of both straight-through and crossover Ethernet cables without user intervention.

Any standard CAT5 (or higher) twisted-pair Ethernet cable can be used.

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^2C 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

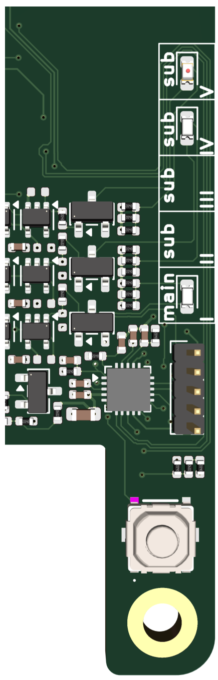


Figure 4: Instance ID indication on the PCB

Instance ID HW resources:

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

Instance Mode	Target Address	IRQ
I	0x9A	GP6
IV	0x98	GP14
V	0x92	GP25

Table 2: Exclusive HW resources

Instance Mode	Signal	Pin
I & IV & V & SCL1	GP3	
I & IV & V & SDA1	GP2	
I & II & III & IV & V	RESET	GP13
I & II & III & IV & V	SCL0	GP1
I & II & III & IV & V	SDA0	GP0

Table 3: Shared HW resources

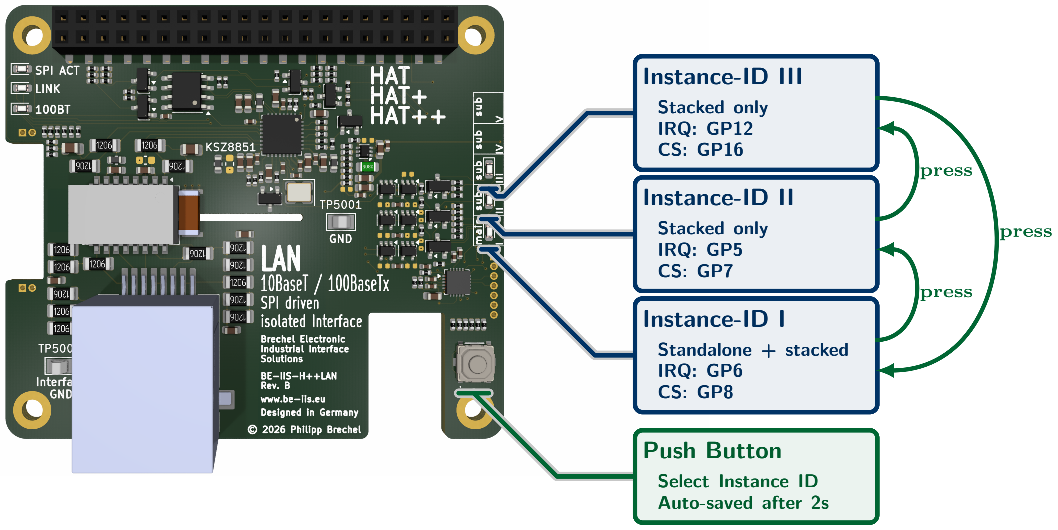


Figure 5: HAT Instance ID HW resources

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 **I²C 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-LAN-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 **I²C 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 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**
  - 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

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]	3.1	3.30	3.5
5 V Input [V]	4.5	5	5.5

Table 4: Voltage supply

6.2 Current Consumption

Parameter	Typ	Unit
Current @ 5 V	85	mA
Current @ 3.3 V	15	mA

Table 5: 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 6: Operating conditions

7.2 Usage

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

Table 7: 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-LAN
Condition	Assembly kit
Status	Partially assembled
Included Items	1× BE-IIS-HPP-LAN-PCBA-FT 4× M2.5x16 mm spacers 1× 2×20 pin stackable header
REACH & RoHS	Compliant with EU Directive 2011/65/EU and REACH Regulation (EC) No 1907/2006

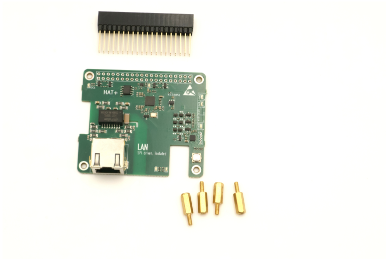


Figure 6: Delivery condition

9 Mechanical

9.1 Board Format

- Form factor: Raspberry Pi HAT+
- Mechanical dimensions: Raspberry Pi HAT compatible [3]
- Mounting hole pattern: Raspberry Pi HAT compatible [3]
- Stackszize: 15mm

9.2 3D Data

- Available on the BE-IIS product page [4]

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.

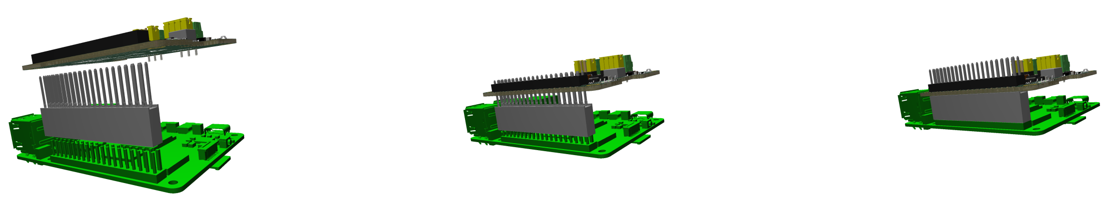


Figure 7: Assembly of the 2x20-pin stackable header between Raspberry Pi and HAT

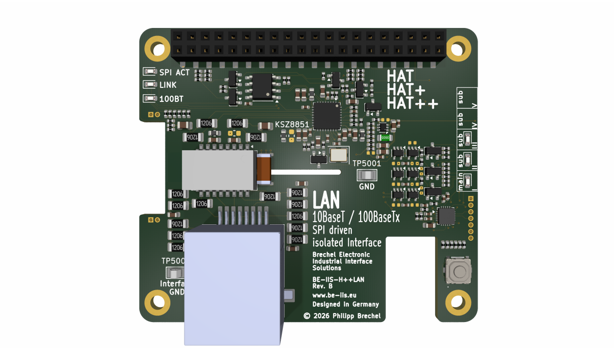
10.2 Assemble Spacer

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

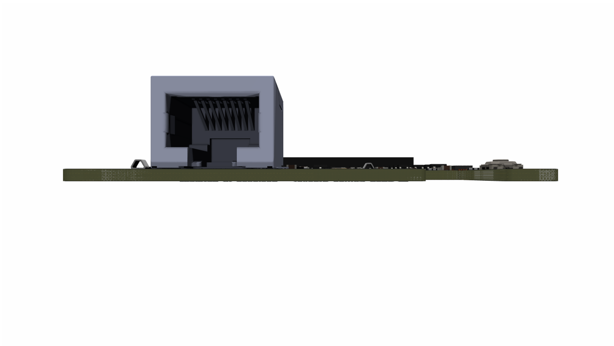
- Recommended spacer height: see Section **??**
- Fix the PCB using appropriate screws and spacers
- Ensure stable mechanical mounting to avoid stress on the connector

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

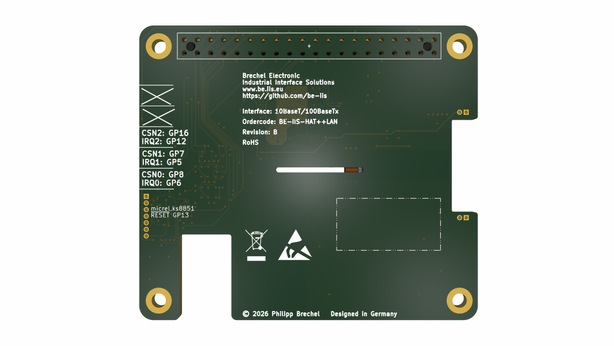
10.3 Board Overview



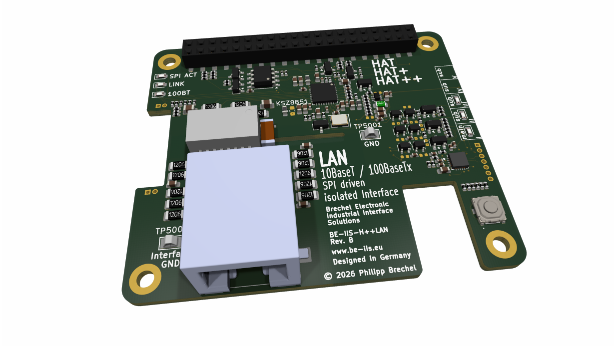
Top view



Front view



Bottom view



Side view

Figure 8: BE-IIS-HPP-LAN – mechanical overview

11 References

1. KSZ8851 Product Webside
2. BE-IIS Installer (Software and Setup Tools)
3. Raspberry Pi HAT+ Specification
4. Schematic, PCB-Viewer, BOM, 3D-Model

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)