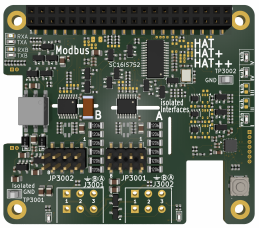
Isolated Dual RS-485 (Modbus) Industrial HAT++ for Raspberry Pi

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



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

Modbus over RS-485 enables robust long-distance multi-drop communication and is a well-established, future-proof industrial communication standard.

This HAT provides two fully independent, galvanically isolated RS-485 interfaces with automatic direction detection 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

- Dual Modbus (RS-485) HAT
- SC16IS752 dual I2C to UART controller
- up to 1.15MBit/s
- Galvanically isolated Interfaces
- Configurable 120 Ω bus termination
- RSP HAT+ compliant (2024)
- Stackable HAT (BE-IIS-HAT++)
- Configurable addressing and IRQ routing
- RoHS compliant
- Quality component suppliers

Product Description

The MODBUS Industrial HAT is a Raspberry Pi HAT+ compliant interface board supporting Modbus RTU/ASCII over RS-485 in accordance with TIA/EIA-485-A.
It integrates an SC16IS752 dual UART controller and an isolated ISO1500 RS-485 transceiver, providing galvanic isolation between the logic domain and the field bus interface.

Applications

- MODBUS network evaluation
- Embedded system prototyping
- Education and training
- Gateway and protocol converter development
- Industrial communication testing

Contents

1 Introduction

The BE-IIS HAT++ Modbus Industrial HAT is a Raspberry Pi HAT+ compliant interface board providing reliable Modbus RTU communication over RS-485 for industrial and laboratory environments.

The board integrates an SC16IS752 [1] dual UART controller for communication with the RS-485 interfaces. The UART channels are externally accessible and independently configurable.

The board integrates two fully independent RS-485 interfaces with galvanic isolation, ensuring electrical separation from the host system and between the channels.

The board integrates two fully independent RS-485 interfaces with galvanic isolation, ensuring electrical separation from the host system and between the channels. Each interface features automatic direction detection, eliminating the need for manual DE/RE control via software.

Modbus over RS-485 enables robust long-distance multi-drop communication and represents a mature and widely adopted industrial standard, well suited for reliable operation in electrically noisy environments.

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)**, **CAN/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 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 was designed for evaluation, prototyping, development, 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-MODBUS_B/

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

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

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

STL **3D Model (STEP/STL)**
https://www.be-iis.eu/products/BE-IIS-HPP-MODBUS_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++MODBUS-I2C enables Modbus RTU communication over RS-485 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 Modbus node.

Communication between the Raspberry Pi and the onboard RS-485 interfaces is implemented via an I²C-connected UART bridge, providing two independent serial channels. The RS-485 transceivers feature automatic direction detection, eliminating the need for manual DE/RE control.

3.2 Main Features

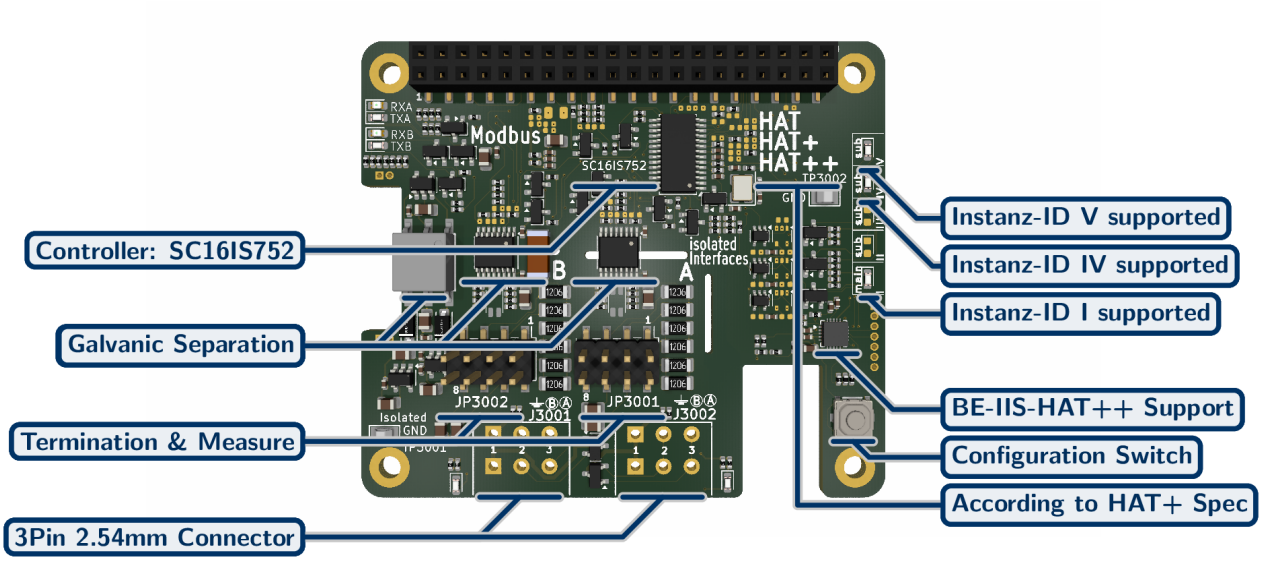


Figure 1: BE-IIS-MODBUS-I2C top view with annotations

Device

- Supplier: NXP
- SC16IS752I dual UART controller
- Supplier: Texas Instruments
- ISO1500 isolated RS-485 transceiver

I2C Interface

- 0x9A, 0x98, and 0x92 Device Address supported
- Persistent selection via push button

Isolation

- Galvanically seperated Interface
- Isolator rated up to 3KV

Protocol Support

- Supports Modbus RTU and ASCII protocols
- Half-duplex RS-485 interface
- Compliant with TIA/EIA-485-A standard

Direction Control (DIR)

- Automatic RS-485 direction control (DE/RE)
- Modbus RTU: configurable 15 ms turnaround timeout

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.

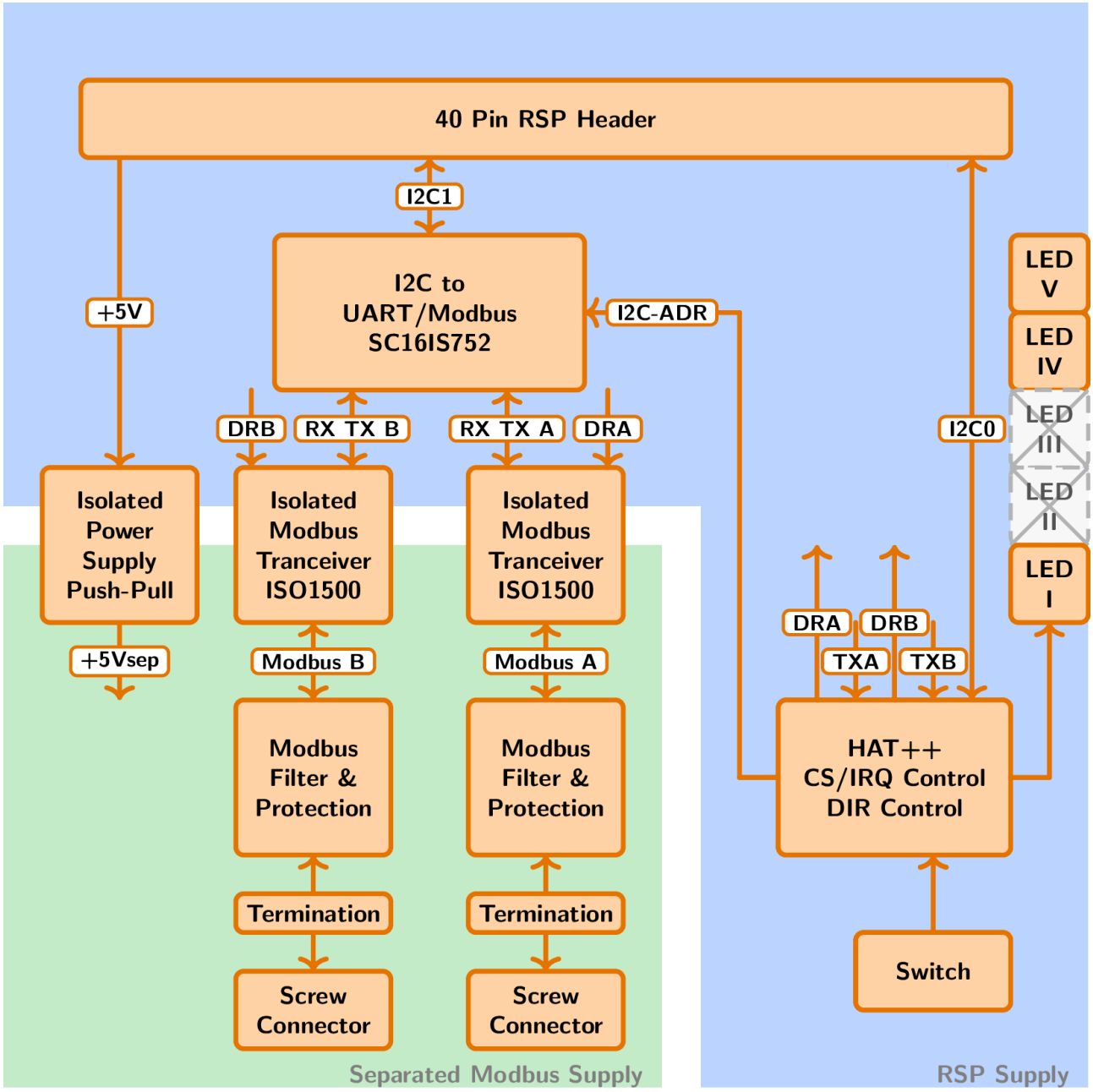


Figure 2: Simplified block diagram

3.4 Hardware Components

The hardware architecture and component interconnection are designed to realize a robust and fully **galvanically isolated Modbus RTU interface**. The system is optimized for industrial environments and fully aligned with the **HAT++ ecosystem**, enabling flexible and conflict-free integration in stacked configurations.

3.4.1 SC16IS752 Serial to UART Converter

The **SC16IS752** is a dual-channel UART bridge providing two independent UART interfaces via an I^2C interface.

Key characteristics:

- Dual UART channels
- I^2C interface to Raspberry Pi
- Integrated FIFOs for reliable data buffering

Oscillator:

- External oscillator defines baud rate generation
- Frequency: 22.1184 MHz

For further details, refer to the supplier's product documentation (see 1).

3.4.2 ISO1500 Digital Modbus Isolator

The **ISO1500** provides galvanic isolation for the RS-485 interface.

Functionality:

- Signal isolation between Raspberry Pi and field side
- Protection against ground potential differences
- Improved EMC robustness

For details on system-level isolation, see section 3.5. For further details, refer to the supplier's product documentation (see 2).

3.4.3 Isolated Power Supply

The board integrates an isolated power supply based on a push-pull converter topology.

Characteristics:

- Input: 5 V (Raspberry Pi side)
- Output: 5 V (isolated field side)
- Galvanic isolation between logic and field domain

For details on system-level isolation, see section 3.5.

3.4.4 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
- MODBUS direction control (automatic TX/RX switching)

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
- MODBUS direction control (automatic TX/RX switching)

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
TR3001	Bourns	PAD002-T764113S	Transformer, see 3.4.3
C3017	PSK	FK21X102K502EGG	X1/Y2 Capacitor
U3003,U3004	TI	ISO1500DBQR	Modbus TxRx, see see 3.4.

Table 1: Isolation Components

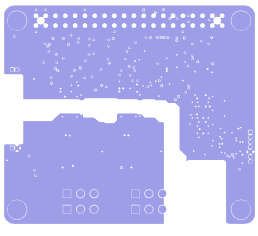


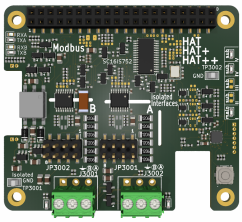
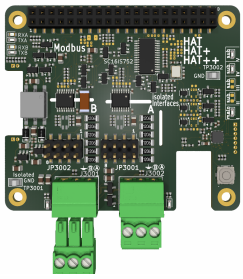
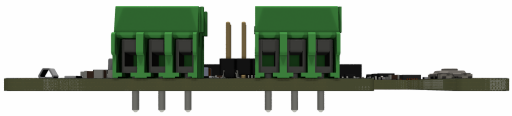
Figure 3: Isolation Barrier

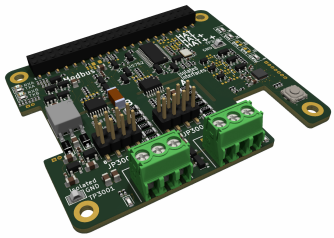
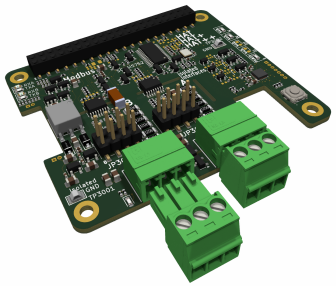
3.6 Connectors

Connectors J3001 and J3002 provide the physical RS-485 bus interfaces.

- **J3001**: Interface B
- **J3002**: Interface A

Each connector supports two possible mounting positions: a front position and a rear position. This allows flexible integration depending on the enclosure or mechanical constraints of the target system.

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 [4]

Option2:
Wuerth Elektronik WR-TBL 300VAC 10A 3P
Order Code: 691214110003 [5]

Note: or equivalent component with compatible mechanical and electrical specifications
Pinning: 1: SHIELD (optional, cable shield) 2: MODBUS-A (D+) 3: MODBUS-B (D-)

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 [6]
PCB Connector: Phoenix Contact - MC 1,5/ 3-ST-3,5 [7]
Order Code: 1844223(Header), 1840379(Connector)

Option2:
PCB Header: Wurth Electronic - WR-TBL 300VAC 12A 3P [8]
PCB Connector: Wurth Elecztronic - WR-TBL 300VAC 10.5Am Vertical 26-16AWG [9]
Order Code: 691305140003(Header), 691361100003 (Connector)

Pinning: 1: SHIELD (optional, cable shield) 2: MODBUS-A (D+) 3: MODBUS-B (D-)
Note: This connector is included in the delivery

3.7 Jumper Configuration JP3001 / JP3002

Jumpers **JP3001** and **JP3002** configure the RS-485 bus termination and optional external biasing networks for Interface 1 and Interface 2, respectively.
For proper Modbus operation, the 120 Ω termination resistor must only be enabled at the two physical end nodes of each RS-485 bus segment.

Bus topology requirements

- **End nodes:** 120 Ω termination enabled
- **Intermediate nodes:** No termination

The RS-485 transceiver (ISO1500DBQR) provides integrated fail-safe biasing to ensure a defined idle bus state.
In addition, optional external biasing networks are available on the board:

- Line A pulled up to +5 V via 5 kΩ
- Line B pulled down to GND via 5 kΩ

External biasing is typically not required due to the integrated fail-safe functionality of the transceiver. However, it may be enabled if required by specific system configurations or network conditions.
If used, external biasing should be enabled at only one node per RS-485 segment.

Point-to-point operation
In point-to-point configurations, both connected devices are considered end nodes. Therefore, termination must be enabled on both sides.
Figures 4 illustrate valid end-node configurations:

- Right: Termination enabled, bias disabled
- Left: Termination and external bias enabled

If additional nodes are present between the endpoints, these intermediate nodes must remain unterminated and without external biasing, as shown in Figure 5 (left).
Normally, external biasing is enabled at one end node only. The configuration shown in Figure 5 (right) is electrically valid but uncommon.

Pin	Signal	Pin	Signal
1	5 kΩ Pull-up (A to +5 V)	2	A
3	120 Ω (A–B termination)	4	A
5	120 Ω (A–B termination)	6	B
7	5 kΩ Pull-down (B to GND)	8	B

Table 2: JP3001 / JP3002 pin assignment for termination control and biasing.

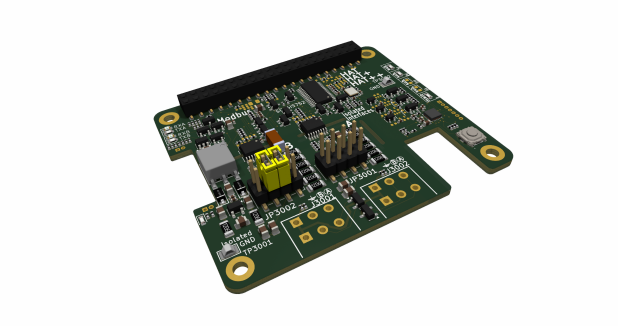


Figure 4: JP3001 / JP3002 RS-485 end-node configuration. Left: 120 Ω termination enabled, bias disabled.
Right: No jumpers installed.

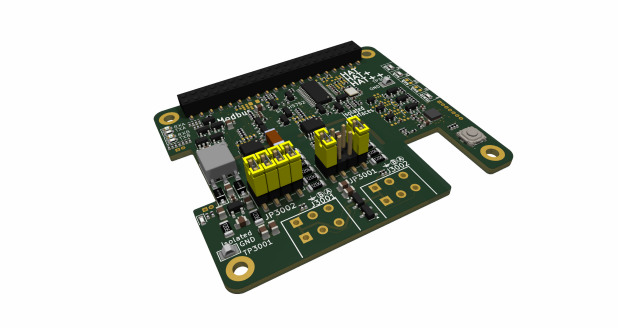


Figure 5: JP3001 / JP3002 intermediate-node configuration. LEFT: 120 Ω termination and 5 kΩ bias enabled. Right: Bias enabled only (termination disabled) – uncommon but valid.

Configuration summary

- Enable termination without biasing at one bus endpoint (Figure 4, right
- Intermediate nodes must remain unterminated and without biasing (Figure 5, right)
- If needed, enable termination with biasing at the opposite endpoint (Figure 4, left)

3.8 Indicators (LEDs)

In addition to the LED status bar on the side, which indicates the selected instance mode, additional status LEDs are located on the opposite side of the HAT:

- **Interface 1 RX/TX**: Bus activity of RS-485 Interface Bus A
  - *Green*: Receive activity (RX)
  - *Red*: Transmit activity (TX)
- **Interface 2 RX/TX**: Bus activity of RS-485 Interface Bus B
  - *Green*: Receive activity (RX)
  - *Red*: Transmit activity (TX)

3.9 Signal Polarity and Wiring Orientation

The RS-485 interface uses differential signaling with lines A and B.

The pinning of both connectors is identical and follows the PCB marking:

- **Pin 0**: Isolated GND
- **Pin 1**: Line B (inverting)
- **Pin 2**: Line A (non-inverting)

Correct polarity shall be observed when connecting the bus. Line A must be connected to line A and line B to line B across all nodes.

Both interfaces share a common isolated ground reference. This ground is galvanically separated from the Raspberry Pi (host) ground.

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

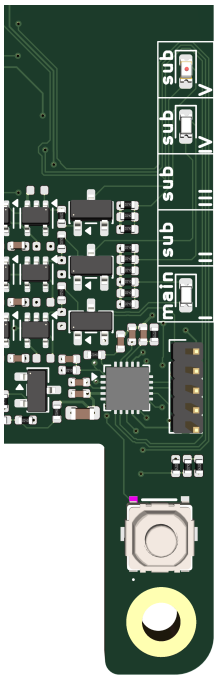


Figure 6: Instance ID indication on the PCB

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 3: 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 4: Shared HW resources

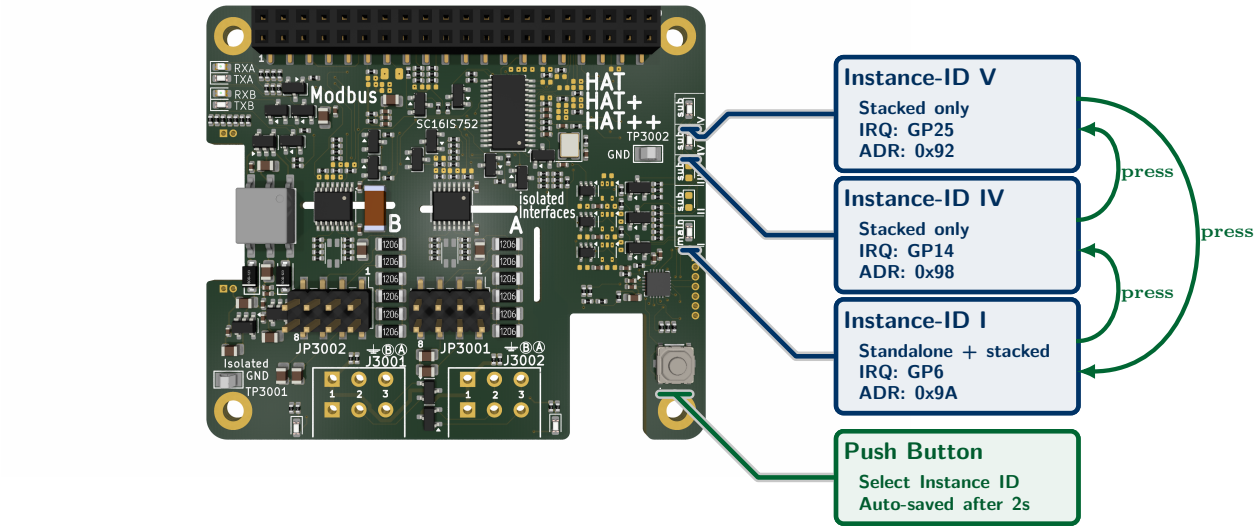


Figure 7: HAT Instance ID HW resources

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-MODBUS-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 [3]	**Software:** • Run the BE-IIS installer from GitHub [3] • 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]	3.0	3.30	3.6
5 V Input [V]	4.5	5	5.5

Table 5: Voltage supply

6.2 Current Consumption

Parameter	Typ	Unit
Current @ 5 V	200	mA
Current @ 3.3 V	15	mA

Table 6: 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 7: Operating conditions

7.2 Usage

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

Table 8: 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-MODBUS
Condition	Assembly kit
Status	Partially assembled
Included Items	1× BE-IIS-HPP-MODBUS-PCBA-FT 4× M2.5x16 mm spacers 1× 2×20 pin stackable header 8x Jumper 2x PCB Header (see 3.6) 2x PCB Connector (see 3.6)
REACH & RoHS	Compliant with EU Directive 2011/65/EU and REACH Regulation (EC) No 1907/2006

Figure 8: Delivery condition

9 Mechanical

9.1 Board Format

- Form factor: Raspberry Pi HAT+
- Mechanical dimensions: Raspberry Pi HAT compatible [10]
- Mounting hole pattern: Raspberry Pi HAT compatible [10]
- Stackszize: 16mm

9.2 3D Data

- Available on the BE-IIS product page [11]

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.

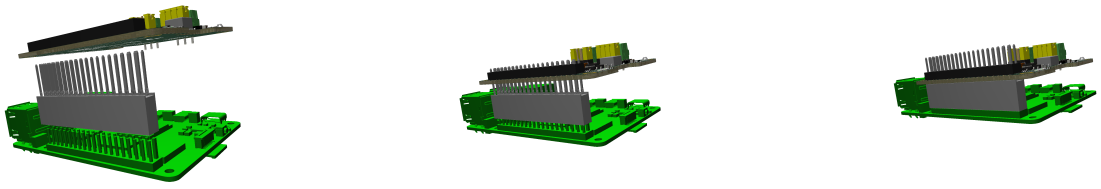


Figure 9: Assembly of the 2x20-pin stackable header between Raspberry Pi and HAT

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

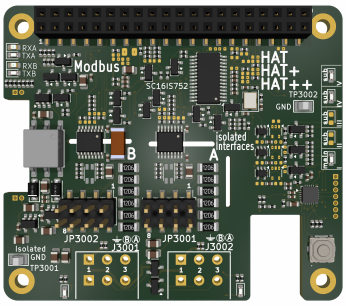
Important notes:

- 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

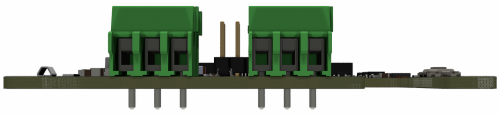
Correct assembly ensures reliable electrical contact and proper usability of the terminal interface.

10.4 Board Overview

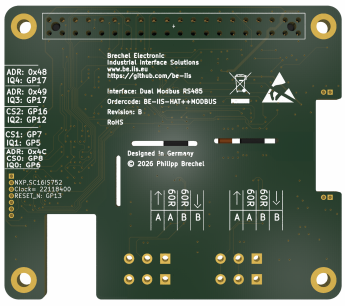
11 References



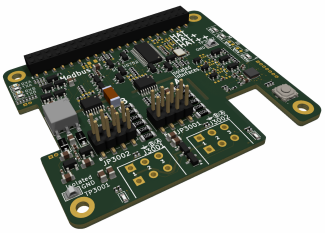
Top view



Front view



Bottom view



Side view

Figure 10: BE-IIS-HPP-MODBUS – mechanical overview

1. SC16IS752IPW Product Webside
2. ISO1500 Product Webside
3. BE-IIS Installer (Software and Setup Tools)
4. PhoenixContact PT 1,5/ 3-3,5-H - PCB terminal block
5. Wurth Electronic - WR-TBL Series 2141 - 3.50 mm Horiz. Entry Modular
6. PhoenixContact - MC 1,5/ 3-G-3,5 - PCB header
7. PhoenixContact - MC 1,5/ 3-ST-3,5 - PCB connector
8. Wurth Electronic - WR-TBL 300VAC 12A 3P - PCB header
9. Wurth Electronic - WR-TBL 300VAC 10.5Am Vertical 26-16AWG
10. Raspberry Pi HAT+ Specification
11. 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`