BE-IIS Quick Reference

Installer, HAT+/HAT++, Journal and Repository Structure

For Raspberry Pi OS: Git-based installation, automatic HAT++ detection after boot and diagnostics with standard Linux tools.

1. Installer and Repository

Full Installation

`sudo apt install -y git`	Install Git
`git clone` https://github.com/be-iis/be-iis-installer.git	Clone repository:
`cd be-iis-installer`	Enter repository
`sudo ./scripts/install/install-all.sh`	Install all components
`sudo reboot`	Activate overlays, modules and service

What the Installer Configures

`01_install_eepdump.sh`	Read HAT EEPROMs
`02_install-mods.sh`	Required kernel modules
`03_install-overlays.sh`	Device-tree overlays
`04_install_hatpp_service.sh`	HAT++ system service
`05_install-udev-rules.sh`	Stable device names
`06_I2C_VC_ena.sh`	Enable HAT-ID I^2C

GitHub Structure

`common/`	Shared shell functions
`docs/`	Guides and technical docs
`examples/`	Example configurations
`platform/`	OS integration, systemd, udev
`products/`	Per-board overlays, scripts and tests
`scripts/`	Install, network and SSH entry points
`tools/`	EEPROM, kernel and overlay helpers

Typical Product Directory

`overlays/src/rpi/`	DTS sources and Makefile
`overlays/build/rpi/`	Compiled DTBO files
`scripts/`	Board setup and configuration
`systemd/ / udev/`	Optional integration
`test/`	Production and functional tests

2. HAT+ and HAT++

Instances and ID Addresses

I	`0x50`	Standard HAT+; detected during boot
II	`0x60`	Loaded by service after boot
III	`0x70`	Loaded by service after boot
IV	`0x74`	Loaded by service after boot
V	`0x76`	Loaded by service after boot

Each stack needs exactly one Instance I. Additional boards use different Instances II to V. Select with the button and verify the LEDs.

Automatic Boot Flow

1. Raspberry Pi firmware detects Instance I as a normal HAT+.
2. `be-iis-hatpp.service` scans the additional ID addresses.
3. `eepdump` reads the overlay name from the HAT data.
4. `dtoverlay` loads the matching instance variant.
5. Kernel modules and udev names become available.

Verify Detection

`i2cdetect -y 0`	Look for ID addresses 50/60/70/74/76
`ls /sys/bus/i2c/devices/0-00{50,60,70,74,76}`	Detected HAT-ID devices
`dtoverlay -l`	List runtime overlays
`lsmod | grep -E 'lan865|adin|mcp251|sc16'`	Check common driver modules
`ls -l /dev/beiis-*`	Stable UART/-Modbus device names
`ip -br link`	Network and CAN interfaces

Important HAT++ Rules

1. Instances must not be duplicated in one stack.
2. Unused `spidev` nodes must remain free so additional chip selects can be loaded later.
3. Reboot after installer, overlay or boot-configuration changes.
4. Keep product files below `products/<board>/`; do not duplicate shared platform files.

3. systemd, Journal and Troubleshooting

BE-IIS Service and Journal

`systemctl status be-iis-hatpp.service`	Service state and recent logs
`systemctl is-enabled be-iis-hatpp.service`	Check boot enablement
`sudo journalctl -u be-iis-hatpp.service -b`	Service logs for this boot
`sudo journalctl -b | grep 'BE-IIS'`	Complete HAT++ detection output
`sudo journalctl -u be-iis-hatpp.service -f`	Follow logs live
`sudo systemctl restart be-iis-hatpp.service`	Run detection again

Control Services – Examples

`systemctl status ssh.service`	Show status
`sudo systemctl start ssh.service`	Start now
`sudo systemctl stop ssh.service`	Stop now
`sudo systemctl restart NetworkManager.service`	Restart
`sudo systemctl enable ssh.service`	Start at boot
`sudo systemctl disable ssh.service`	Remove boot startup
`sudo systemctl enable -now my-app.service`	Enable and start immediately
`systemctl -failed`	Failed units

Unit Files, Targets and Timers

`systemctl list-units -type=service`	Loaded service units
`systemctl list-unit-files -type=service`	Installed unit files
`systemctl cat my-app.service`	Show effective unit
`sudoedit /etc/systemd/system/my-app.service`	Edit a custom unit file
`sudo systemctl daemon-reload`	Reload changed unit files
`systemctl get-default`	Default boot target
`systemctl list-units -type=target`	Loaded targets
`systemctl list-timers -all`	Timers and next activation

Typical Troubleshooting

1. `systemctl status be-iis-hatpp.service`
2. `journalctl -xeu be-iis-hatpp.service`
3. `systemctl cat be-iis-hatpp.service`
4. `dtoverlay -l` and `i2cdetect -y 0`
5. `dmesg | grep -Ei 'BE-IIS|lan865|adin|mcp251|sc16'`
6. Check instance LEDs, wiring and reboot.

Important: Keep These Four Things Separate

BE-IIS
BRECHEL ELECTRONIC
Industrial Interface Solutions

`start` / `stop`

Changes only the current runtime state.

Instance I

Normal HAT+ detection by Raspberry Pi firmware.

`enable` / `disable`

Changes automatic startup during boot.

Instances II–V

Detected and loaded after boot by `be-iis-hatpp.service`.