



INDUSTRIAL SHIELDS

Gateberry



Gateberry:

- Raspberry Pi 4
- UPSberry (UPS Smart Shield and Real Time Clock (RTC) for Raspberry Pi)
- Enclosure

Industrial Shields

Gateberry User Guide

Revised March 2025

Preface

This User Guide has been implemented by Boot & Work, S.L. working under the name Industrial Shields.

Purpose of the manual

The information contained in this manual can be used as a reference to operate, to functions, and to the technical data of the signal modules, power supply modules and interface modules.

Intended Audience

This User Guide is intended for the following audience:

- Persons in charge of introducing automation devices.
- Persons who design automation systems.
- Persons who install or connect automation devices.
- Persons who manage working automation installation

Intended use of Industrial Shields products

Consider the following:

Industrial Shields products should only be used for the cases of application foreseen in the catalogue and the associated technical documentation. If third-party products and components are used, they must have been recommended or approved by Industrial Shields.

The correct and safe operation of the products requires that your transport, storage, installation, assembly, operation and maintenance have been carried out in a correct manner. It must respect the permissible ambient conditions. You should also follow the indications and warnings that appear in the associated documentation.

The product / system dealt with in this documentation should only be handled or manipulated by qualified personnel for the task entrusted and observing what is indicated in the documentation corresponding to it, particularly the safety instructions and warnings included in it. Due to their training and experience, qualified personnel are in a position to recognize risks resulting from the handling or manipulation of such products / systems and to avoid possible hazards.

Disclaimers

Weights and Dimensions

Dimensions and weights are nominal and they are not used for manufacturing purposes, even when tolerances are shown.

Performance Data

The performance data given in this manual is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of INDUSTRIAL SHIELDS's test conditions, and the users must correlate it to actual application requirements. Actual performance is subject to the INDUSTRIAL SHIELDS Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons.

It is our practice to change model numbers when features are changed, or published ratings or when significant construction changes are made. However, some specifications of the products may be changed without any notice. When in doubt, special numbers may be assigned to fix or establish key specifications for your application on your request. Please consult with your INDUSTRIAL SHIELDS representative at any time to confirm actual specifications of purchased products

Errors and Omissions

The information in this document has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

These components may only be operated in closed housings or in higher-level control cabinets with protective covers that are closed, and when all the protective devices are used. These components may only be handled by qualified and trained technical personnel who are knowledgeable and observe all the safety information and instructions on the components and in the associated technical user documentation. When carrying out a risk assessment of a machine in accordance with the EU Machinery Directive, the machine manufacturer must consider the following residual risks associated with the control and drive components of a PDS.

Table of Contents

1 General Description.....	8
1.1. General Specifications.....	8
1.1.1. Raspberry Pi 4.....	8
1.1.2. UPSberry.....	9
1.2. Available Communications & Accessories.....	9
1.3. Precautions.....	10
1.3.1. Raspberry Board.....	10
1.3.2. Intended Audience.....	10
1.3.3. General Precautions.....	10
1.3.4. Isolation Precautions.....	10
1.3.5. USB Port Precautions.....	11
1.4. Overall device pinout.....	12
1.4.1. UPSberry.....	13
1.5. Main parameters.....	13
1.6. Raspberry Pi 4 Pinout Connector.....	14
1.7. Unmount/Mount instructions.....	15
1.8. Connectors.....	17
2 Raspberry Pi Software.....	20
2.1 Standard image installation.....	20
2.2 Controlled Shutdown with UPS.....	20
2.3 Executing Commands Before Shutdown.....	21
2.4 RTC.....	22
2.5 RS-485.....	22
2.6 Expansion Board Slot.....	23
2.6.1 LoRa.....	23
2.6.1.1 Module Overview.....	23
2.6.1.2 Software implementation.....	24
2.6.2 4G / NBIoT / GPRS.....	24
2.6.2.1 Module Overview.....	24
2.6.2.2 Software implementation.....	25
2.6.3 4G LTE-E.....	26
2.6.3.1 Module Overview.....	26
2.6.3.2 Software implementation.....	26
2.6.4 CANBus.....	27
2.6.4.1 Module Overview.....	27
2.6.4.2 Software implementation.....	27

2.6.5 GPS.....	29
2.6.5.1 Module Overview.....	29
2.6.5.2 Software implementation.....	29
3 Revision Table.....	30



INDUSTRIAL SHIELDS

1 General Description

This device consists of a Raspberry Pi 4 attached to an UPSberry (UPS Smart Shield and Real Time Clock (RTC) for Raspberry Pi), all in a perfectly fitted enclosure.

1.1. General Specifications

1.1.1. Raspberry Pi 4

Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	2GB, 4GB or 8GB LPDDR4 (depending on model)
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)
Video & Sound	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES 3.1 graphics, Vulkan 1.0
SD card support	Micro SD card slot for loading operating system and data storage
Input Power	5Vdc via USB-C connector (minimum 3A*) 5Vdc via GPIO header (minimum 3A*) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Operating temperature 0–50°C

*A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total.

1.1.2. UPSberry

- Plug & Play UPS Smart Shield and Real Time Clock (RTC) for Raspberry Pi
- Avoid any SD corruption issue
- 12 Vdc or 24 Vdc
- With RS485 protocol available
- Compatible with all Raspberry Pi 4
- Recharge Time: Less than 1 minute
- Reconnecting time: Less than 20 seconds
- With reboot button (or the possibility to add an external one)
- Vin ESD protection

Operational Voltage	Current	Max. Current	Super Capacitors	Certificate	Possible MCU Connections	RTC
12 or 24 Vdc (Antipolarity + Filter)	3 A (Autoprotected by chip)	3.5 A	x2 (25F) 2.7V	CE, RoHs	-Raspberry Pi 4 models	DS3231 Coin cell: CR1220

Note: The power consumption requires a DC power supply ranging from a minimum of **20 W** to a maximum of **30 W**.

1.2. Available Communications & Accessories

- RS-485 Port
- ICSP Connector
- I2C (3.3 Vdc)
- USB Type-C (Power Only)
- RTC
- Raspberry Pi Connector
- Expansion Board Slot

1.3. Precautions

Read this manual before attempting to use the Gateberry and follow its descriptions for reference during operation.

1.3.1. Raspberry Board

The Gateberry uses a Raspberry Pi 4 Board as controller.

1.3.2. Intended Audience

This manual is intended for technicians, which must have knowledge on electrical systems.

1.3.3. General Precautions

The user must operate Gateberry according to the performance specifications described in this manual.

Before using the Gateberry under different conditions from what has been specified in this manual or integrating into nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your INDUSTRIAL SHIELDS representative. Ensure that the rating and performance characteristics of the Gateberry are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment double safety mechanisms. This manual provides information for programming and operating the Gateberry.

1.3.4. Isolation Precautions

Description:

This equipment does **not include galvanic isolation between the grounds** of the different systems. This means that if an external device or sensor that shares the same ground reference (GND) with the system is connected, any potential difference between these grounds could damage the connected components. To avoid issues with interference, ground loops, or damage to external equipment, ensure that all connected devices share the same ground reference or use systems with appropriate isolation.

Recommendations:

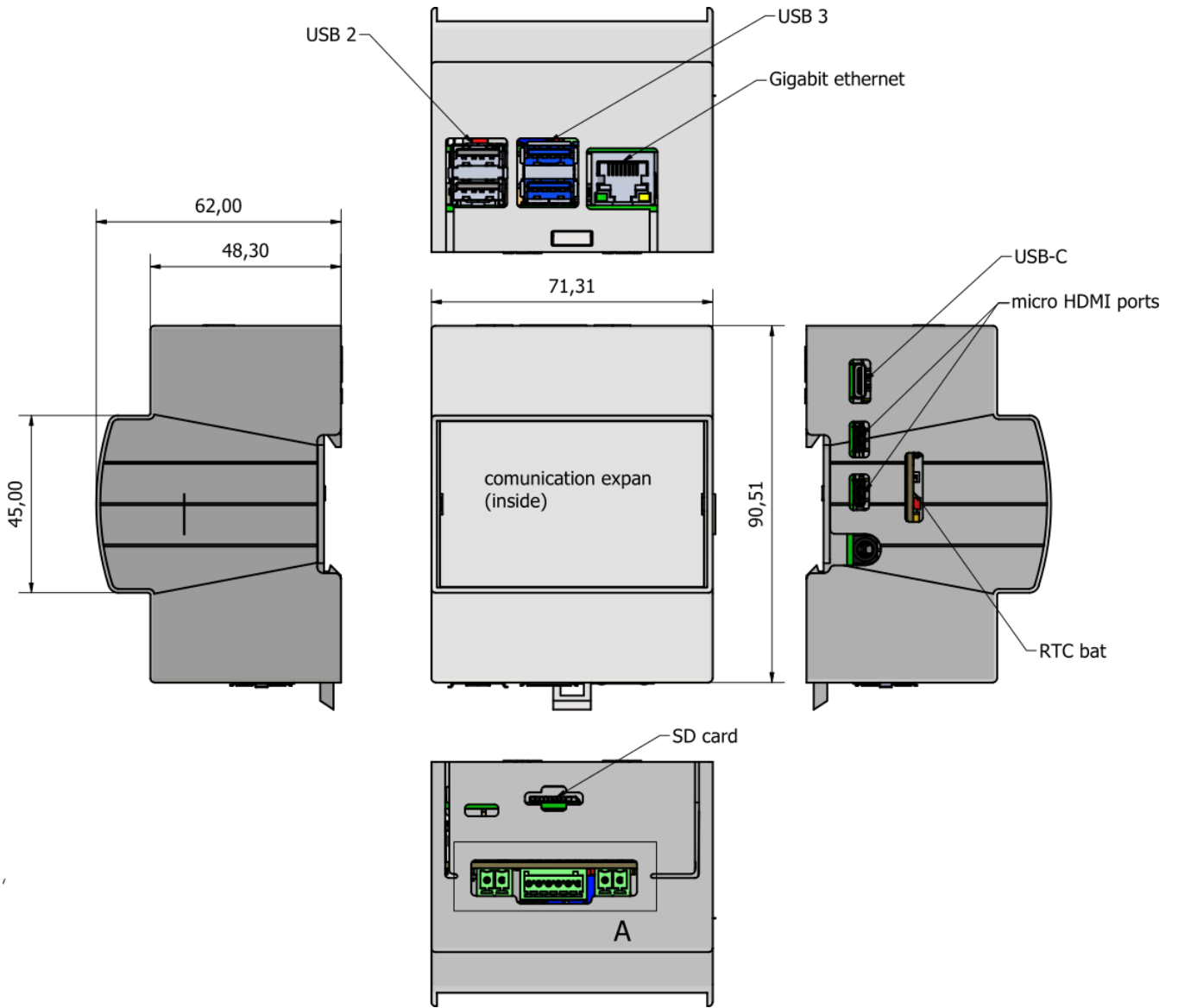
- **Connection Review:** Verify that all ground connections are properly made and that there are no significant potential differences between them.
- **Use of Isolation:** Consider using **galvanic isolators** or **isolation transformers** if it is necessary to connect equipment with different ground references.

1.3.5. USB Port Precautions

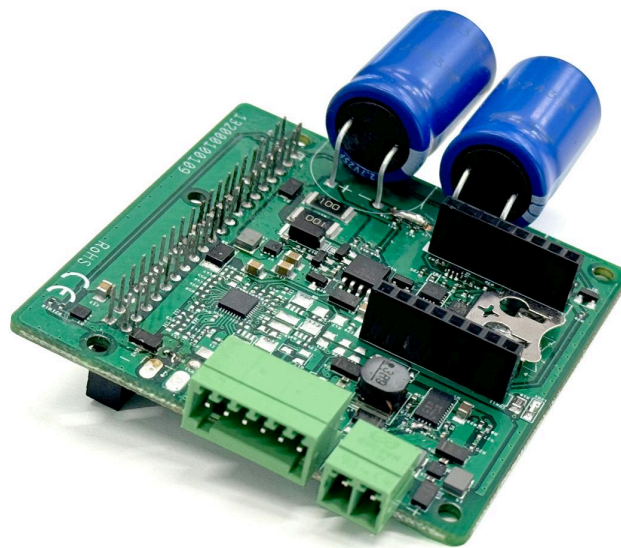
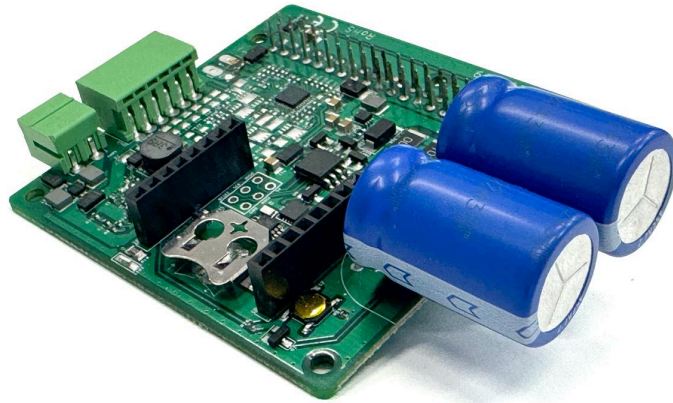
When connecting devices to the USB port, ensure their total power consumption does not exceed 500 mA. At this maximum limit, the USB voltage remains stable at 4.2V.

Regardless of USB power usage, the UPS board supplying power to the Raspberry Pi maintains a stable voltage of at least 4.9V. Even at a 500 mA load, the measured voltage remains at 4.93V, well above the 4.85V safety threshold.

1.4. Overall device pinout



1.4.1. UPSberry



1.5. Main parameters

Parameter	Value	Conditions
Input parameters		
Input voltage (screw terminal)	7Vdc ... 28Vdc	At screw terminal
Input voltage (USB terminal)	5Vdc	USB terminal
Average input current	0.5A	Charge mode + RPi Booting at Vin = 24V
Output parameters		
Output voltage range	4.75Vdc ... 5.5Vdc	
Output current range	0Aavg ... 2Aavg	
Output ripple	20mVpp	Raspberry Pi as a web server
Control parameters		
Pin 16 (GPIO23)	3.3Vdc – 5Vdc (Inputs max V)	
Pin 13 (GPIO27)	3.3Vdc – 5Vdc (Outputs max V)	

1.6. Raspberry Pi 4 Pinout Connector

	NC	1	2	Vin	
SDA	GPIO 2	3	4	Vin	
SCL	GPIO 3	5	6	GND	
TX	GPIO 4	7	8	GPIO 14	D
	GND	9	10	GPIO 15	R
TERM_TX	GPIO 17	11	12	NC	
DE/RE	GPIO 27	13	14	GND	
	NC	15	16	GPIO 23	To Attiny
	NC	17	18	GPIO 24	From Attiny
MOSI	GPIO 10	19	20	NC	
MISO	GPIO 9	21	22	GPIO 25	INT
SCLK	GPIO 11	23	24	GPIO 8	CS
	GND	25	26	NC	
	NC	27	28	NC	
RX	GPIO 5	29	30	GND	
	NC	31	32	NC	
	NC	33	34	GND	
	NC	35	36	GPIO 16	AN
RST	GPIO 26	37	38	GPIO 20	PWM
	GND	39	40	GPIO 21	INT

I2C	RS-485	UPS	RTC	EXP BOARD
-----	--------	-----	-----	-----------

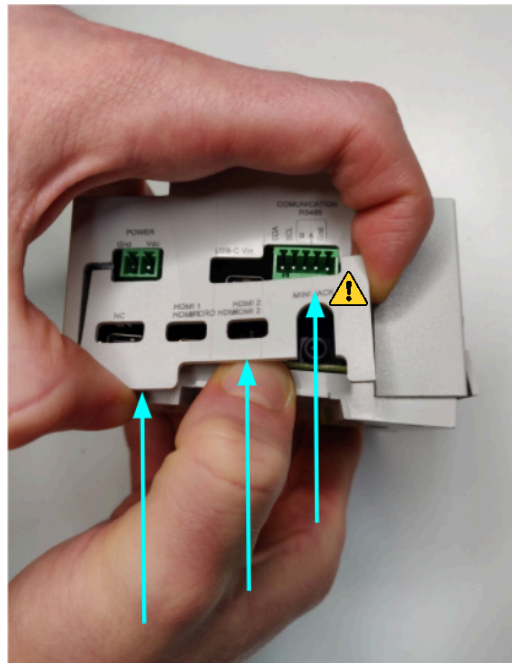
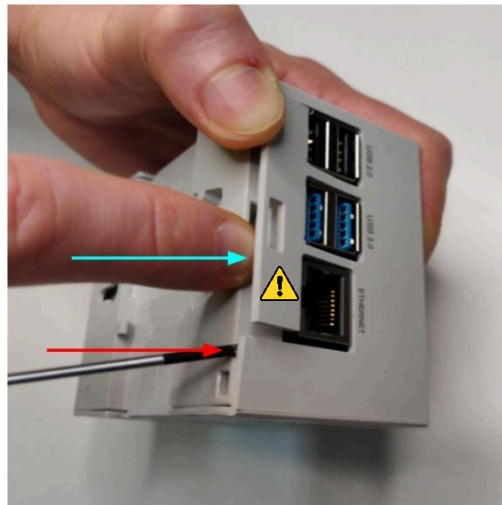
- GPIO24 (Raspberry's 18 pin) is used by UPS to report if an emergency power-off is needed. If the UPSberry detects that the external power supply has been lost, this pin is going to be connected to GND. In the opposite case, this pin is not going to be connected. Because of this, it is recommended to configure this pin with a pull-up software.
- GPIO23 (Raspberry's 16 pin) is used to report a finished saving process. If the system fails, the UPSberry will maintain the power supply till it receives a low logic value from this pin. If this pin is not connected, in the case of a failure the UPSberry will provide power till the capacitors run out of energy.
- GPIO27 (Raspberry's 13 pin) is the RS-485 half duplex control. The transmission is enabled with a positive logical value. The reception is enabled with a negative logical value.
- The reset white connector must have a normally open push button and, when it is pushed, the power supply is going to be removed from the Raspberry. This job can be equally done pushing the reset button in the shield.
- **CAUTION:** This shield is used to discharge the capacitors but, when they are fully charged, it is very important to ensure that no short-circuits will happen, as it may break the shield.

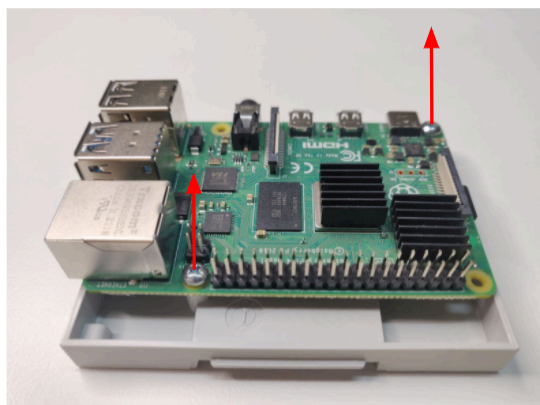
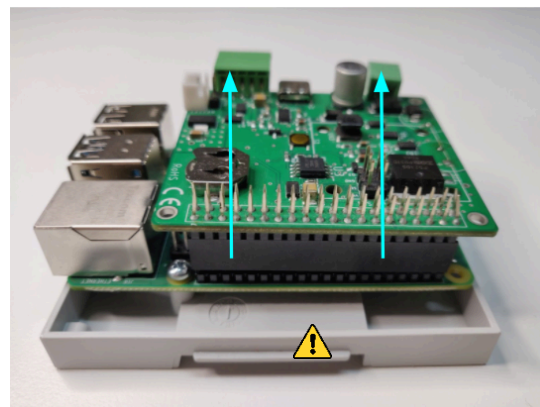
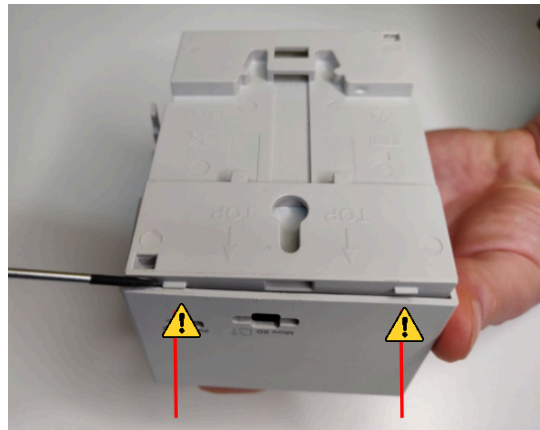
1.7. Unmount/Mount instructions

The device must be unmounted to place the Raspberry within the UPSberry (if required) or to access the cell coin slot.

The steps to unmount it are the following ones:

- Carefully, push with the fingers where the blue arrow is located and push with a screwdriver (or screw/unscrew if required) where the red arrow is located. Be delicate with the spots pointed with the alert sign. Remember to remove all the connectors and the microSD card to mount/unmount the device and place them again when the process is finished.



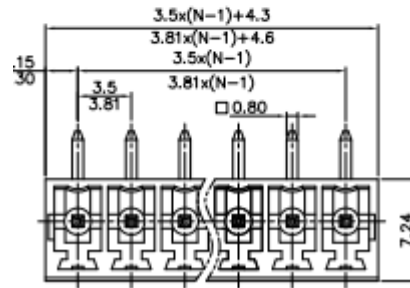
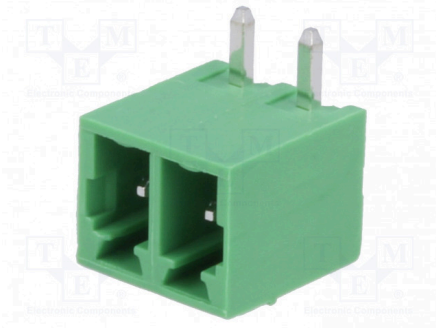


- The steps to mount it are the same but inversely (be careful with the plastic tabs and make sure that everything is placed the right way).

1.8. Connectors

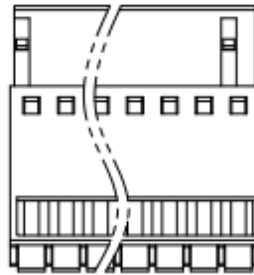
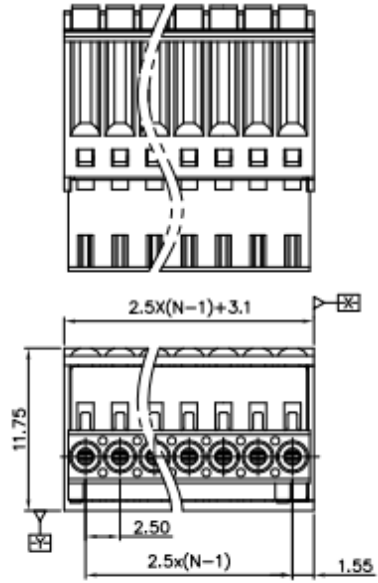
- Power connector:

SKU: 15EDGRC-3.81-02P-14-00A(H)



- Communications connector:

SKU: 15EDGKD-2.5-XXP-1Y-00A(H)

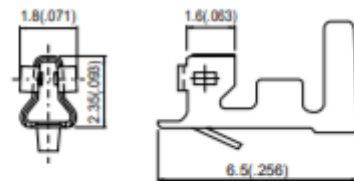


- Reset connector:

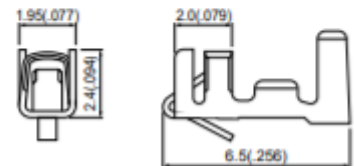
SKU: B2B-XH-A (LF)(SN)

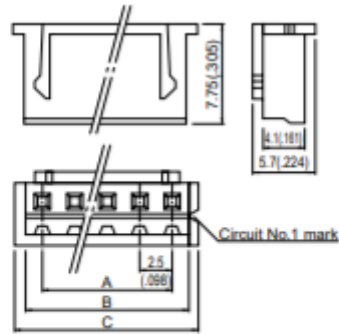


Shape A

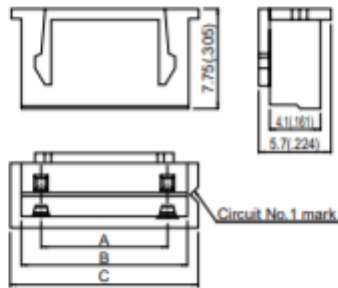


Shape B

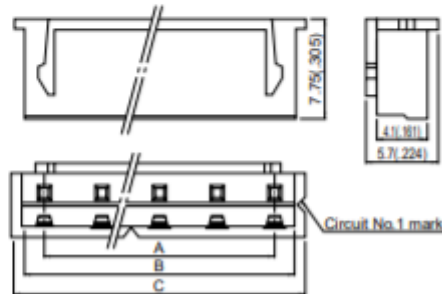




Plugged up
(2 circuits 10.0mm(.394") pitch)



(6 circuits 5.0mm(.197") pitch)



2 Raspberry Pi Software

To start working with the UPS on your Raspberry Pi module, the OS needs to be properly installed and configured. This section goes through all the necessary steps to configure it properly.

2.1 Standard image installation

Raspberry must be used with an SD card inserted in the available slot which has a correct image installed. By default, the SD card is not provided. You will need to acquire one and install an appropriate OS image. The most used and recommended one is Raspberry Pi OS, previously known as Raspbian.

However, in case you want to install another operating system compatible with Raspberry Pi 4B it can be done, but keep in mind that all the services, scripts and executables must be added to the new image to set up the board features (the additional communications, the internal UPS, etc.).

These are the steps to install a fresh Raspberry Pi OS image:

1. Download the Raspberry Pi Imager from this URL: <https://www.raspberrypi.org/software/>
2. Download an operating system image from Raspberry Pi. All the operating system images available here: <https://www.raspberrypi.org/software/operating-systems/>
3. Take your microSD card with an adapter for your PC, and write the image using the Raspberry Pi Imager.
4. Once the process ends, you can introduce the microSD card in your Raspberry PLC.

2.2 Controlled Shutdown with UPS

The Raspberry based PLCs can perform a controlled shutdown once the UPS detects that the main power supply has been cut. This ensures all active processes are ended and the SD card image and files are protected.

To enable the controlled shutdown on the Raspberry Pi, follow these steps:

1. **Modify the configuration file:** The process is now managed using device tree overlays. Add the following two lines to the `/boot/firmware/config.txt` file:

```
dtoverlay=gpio-poweroff,gpiopin=23,active_low
dtoverlay=gpio-shutdown,gpio_pin=24,gpio_pull=up
```

- The `gpio-poweroff` overlay ensures the system powers off gracefully using GPIO 23.
 - The `gpio-shutdown` overlay detects when the system should shut down using GPIO 24.
2. **Save and reboot:** After editing and saving the file, reboot the Raspberry Pi to apply the changes.

The system will now perform a controlled shutdown when the main power is lost, ensuring the protection of data and processes.

2.3 Executing Commands Before Shutdown

To execute specific commands before the system powers off, follow these steps:

1. **Create the necessary directory:** Create the required directory to store the pre-poweroff hooks:

```
sudo mkdir -p /etc/rpishutdown/hooks
```

2. **Download the service file and hook script:** Fetch the required files for the pre-poweroff service and the check script:

```
sudo curl -L
https://apps.industrialshields.com/main/rpi_experimental/rpiplc/rpishutd
own-pre-poweroff.service \
    -o /lib/systemd/system/rpishutdown-pre-poweroff.service
sudo curl -L
https://apps.industrialshields.com/main/rpi_experimental/rpiplc/check-pr
e-poweroff-hook.sh \
    -o /etc/rpishutdown/hooks/check-pre-poweroff
```

Any type of file (Bash Scripts, Python, C++, etc.) can be executed, but the most important thing is that the file must be called “pre-poweroff” without any extension (and you must indicate the file type with the corresponding Shebang). The script execution time cannot surpass ~6 seconds, as it is the Raspberry’s Power Off period.

3. **Grant execution permissions:** Make the pre-poweroff script executable:

```
sudo chmod ugo+x /etc/rpishutdown/hooks/check-pre-poweroff
```

4. **Enable and start the pre-poweroff service:** Reload the systemd daemon and enable the new service:

```
sudo systemctl daemon-reload
sudo systemctl enable rpishutdown-pre-poweroff
sudo systemctl start rpishutdown-pre-poweroff
```

This setup ensures that any custom commands in the check-pre-poweroff script will be executed before the system powers off.

2.4 RTC

To enable the RTC functionalities, the steps to follow are:

1. **Update your system:** Make sure your system has the latest updates by running:

```
sudo apt update
sudo apt -y upgrade
```

2. **Enable I2C on the Raspberry Pi:** Open the configuration file based on your system version:

- For Bullseye: `/boot/config.txt`
- For Bookworm or later: `/boot/firmware/config.txt`

Uncomment the following line if it's commented:

```
dtparam=i2c_arm=on
```

3. **Enable the RTC overlay:** Add the following line at the end of the file, after the `[all]` section:

```
dtoverlay=i2c-rtc,pcf85363
```

4. **Restart the device:** Save the changes and reboot the system to apply the configuration.

2.5 RS-485

The UPSberry module contains a 2-wire RS-485 (or RS-422) transceiver. The transceiver is driven by the Raspberry Pi UART interface on the GPIO14 and GPIO15 pins. We send and receive data by `/dev/ttyS0`.

- In order to ensure you have got the latest updates, you should run the following commands:

```
sudo apt update
sudo apt -y upgrade
```

- At the end of the `/boot/firmware/config.txt` file, after the `"[all]"` statement, introduce this line (Use `/boot/firmware/config.txt` for Bullseye based OS) :

```
enable_uart=1
```

The following table includes the RS-485 port requirements:

Parameter	Value	Conditions
RS-485 Interface Communication Type	2-wire, half duplex	Transceiver chip ISL8483E
RX/TX direction control	Controlled by GPIO24	Diver enabled by a low TX data bit Driver disabled 25 μ s after the end of a low TX data bit
Failsafe biasing	390R to GND 390R to 5V	
Line Termination	120R	390R//220R//390R+2*10R
Protection	ESD to +/- 15 kV	No Surge/Burst protection
Echo-feature	RX always enabled TX enabled at data	Optional
Data transmission speed	0...250kbps	25 μ s transmission pause between TX and RX required

2.6 Expansion Board Slot

The UPSberry module contains an expansion slot board, to be able to add new features such as 4G, NB, GPRS, LoRa, CANBus and more. These expansion boards can use up to 3 different communication protocols: UART, SPI or I2C.

2.6.1 LoRa

2.6.1.1 Module Overview

The LoRa (Long Range) expansion board is designed for low-power, long-range communication and operates on different frequency bands depending on the region. The module used is either the LR Click for Europe and Asia or the LR 2 Click for North America.

- Model: RN2483 (EU, Asia) / RN2903 (NA)
- Type: LoRaWAN
- Key Features:
 - Operates in sub-GHz frequency bands (868 MHz for EU, 915 MHz for NA)
 - Low power consumption, making it ideal for battery-powered applications
 - Adaptive Data Rate (ADR) for optimized communication
 - AES-128 encryption for secure transmission
 - UART interface for easy integration
- Applications: IoT networks, remote monitoring, smart agriculture, and long-distance wireless communication.

2.6.1.2 Software implementation

The LoRa module utilizes the UART pins from the expansion slot, which are GPIO4 for TX and GPIO5 for RX. These pins correspond to UART3, and to enable it, add the following line to `/boot/firmware/config.txt` (for Bookworm) or `/boot/config.txt` (for Bullseye):

```
dtoverlay=uart3
```

After a reboot, the port `/dev/ttyAMA3` will be enabled.

To test LoRa:

1. **Reset the module** (set pin 26 to high):

```
pinctl set 26 op dh
```

2. **Open the port** with your desired baud rate (e.g., 57600) and leave it listening in one terminal:

```
stty 57600 -F /dev/ttyAMA3 raw -echo
cat /dev/ttyAMA3
```

3. Open another terminal, **send a command** (e.g., `mac pause`) and check the response in the listening terminal:

```
echo -e "mac pause\r" > /dev/ttyAMA3
```

For further details, refer to the [LoRa Command Reference User Guide](#).

2.6.2 4G / NB-IoT / GPRS

2.6.2.1 Module Overview

The 4G / NB-IoT / GPRS expansion board provides cellular communication capabilities, allowing devices to connect to mobile networks using different technologies. The module used is either the LTE IoT 6 Click with the SARA-R412M modem or the LTE IoT 7 Click with the SARA-R422M modem.

- Model: SARA-R412M / SARA-R422M
- Type: LTE Cat M1 / NB-IoT / GPRS
- Key Features:
 - Supports LTE Cat M1, NB-IoT, and 2G fallback (GPRS)
 - Low power consumption for IoT applications
 - Integrated GNSS receiver for positioning (SARA-R422M)
 - Secure communication with TLS/SSL support
 - UART interface for communication
- Applications: Remote monitoring, smart metering, asset tracking, and industrial IoT.

2.6.2.2 Software implementation

The 4G / NB-IoT / GPRS module utilizes the UART pins from the expansion slot, which are GPIO4 for TX and GPIO5 for RX. These pins correspond to UART3, and to enable it, add the following line to `/boot/firmware/config.txt` (for Bookworm) or `/boot/config.txt` (for Bullseye):

```
dtoverlay=uart3
```

After a reboot, the port `/dev/ttyAMA3` will be enabled.

To test the 4G / NB-IoT / GPRS module (*SARA-R412M*):

1. **Download, grant execution permissions and run** the setup script file:

```
wget https://apps.industrialshields.com/main/rpi/gateberry/setup-sara.sh
chmod +x setup-sara.sh
sudo ./setup-sara.sh
```

2. **Enter the required data** when prompted by the script:
 - a. APN credentials
 - i. APN
 - ii. ISP username
 - iii. ISP password
 - b. RAT (Radio Access Technology), select one of the following:
 - i. LTE Cat M1 (4G)
 - ii. NB-IoT (Narrow Band)
 - iii. GPRS / eGPRS.
3. **Script-generated service:** The script creates a service called `pppd` located in `/lib/systemd/system/pppd.service`, which executes a Python file created in `/usr/src/network-setup.py` every time the Raspberry PLC boots.

You can see the log messages with `journalctl -f`.

After 50-120 seconds of executing the script an interface named `ppp0` appears.

4. **Verify that the ppp0 interface has been created** by running:

```
ip a show ppp0
```

5. **Test the connection** to ensure everything is working correctly by pinging:

```
ping 8.8.8.8 -c4 -I ppp0
ping www.google.com -c4 -I ppp0
```

2.6.3 4G LTE-E

2.6.3.1 Module Overview

The 4G LTE Click provides a multimode cellular network solution, enabling LTE Cat 1 and 2G connectivity. It features the compact LARA-R2 series from u-blox, supporting up to three LTE bands and multiple GSM bands.

- Model: LARA-R2 series
- Type: LTE Cat 1 / 2G
- Key features:
 - Supports LTE Cat 1 and 2G connectivity
 - High-speed data rates up to 10.3 Mbps (downlink) / 5.2 Mbps (uplink)
 - Embedded TCP/UDP stack, HTTP/HTTPS transfer protocols
 - IPv4/IPv6 dual-stack support
 - Secondary antenna for RX diversity
 - Antenna detection feature
 - Embedded TLS 1.2 protocol for enhanced security
 - Network status indication
- Applications: Remote monitoring, industrial IoT, smart metering, asset tracking, and high-speed data communication.

2.6.3.2 Software implementation

The 4G LTE-E module utilizes the UART pins from the expansion slot, which are GPIO4 for TX and GPIO5 for RX. These pins correspond to UART3, and to enable it, add the following line to `/boot/firmware/config.txt` (for Bookworm) or `/boot/config.txt` (for Bullseye):

```
dtoverlay=uart3
```

After a reboot, the port `/dev/ttyAMA3` will be enabled.

To test the 4G LTE-E module (*LARA-R2*):

1. **Download, grant execution permissions and run** the setup script file:

```
wget https://apps.industrialshields.com/main/rpi/gateberry/setup-lara.sh
chmod +x setup-sara.sh
sudo ./setup-sara.sh
```

2. **Enter the required data** when prompted by the script:
 - a. APN credentials
 - i. APN
 - ii. ISP username
 - iii. ISP password

- b. RAT (Radio Access Technology), select one of the following:
 - i. LTE Cat M1 (4G)
 - ii. NB-IoT (Narrow Band)
 - iii. GPRS / eGPRS.
3. **Script-generated service:** The script creates a service called *pppd* located in */lib/systemd/system/pppd.service*, which executes a Python file created in */usr/src/network-setup.py* every time the Raspberry PLC boots.

You can see the log messages with “*journalctl -f*”.

After 50-120 seconds of executing the script an interface named *ppp0* appears.

4. **Verify that the ppp0 interface has been created** by running:

```
ip a show ppp0
```

5. **Test the connection** to ensure everything is working correctly by pinging:

```
ping 8.8.8.8 -c4 -I ppp0
ping www.google.com -c4 -I ppp0
```

2.6.4 CANBus

2.6.4.1 Module Overview

The CAN (Controller Area Network) expansion board is designed for industrial and automotive communication, allowing microcontrollers and devices to exchange data efficiently. It is based on the MCP2515 microchip.

- Communication Interface: SPI
- Model: MCP2515
- Type: CAN V2.0B
- Key Features:
 - Speed of up to 1 Mb/s
 - Receive buffer with masks and filters
 - Three transmit buffers with prioritization and abort features
 - High-speed SPI interface (10 MHz)
 - Supports extended frame formats
- Applications: Communication with industrial controllers, automotive ECUs, robotics, and automation systems.

2.6.4.2 Software implementation

The CANBus module utilizes the SPI pins from the expansion slot, which are GPIO9 for MISO, GPIO10 for MOSI, GPIO11 for SCLK and GPIO8 for CS. These pins correspond to SPI0, which must be enabled for the CANBus module to function properly.

1. **Enable SPI:** Run the following command to open the Raspberry Pi configuration menu:

```
sudo raspi-config
```

Navigate to *Interface Options > SPI > Yes* to enable SPI communication.

2. **Enable SPI0 and the MCP2515 overlay:** Add the following line to */boot/firmware/config.txt* (for Bookworm) or */boot/config.txt* (for Bullseye):

```
dtoverlay=mcp2515,spi0-0,oscillator=10000000,interrupt=8,speed=125000
```

This configures the CAN module with a 10 MHz oscillator, sets GPIO8 as the interrupt pin, and defines a bitrate of 125000 bps.

3. **Check if the module is properly initialized:** Run the following command to verify that the MCP2515 module has been detected:

```
dmesg | grep mcp
```

If the module is correctly initialized, you should see messages confirming its activation.

4. **Enable the *can0* interface:** To bring up the CAN0 interface and set the bitrate to 125000 bps, execute:

```
sudo ip link set can0 up type can bitrate 125000
```

This prepares the interface for communication over the CANBus.

5. **Install the *can-utils* package:** To send and receive CAN messages, install the *can-utils* package:

```
sudo apt install can-utils
```

This package provides essential tools for testing and debugging the CANBus.

6. **Test the CANBus communication**
 - a. **Start listening for messages:** Use the *candump* command to listen for incoming CAN messages on the *can0* interface:

```
candump can0
```

- b. **Send a test message:** To send a test message with ID 111 and a payload of FF, run:

```
cansend can0 111#FF
```

2.6.5 GPS

2.6.5.1 Module Overview

The GPS expansion board is based on the u-blox LEA-6S GPS module, providing high-accuracy positioning and navigation.

- Model: u-blox LEA-6S
- Type: GPS receiver module
- Key Features:
 - Supports GPS L1 band (1575.42 MHz)
 - High sensitivity with a cold start time of < 35s
 - Low power consumption for embedded applications
 - Built-in RTC for quick satellite acquisition
 - I2C communication for easy integration
- Applications: Navigation, asset tracking, geofencing, and precision agriculture.

2.6.5.2 Software implementation

The GPS module communicates with the Raspberry Pi via I2C, using GPIO2 for SDA and GPIO3 for SCL. It requires I2C to be enabled for proper operation. The GPS Click Board has a default I2C address of `0x42`, which must be detected by the system before use.

1. **Enable I2C:** Run the following command to open the Raspberry Pi configuration menu:

```
sudo raspi-config
```

Navigate to *Interface Options* > *I2C* > *Yes* to enable I2C communication.

2. **Verify the GPS module is detected:** Install the *i2c-tools* package, which allows scanning of I2C devices:

```
sudo apt install i2c-tools
```

Run the following command to detect connected I2C devices:

```
i2cdetect -y 1
```

Look for address `0x42` in the output, confirming that the GPS module is recognized by the system.

3. **Download and run the GPS test script:** Download the provided *gps.py* script to communicate with the GPS module and run it using Python:

```
wget https://apps.industrialshields.com/main/rpi/gateberry/gps.py
python3 gps.py
```

3 Revision Table

Revision Number	Date	Changes
0	28/12/2021	First implementation
1	25/07/2023	Minor changes
2	03/09/2024	Improved text and formatting
3	05/09/2024	Updated “Raspberry Pi Software” section
4	27/09/2024	Added section 1.3 (“Precautions”)
5	03/12/2024	New version implementation
6	14/01/2025	Removed references to old rpishutdown configuration
7	26/02/2025	Added “Expansion Board Slot” section
8	07/03/2025	Corrections and improved formatting

About Industrial Shields:

Direction: Camí del Grau, 25

Zip/Postal Code: 08272

City: Sant Fruitós de Bages (Barcelona)

Country: Spain

Telephone: (+34) 938 760 191 / (+34) 635 693 611

Mail: support@industrialshields.com