INDUSTRIAL SHIELDS

RASPBERRY PLC FAMILY

RASPBERRY PLC FAMILY – v6

RASPBERRY PLC FAMILY User Guide

Revised, January 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 and get a better understanding of 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.

Warnings:

- Unused pins should not be connected. Ignoring the directive may damage the controller.
- Before using this product, it is the responsibility of the user to read the product's User Guide and all accompanying documentation.
- Industrial Shields PLCs must be powered between 12Vdc and 24Vdc. If a higher voltage is supplied to the equipment it can suffer irreversible damage.
- Maintenance must be performed by qualified personnel familiarised with the construction, operation, and hazards involved with the control.
- Maintenance should be performed with the control out of operation and disconnected from all sources of power.
- The Industrial Shields Family PLCs are Open Type Controllers. It is required that you install the Raspberry PLC in a housing, cabinet, or electric control room. Entry to the housing, cabinet, or electric control room should be limited to authorised personnel.
- Inside the housing, cabinet or electric control room, the Industrial Shields PLC must be at a minimum distance from the rest of the components of a minimum of 25 cm, it can be severely damaged.
- Failure to follow these installation requirements could result in severe personal injury and/or property damage. Always follow these requirements when installing Raspberry family PLCs.
- In case of installation or maintenance of the PLC please follow the instructions marked in the Installation and Maintenance section on the User Guide.
- Do not disconnect equipment when a flammable or combustible atmosphere is present.
- Disconnection of equipment when a flammable or combustible atmosphere is present may cause a fire or explosion which could result in death, serious injury and/or property damage.
- Inside the enclosure, there are supercapacitors of 25F, which can be dangerous. Be careful with them.

Table of Contents

1 General Description RASPBERRY PLC FAMILY Product

1.1 Reference Table

Reference Table			
Model	**RAM Memory**		
	2GB RAM	**4GB RAM**	**8GB RAM**
PLC Raspberry General Family			
Raspberry PLC Ethernet CPU (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0000	012003XX0000	012004XX0000
Raspberry PLC Ethernet 21 I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0200	012003XX0200	012004XX0200
Raspberry PLC Ethernet 42 I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0400	012003XX0400	012004XX0400
Raspberry PLC Ethernet 58 I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0600	012003XX0600	012004XX0600
Raspberry PLC Ethernet 19R I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0100	012003XX0100	012004XX0100
Raspberry PLC Ethernet 38R I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0300	012003XX0300	012004XX0300
Raspberry PLC Ethernet 57R I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0500	012003XX0500	012004XX0500
Raspberry PLC Ethernet 38AR I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0700	012003XX0700	012004XX0700
Raspberry PLC Ethernet 57AAR I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0800	012003XX0800	012004XX0800
Raspberry PLC Ethernet 50RRA I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX0900	012003XX0900	012004XX0900
Raspberry PLC Ethernet 53ARR I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX1000	012003XX1000	012004XX1000
Raspberry PLC Ethernet 54ARA I/Os Analog/Digital PLUS (Raspberry Pi 4B X GB RAM Included + pSLC SD W/Linux)	012002XX1100	012003XX1100	012004XX1100

The "**XX**" in the reference number indicates key specifications:

- **First Character**: Expansion module on Slot 1.

- **Second Character**: Expansion module on Slot 2.

1.2 Zones Table

Model	Zone 0	Zone A	Zone B	Zone C
Raspberry PLC Ethernet CPU	☑	-	-	-
Raspberry PLC 21+	☑	Analog / Digital	-	-
Raspberry PLC 42+	☑	Analog / Digital	Analog / Digital	-
Raspberry PLC 58+	☑	Analog / Digital	Analog / Digital	Analog / Digital
Raspberry PLC 19R+	☑	Relay	-	-
Raspberry PLC 38R+	☑	Relay	Relay	-
Raspberry PLC 57R+	☑	Relay	Relay	Relay
Raspberry PLC 38AR+	☑	Analog / Digital	Relay	-
Raspberry PLC 53ARR+	☑	Analog / Digital	Relay	Relay
Raspberry PLC 57AAR+	☑	Analog / Digital	Analog / Digital	Relay
Raspberry PLC 54ARA+	☑	Analog / Digital	Relay	Analog / Digital
Raspberry PLC 50RRA+	☑	Relay	Relay	Analog / Digital

1.3 Measures Table

Model	Height (mm)	Width (mm)	Depth (mm)	Weight (g)
Raspberry PLC Ethernet CPU	119.5	84.60	101	373
Raspberry PLC 21+	119.5	109.20	101	490
Raspberry PLC 42+	119.5	133.80	101	598
Raspberry PLC 58+	119.5	158.40	101	710.5
Raspberry PLC 19R+	119.5	109.20	101	490
Raspberry PLC 38R+	119.5	133.80	101	598
Raspberry PLC 57R+	119.5	158.40	101	710.5
Raspberry PLC 38AR+	119.5	133.80	101	598
Raspberry PLC 53ARR+	119.5	158.40	101	710.5
Raspberry PLC 57AAR+	119.5	158.40	101	710.5
Raspberry PLC 54ARA+	119.5	158.40	101	710.5
Raspberry PLC 50RRA+	119.5	158.40	101	710.5

1.4   Zone – Nomenclature

The nomenclature shown in this point will be used in the whole User Guide, so it is important to understand it, as the different parts of the PLCs are referenced.

The nomenclature to differentiate the zones is based on Alphanumeric values, being 0 the internal communication shield and A, B or C the I/Os or Relay shield.
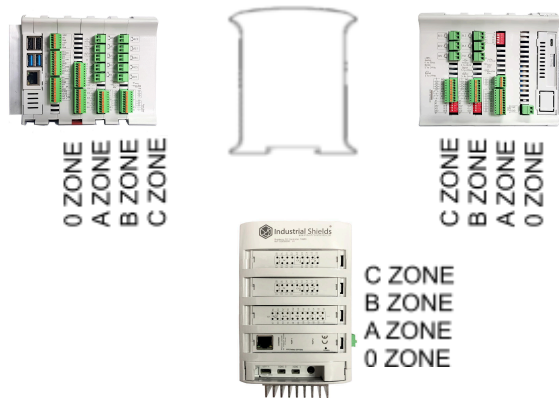
- The inputs in the zone A/B/C are named IX.X, being X any number suitable in the Shield. Outputs are named as QX.X.

1.5   Zone Distribution
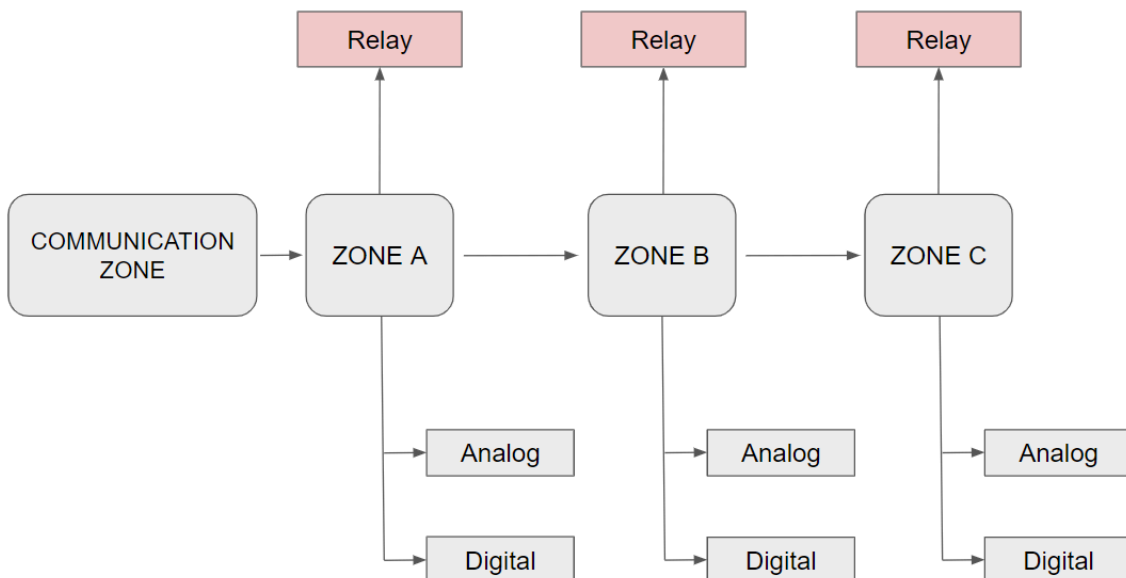
This is the possible zones scheme, whether the zone is Analog/Digital or Relay depends on your device, you can consult it on the [Zones Table](). The distribution of the different features that provide the Raspberry PLC Family are described below.

1.6 Zone Features

Shield	Zone Characteristics	
Communication Shield	(2x) Ethernet (4x) USB (1x) I2C (1x) TTL (2x) HALF Duplex RS-485 (1x) SPI external Port (1x) RTC	(1x) Bluetooth (1x) Wi-Fi (1x) μSD Socket (x2) μHDMI (x2) PREL (x2) OUT (Opto-isolated output)
Analog Shield	13 Inputs:	13 Digital inputs, (6 of which can work as Analog Input and 7 of which work as Opto-isolated inputs*)
	8 Outputs:	8 Digital Outputs, 3 of which can work as Analog Output
Relay Shield	6 Inputs:	6 Digital Inputs, (4 of which can work as Analog Input and 2 of which work as Opto-isolated inputs*)
	11 Outputs:	8 Relay Outputs
		3 Digital Outputs, 3 of which can work as Analog Output

*2 of them can work as interrupt inputs

1.7   Mechanical dimension

These are the possible mechanical dimensions, you can consult the specific measures for your device on the [measures table](#).
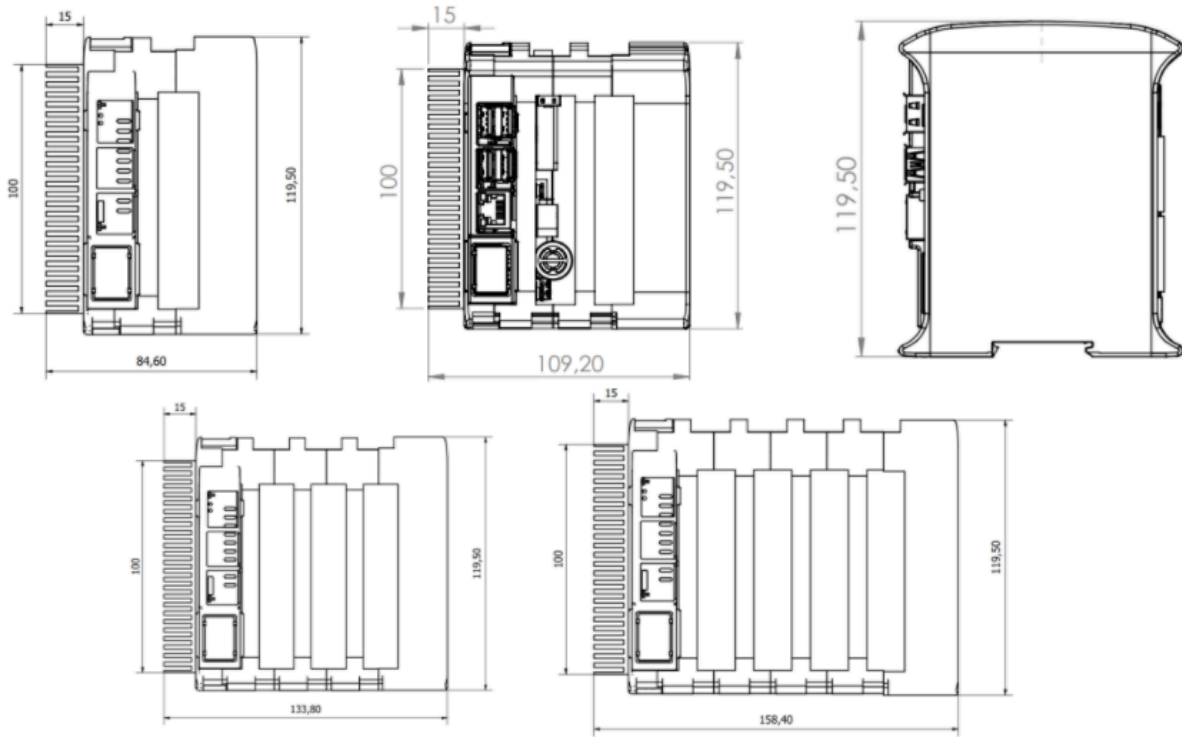
1.8 General Features

CONNECTABLE PLC RASPBERRY 24 VCC		
MODEL TYPE	**Controller General Specifications**	**Information**
Input Voltage	12 to 24 Vdc	Fuse Protection (2.5A) Polarity protection
I max.	1.5 A	
Size	Consult the Measure Table section	
Clock Speed	1.5 GHz	
Flash Memory	-	
SRAM	2 - 4 - 8 GB	
Communications	I2C – Ethernet (x2) – USB (x4) – (x2) RS485 – SPI – WiFi – Bluetooth - Serial TTL - µSD - RTC	MAX485 – W5500
Digital GPIO **(3.3 V)**	3.3 Vdc ⎓	SCL/SDA: Rising edge time: 178 ns Falling edge time: 2.72 us TX/RX/SO/SI/SCK/GPIO8: Rising edge time: 162 ns Falling edge time: 162 ns
PRELAY (Opto-mosfet) **(0-60 V)**	0 to 60 Vdc I max: 500 mA ⎓	Switching speed: Maximum Frequency: 250 Hz Turn-on time (tON): maximum 3 ms Turn-off time (tOFF): maximum 1 ms
Opto Isolated Output **(5-24 V)**	5 to 24 Vdc I max: 70 mA ⎓	With 0 delay, freq is 495 Hz with pulse width of 1.888 ms Rising edge time: 744 ns Falling edge time: 55.6 us
An/Dig Input 12 bits **(0-10Vcc)**	0 to 10 Vdc Rated Voltage: 10 Vdc 5 to 24 Vdc I min: 2 to 12 mA Galvanic Isolation Rated Voltage: 24 Vdc ⎓	Max read time: 840 us *Check Section 11 for digital inputs threshold detection

Digital Isolated Input (24Vcc)	5 to 24 Vdc I min: 2 to 12 mA I max: 300 mA Galvanic Isolation Rated Voltage: 24 Vdc ‐‐‐	Max read time: 430 us
* Interrupt isolated Input HS (24Vcc)	5 to 24 Vdc I min: 2 to 12 mA Galvanic Isolation Rated Voltage: 24 Vdc ‐‐‐	
Analog Output 12 bits (0 - 10 Vcc)	0 to 10 Vdc I max: 20 mA Separated PCB Ground ‐‐‐	Max write time: 1 ms Rising edge time: 120 ms Falling edge time: 173 ms
Digital Isolated Output (24 Vcc)	5 to 24 Vdc I max: 70 mA Galvanic Isolation Diode Protected for Relay ‐‐‐	With 0 delay, freq is 495 Hz with pulse width of 1.888 ms Rising edge time: 744 ns Falling edge time: 55.6 us
Relay Output (30Vdc / 250 Vac)	30 Vdc (Imax: 3 A) 250 Vac (Imax: 5 A) Galvanic Isolation Diode Protected for Relay ‐‐‐	
PWM Isolated Output 12 bits (24 Vcc)	5 to 24 Vdc I max: 70 mA Galvanic Isolation Diode Protected for Relay ‐‐‐	
Expandability	I2C: 127 elements – Serial Port RS485	
* By using this type of signal can no longer use Digital signal (24 Vdc)		

1.9   Power consumption of the Raspberry PLC Family PLCs

Model	Current resting (mA)	Power resting (W)	Current active (mA)	Power active (W)	Current stress (mA)	Power (W)
Ethernet CPU	160	3.84	-	-	312	7.488
21+	150	3.6	164	3.936	290	6.96
42+	157	3.768	185	4.44	320	7.68
58+	160	3.84	200	4.8	338	8.112
19R+	155	3.72	260	6.24	395	9.48
38R+	157	3.768	358	8.592	500	12
57R+	180	4.32	482	11.568	625	15
38AR+	157	3.768	267	6.408	410	9.84
53ARR+	160	3.84	375	9	325	7.8
57AAR+	160	3.84	280	6.72	418	10.032
54ARA+	160	3.84	280	6.72	423	10.152
50RRA+	160	3.84	375	9	522	12.528

The power consumption is measured at 24Vdc.

The **current resting** column denotes the minimum current drawn by the PLC when all outputs are at a LOW level, representing the PLC's baseline power consumption.

In contrast, the **current active** column reflects the current drawn by the PLC when all outputs are at a HIGH level, and there are no connections to any of the outputs. While this value is not the absolute maximum the Raspberry based PLC can draw, it serves as a reference point for expected current values.

Finally, the **current stress** column refers to the current drawn by the PLC when all four cores are under 100% load, simulated using the Linux command "stress --cpu 4." This method provides a more detailed measurement of the PLC's power consumption for a comprehensive table.

2   Technical Specifications:

2.1   General Specifications:
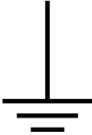
Item		RASPBERRY PLC
Power supply voltage	DC power supply	12 to 24 Vdc
Operating voltage range	DC power supply	11.4 to 25.4 Vdc ▬▬▬
Power consumption	DC power supply	30 W max.
External power supply	Power supply voltage	24 Vdc ▬▬▬
	Power supply output capacity	700 mA
Dielectric strength		1500 Vac at 50/60 Hz for one minute with a leakage current of 10 mA max.
Shock resistance		50 m/s² in the X, Y and Z direction 3 times each, complying with the IEC-60068-2-27:2008 standard.
Ambient temperature (operating)		0 º to 50 ºC with Raspberry OS Lite / 0 º to 45 º with Raspberry OS Desktop
Ambient humidity (operating)		10 % to 90% (no condensation)
Ambient environment (operating)		With no corrosive gas
Ambient temperature (storage)		-20 º to 60 ºC
Power supply holding time		2 ms min.
Weight		Consult the Measure Table section

2.2   Performance Specification:

Raspberry Board	Raspberry Pi 4
I/O control method	Combination of the cyclic scan and immediate refresh processing methods.
Programming language	Linux applications: Bash Scripts, Python, C++, Node-RED, etc.
Program capacity (SRAM)	2 - 4 - 8GB
EEPROM	4 MB/512KB
Clock Speed	1.5GHz
CPU	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

2.3 Symbology

Table that includes all the symbology used in the serigraph of the RASPBERRY PLC FAMILY:

Symbol	Standard No. / Standard Title	Standard Reference No. / Symbol Title	Symbol Meaning
▬ ▬▬ ▬	IEC 60417 / Graphical symbols for use on equipment	5031 / Direct Current	Indicates that the equipment is suitable for direct current only; to identify relevant terminals
∿	IEC 60417 / Graphical symbols for use on equipment	5032 / Alternating Current	Indicates that the equipment is suitable for alternating current only; to identify relevant terminals
⊓	IEC 60417 / Graphical symbols for use on equipment	5130 / Pulse General	To identify the control by which a pulse is started.
⏚	IEC 60417 / Graphical symbols for use on equipment	5017 / Earth, Ground	To identify an earth (ground) terminal in cases where neither the symbol 5018 nor 5019 is explicitly required.
⊗	IEC 60417 / Graphical symbols for use on equipment	5115 / Signal lamp	To identify the switch by means of which the signal lamp(s) is (are) switched on or off.
CE	2004/108/EC / Electromagnetic Compatibility	CE Marking	CE marking indicates that a product complies with applicable European Union regulations
⚠	ISO 7000/ Graphical symbols for use on equipment	0434B / Warning symbol	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury
⚡	ISO 7000/ Graphical symbols for use on equipment	5036 / Dangerous Voltage	To indicate hazards arising from dangerous voltages

3   Precautions

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

3.1 Raspberry Board

The RASPBERRY PLC FAMILY includes a Raspberry Pi 4 Board as controller.

3.2 Intended Audience

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

3.3 General Precautions

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

Before using the RASPBERRY PLC FAMILY 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 Raspberry PLC 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 Raspberry PLC.

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.

4   How to connect PLC to power supply

- Raspberry PI Family PLCs are 12-24Vdc supplied. IMPORTANT: The polarity **IS NOT REVERSIBLE!**

- Make sure that the live and GND connector of the power supply match the PLC.

- Make sure that the power supply mains output is not higher than 24Vdc.
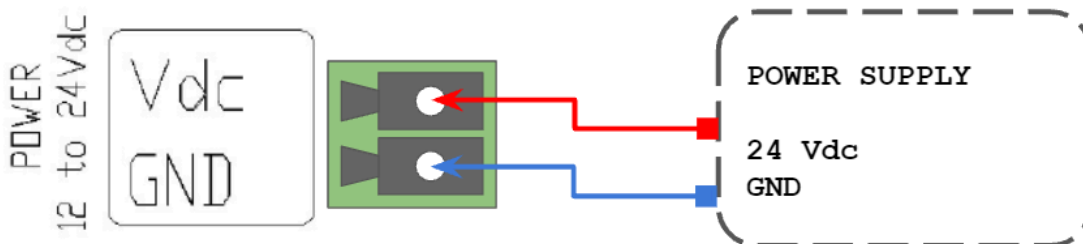
- Suggested power supply:

Compact DIN rail power supply. Assembled on 35mm DIN Rail:
- 24Vdc
- 10A
- 240W

Industrial Shields power supplies provide parallel operation, overvoltage protection, and overcurrent protection. There is a LED inductor for power status, the power supply is certified according to UL.

The standard, Part 1 of IEC 61010, sets the general safety requirements for the following types of electrical devices and their accessories, regardless of where use of the device is intended.

The equipment must be powered from an external power source in accordance with IEC 61010-1, whose output is MBTS and is limited in power according to section 9.4 of IEC 61010-1.
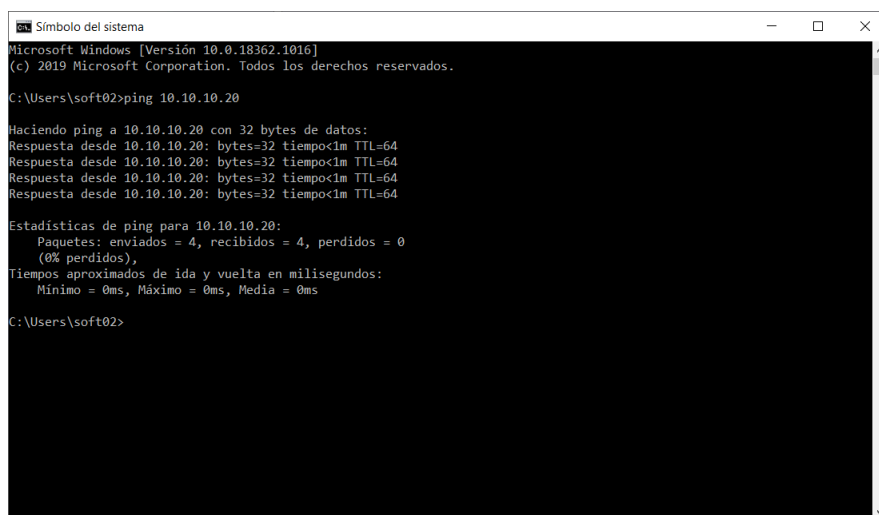
⚠ Warning:

Once the equipment is installed inside an electrical cabinet, the MTBS cables of the equipment must be separated from the dangerous voltage cables.

5 How to access to the Raspberry PLC

5.1 Raspberry PLC access

To facilitate the connection to the Raspberry PLC, our company has set a default IP. For the first time of use, an Ethernet connection between the Raspberry PLC and a PC will be necessary. The Raspberry is given with the local IP address **10.10.10.20/24*(eth0) and 10.10.11.20/24*(eth1)**, the default user is **pi** and the password is **raspberry**. For connecting to it, you must change your local address to be in the same local network as the Raspberry. After the first connection you can add users or change each user's password anytime. In order to know if the Raspberry is connected and the Ethernet connection is going on, a *ping* command can be run on the terminal (for windows users enter *cnm* in the windows searching tab to open it):

*Note: To use this local IP address you must connect the Ethernet cable to the Ethernet port located on the right side of the PLC. Do not use the upper side Ethernet port for this purpose.

The steps to follow for both Linux and Windows are explained below:

5.1.1 Linux

In order to access the Raspberry PLC, Linux users have to enter by SSH protocol, which should have been installed before by the user. As has been said, the Raspberry is given with a local IP address by default for making an easier path for the customer. So, first of all we have to check if our local **Ethernet Network** is set with the same local IP address as our device. In case that you do not know how to set it, check the section 6 of the guide. As have been mentioned before, you can use the *ping* command with the address 10.10.10.20 as option:
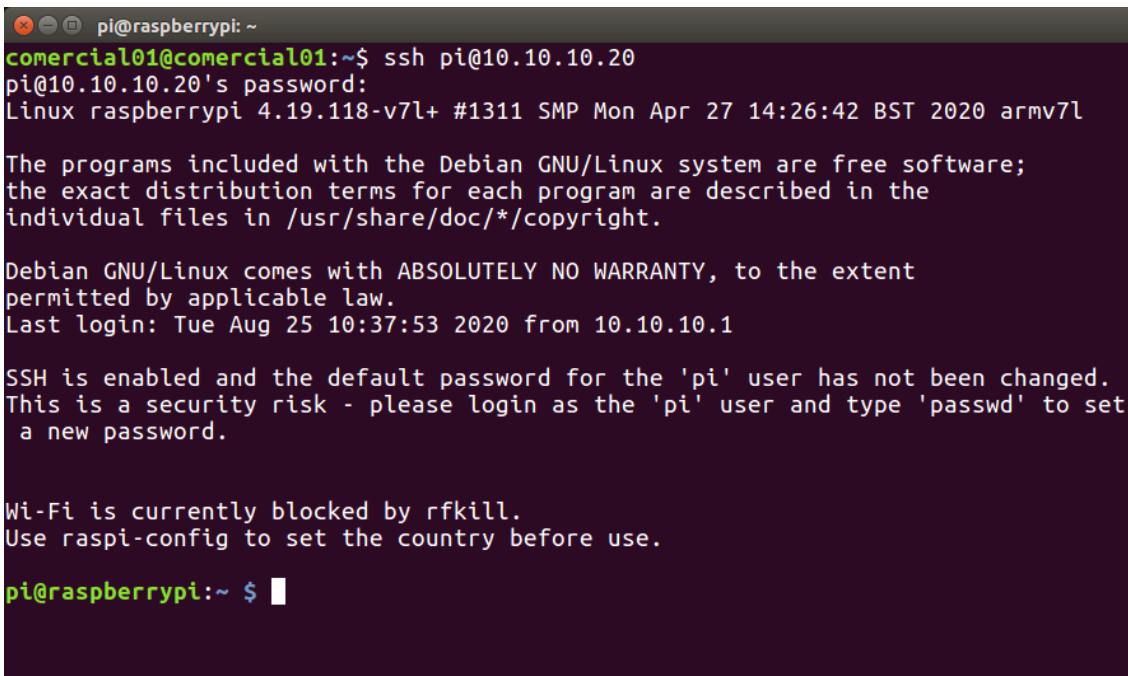
```
ping 10.10.10.20
```

If the ping works successfully means that our connection is ready. If not, you can check section 6 again or read our blog about changing the local IP.

In order o the Raspberry PLC using SSH (Secure Shell), run the command shown below:

```
ssh pi@10.10.10.20
```

The connection will start, but first the password provided before will be required. If the entered username and password are correct, the SSh session will be started.

When an SSH connection is made for the first time, the server delivers the server's public key to the SSH client. The system will alert you to this and offer you the option of accepting the key or rejecting it. You have to accept the key, as it will be stored in the register and will be used to contrast it with the one sent by the server on each connection. If for some reason the key changes, a new notice is generated in which the authenticity of the received key will be raised, since someone could be posing as the server to which we want to connect.

Once connected, there will be full access to the Raspberry PLC and the user will be able to control it and set the functions needed. Also new users or the password could be changed as the device IP. The commands for going through the Raspberry are the same as the Linux terminal. The most useful are the following:

- `cd`: to navigate through the Linux files and directories.
- `ls`: is used to view the contents of a directory.
- `cat`: used to list the contents of a file on the standard output.

- `mkdir`: use this command to create a new directory.
- `rm`: is used to delete directories and the contents within them.
- `touch`: allows you to create a blank new file.

In order to logout and close the connection, just type "exit" in the Raspberry terminal.

5.1.2 Windows

For Windows users, we recommend using the open-source terminal emulator PuTTy SSH client. The latest release of PuTTY can be downloaded from the official web, which contains download links for the latest released version of PuTTY:

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

To connect to the RPI PLC, you must change your local address to be in the same local network as the Raspberry. You can use the *ping* command with the address 10.10.10.20 as an option in the Windows terminal. The Windows terminal can be found typing *cnm* on the windows search bar.

After having installed PuTTY, run the program and the following window should be opened. If it is not the same, click on the top left corner selecting the option *Session.*
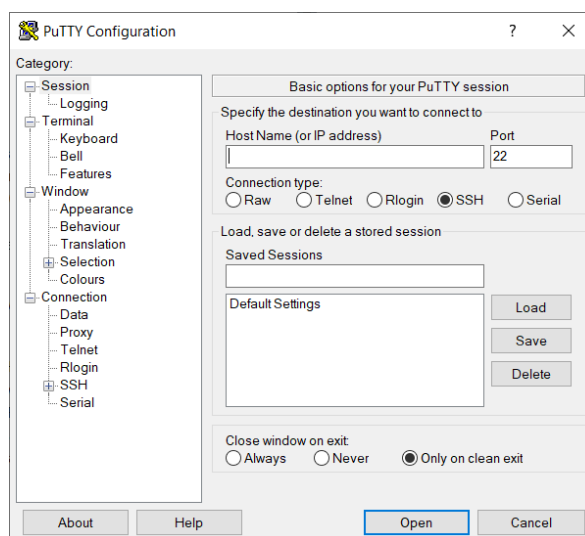
To set up PuTTY the Raspberry IP must be entered on the *HostName* field and make sure that the port 22 is being used and the *SSH* option is selected on the *Connection Type*. After all the settings have been set up, click on the *Open* button to run the program. For a faster start in subsequent sessions, before running the program, enter a name on *Saved Sessions* and click on the *Save* button (a double click on the named configuration will execute the program with the saved configurations).
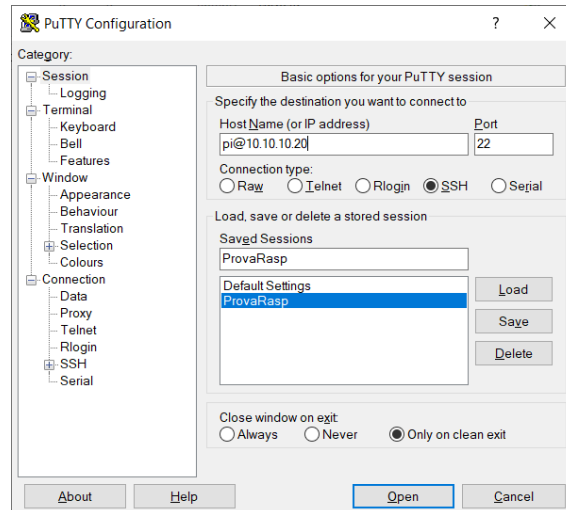
When you start the connection, the terminal window will be opened. Enter your username *pi* and press *Enter*, then enter your password ***raspberry***. If the username and password are correct, the SSH session will be started.

When an SSH connection is made for the first time, the server delivers the server's public key to the SSH client. PuTTy will alert you to this and offer you the option of accepting the key or rejecting it. You have to accept the key, as it will be stored in the register and will be used to contrast it with the one sent by the server on each connection. If for some reason the key changes, PuTTy will generate a new notice in which the authenticity of the received key will be raised, since someone could be posing as the server to which we want to connect.

Once connected, there will be full access to the Raspberry PLC and the user will be able to control it and set the functions needed. Also, new users or the password could be changed as the device IP. The commands for going through the Raspberry are the same as the Linux terminal. The most useful are the following:

- `cd`: to navigate through the Linux files and directories.
- `ls`: is used to view the contents of a directory.
- `cat`: used to list the contents of a file on the standard output.
- `mkdir`: use this command to create a new directory.
- `rm`: is used to delete directories and the contents within them.
- `touch`: allows you to create a blank new file.

To close the connection, just type "exit" in the Raspberry terminal.

6   How to change the IP

To access the Raspberry PLC, the PLC must be on the same local network as the computer which we are working with. For changing the IP we must have connected our PC to an Ethernet connection. By default, the computers are given an IP address of the 192.xxx.xxx.xxx or 172.xxx.xxx.xxx type but will be necessary to change it to the 10.10.10.xxx as the PLC. As we will work with an Ethernet interface, the WiFi interface IP must not be changed.

6.1 Linux

For changing the IP it is not necessary to know the actual one, but it is essential to know the interface name we are working on. To know it, we will run an *ip* command with the option *a*.

```
ip a
```

All the PC interfaces will be shown and we will have to look for the one named **enp,** as it will be the Ethernet one. In order to change it, we will use the **ifconfig** program. To change your IP address on Linux, use the **ifconfig** command followed by the name of your network interface and the new IP address to be changed on your computer. This command needs to be called as an administrator. In order to do this, use the **su** or **sudo** commands.

```
sudo ifconfig <interface_name> 10.10.10.1 netmask 255.255.255.0
```

Now the IP should be changed. In order to check it, run again the *ip a* command. If not, repeat the process or check out our website's blog about changing the IP addresses.

6.2 Windows

The first step is to open the Control Panel. The quickest way to get to it is by clicking on Start and typing its name. In the Control Panel, click on the **Network and Internet** category and then click on **Network and Sharing Center**. If you are in icon view, just click directly on **Network and Sharing Center**. An Ethernet connection must be shown, if not make sure that the Ethernet wire is connected to the Raspberry PLC.
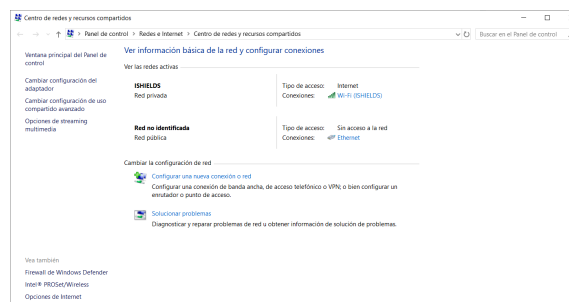
Then we must click on the Ethernet connection and a window about the state of it will be shown, where we can check all the details of the Ethernet connection. For changing the IP, click on the **Properties** button. Then search for **Internet Protocol Version 4 (TCP/IPv4)** and click again on the **Properties** button.
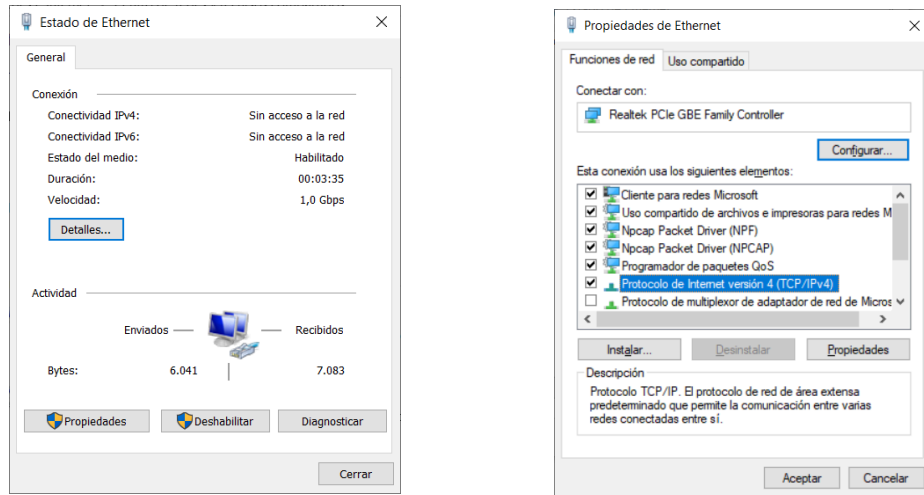
Finally, a window will be displayed where we will be able to change the IP address. Choose the second option and enter the following IP as the following image. Then choose the Accept button and in the **Proprieties of Ethernet** do the same. Finally close the window and the IP will have been changed. In order to check it, click on the network details or run the **ipconfig** command on the Windows terminal. If not, repeat the process or check out our website's blog about changing the IP addresses.
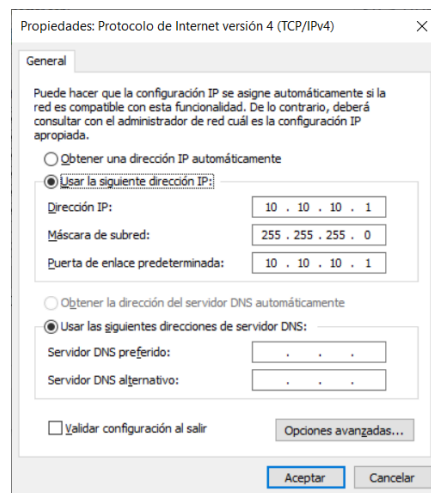
7 How to transfer files from the PC to the Raspberry PLC

Sometimes it is necessary to transfer files from our computer to the Raspberry PLC. The best way to do it is using the Secure Copy Protocol (SCP), a protocol that allows the secure transfer of files between a local host and a remote host based on the Secure Shell Protocol (SSH).

Linux systems carry a built-in SCP, allowing you to copy files using the scp command. However, Windows systems need to use an external app to be able to use this protocol. We recommend working with WinSCP.

Some directories of the Raspberry Pi PLC have administrator permissions, which will not allow transferring files directly to them. As a result, the best option is to copy the files to the **/home/pi** directory and then move them to the folder that you wish.

7.1 Windows

Once downloaded the WinSCP program, we will execute it and a window will open to log in. We will have to choose the protocol which we are using, the port and the IP address of the PLC where we want to connect with his user and password.

In this example, we will be using the default options for the protocol (SFTP) and the port (22). The Raspberry Pi PLC default IP is 10.10.10.20, the user is **pi** and the password is raspberry. After setting all the requirements, press the connect button and the connection will start.
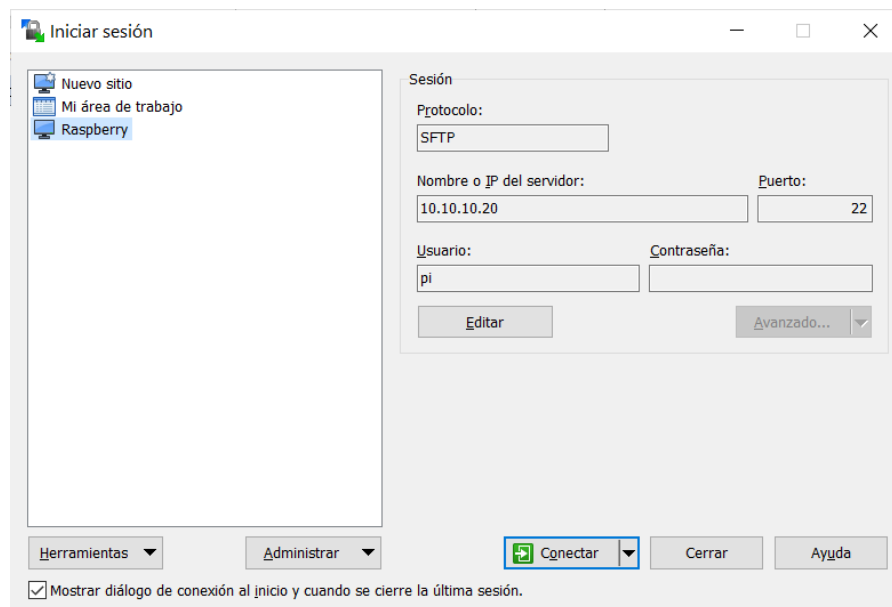
A new window will be opened, seeing the PC files on the left and the Raspberry Pi PLC on the right of the screen. There you can search on the different directories and files and for copying one file from one host to another, just click and drag from one place to the other.
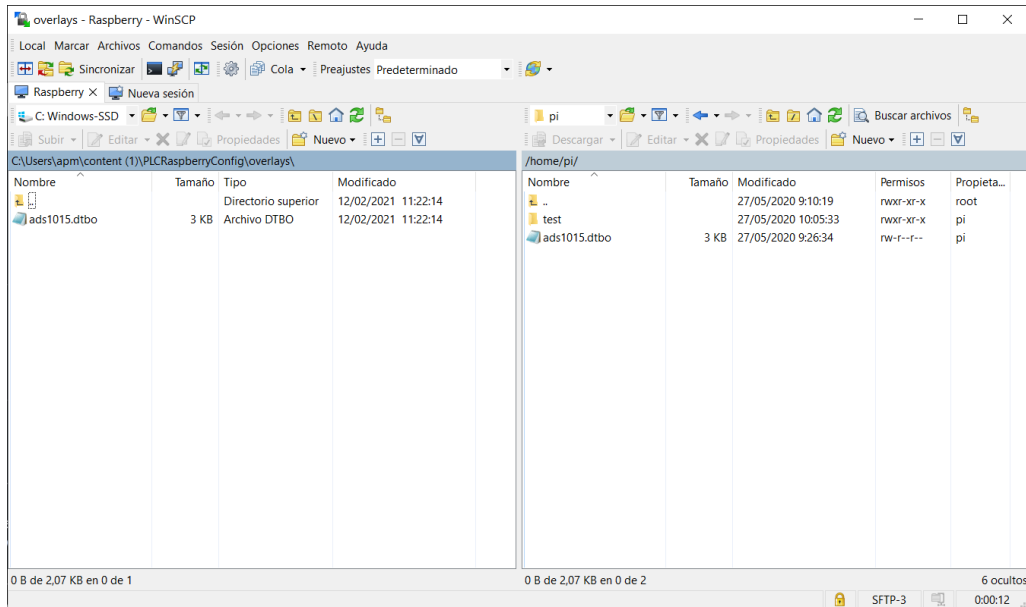
7.2 Linux

Linux systems have the SCP protocol included on them by default. In order to work with them it is only necessary to select a source and the destination. The command syntax is the following:

```
scp <source> <destination>
```

To copy a file from our personal computer to the home directory of the Raspberry Pi PLC, do the following:

```
scp /path/file pi@10.10.10.20:/home/pi
```

8 Raspberry PLC Family Pinout

8.1 I/Os Table

Model	Reference	IOs Table				
		Analog Input	Digital Isolated Input	Digital Isolated Output	Digital/Analogic Output	Relay Output
Normal	01200XXX0000	0	1	0	0	0
21+	01200XXX0200	6	7	5	3	0
42+	01200XXX0400	12	14	10	6	0
58+	01200XXX0600	18	21	15	9	0
19R	01200XXX0100	4	2	0	3	8
38R	01200XXX0300	8	4	0	6	16
57R	01200XXX0500	12	6	0	9	24
38AR	01200XXX0700	10	9	5	6	8
57AAR	01200XXX0800	16	16	10	9	8
50RRA	01200XXX0900	14	11	5	9	16
53ARR	01200XXX1000	14	11	5	9	16
54ARA	01200XXX1100	16	16	10	9	8

8.2   0 Zone connection (Communications)

Base (common unit) 0 Zone		
PLC Connector	*Raspberry Pin*	Function
3.3V	-	Power Supply
5V	-	Power Supply
GND	-	GND
PREL A	-	-
PREL B	-	-
INT31/GPIO4	-	GND
SCL	*GPIO3*	I2C
SDA	*GPIO2*	I2C
GND	-	GND
TX	*GPIO14*	SerialTTL
RX	*GPIO15*	SerialTTL
GND	-	GND
B2-	-	RS485(2)
A2+	-	RS485(2)
OUT 2	-	5V digital/PWM out
IN 2	-	5V digital/PWM in
B1-	-	RS485(1)
A1+	-	RS485(1)
OUT 1	-	5V digital/PWM out
SO	*GPIO 9*	SPI
SI	*GPIO 10*	SPI
SCK	*GPIO 11*	SPI
GPIO8	*GPIO 8*	Raspberry Direct Pin
IN 1	-	5V digital/PWM in
VCC ISO	-	Power Supply
GND ISO	-	GND

Communication Pins

3.3V
5V
Gnd
PREL A
PREL B
GPIO4

SCL
SDA (5V)
Gnd
TX
RX
GND
B2- RS485
A2+
OUT2
IN2

B1- RS485
A1+
OUT1
SO
SI (3.3V)
SCK
GPIO8
IN1
VCC ISO
GND ISO

SS: Chip Select pins. These pins can act as TTL, so they can work for the Chip Select pin af any device

Right Side

ETHERNET 0

USB 3.0 (x2)

USB (x2)

Left Side

Micro SD
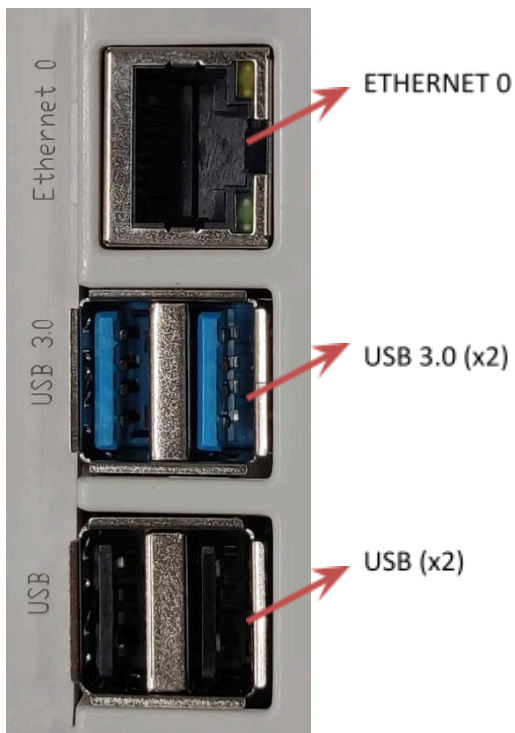
Upper Side
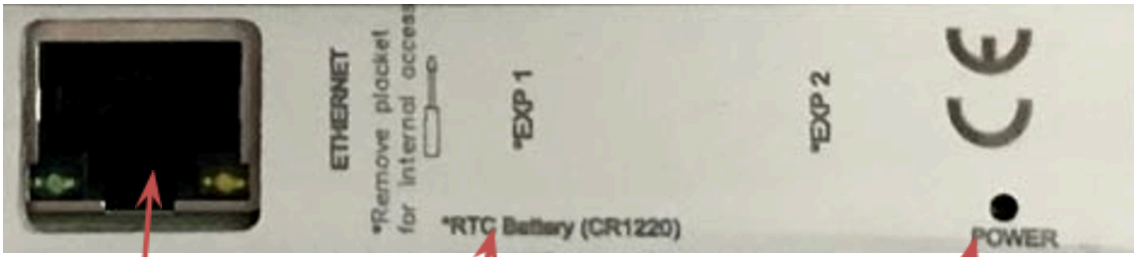
Labels: ETHERNET 1, RTC BATTERY, POWER LED
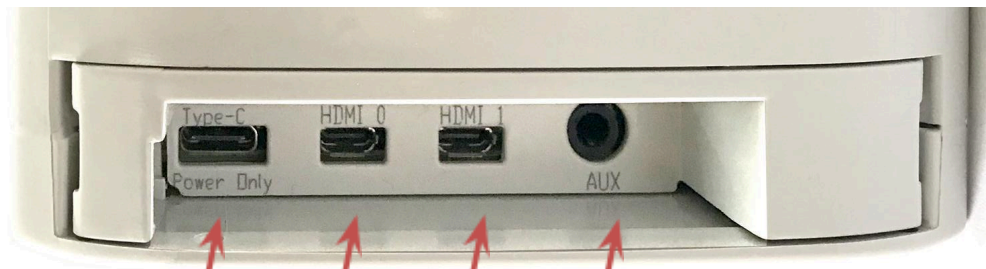
Labels: Type-C (Power Only), HDMI 0, HDMI 1, AUX

8.3 Analog/Digital Zone connection (I/Os)

X Zone	
RPI PLC Connector	**Function**
IX.12	Analog / Digital In
IX.11	Analog / Digital In
IX.10	Analog / Digital In
IX.9	Analog / Digital In
IX.8	Analog / Digital In
IX.7	Analog / Digital In
(-)IX.6	GND I0.6
IX.6/INT[1]	Interrupt X In
(-)IX.5	GND I0.5
IX.5/INT[1]	Interrupt X In
(-)IX.4	GND I0.4
IX.4	Digital Input
(-)IX.3	GND I0.3
IX.3	Digital Input
(-)IX.2	GND I0.2
IX.2	Digital Input
(-)IX.1	GND I0.1
IX.1	Digital Input
(-)IX.0	GND I0.0
IX.0	Digital Input
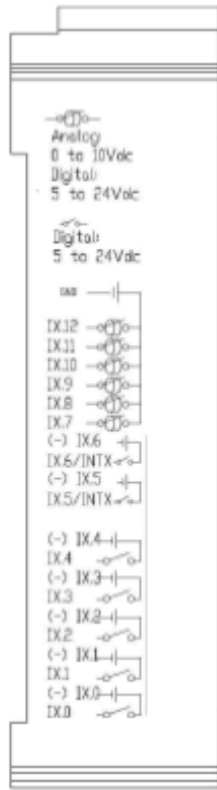
Analog / Digital Inputs

Digital / Interrupt Inputs (isolated)

Digital inputs (isolated)

Top Zone
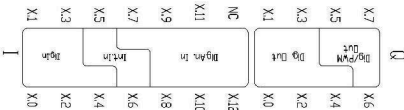
Led indicator I/Os state

X Zone	
RPI PLC Connector	**Function**
GND	GND
AX.7[2]	Analog Out
AX.6[2]	Analog Out
AX.5[2]	Analog Out
Q/Vdc	External Isolated Out Vdc
COM(-)	External Isolated Out GND
QX.7[2]	Digital/PWM Out
QX.6[2]	Digital/PWM Out
QX.5[2]	Digital/PWM Out
QX.4	Digital Out
QX.3	Digital Out
QX.2	Digital Out
QX.1	Digital Out
QX.0	Digital Out
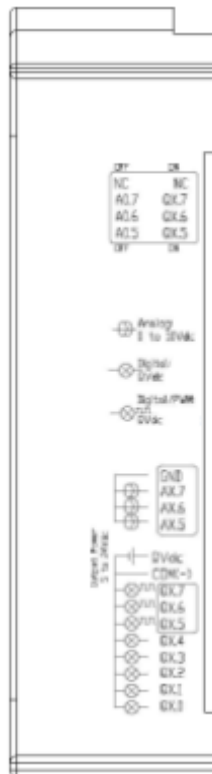
Configuration Switch
(See Section 8 to select the correct configuration for outputs)

Analog Outputs

Voltage Supply/Reference for

Digital/PWM Outputs (isolated)

---

[1] See the Interrupt Equivalence Table in Section 8.5.

[2] See Section 9 to select the correct configuration for outputs.

8.4 Relay Zone connection (I/Os)

X Zone	
RPI PLC Connector	Function
RX.8	Relay Out
RX.7	Relay Out
RX.6	Relay Out
RX.5	Relay Out
RX.4	Relay Out
AX.2[3]	Analog Out
AX.1[3]	Analog Out
AX.0[3]	Analog Out
GND	GND
QX.2[3]	Digital/PWM Out
QX.1[3]	Digital/PWM Out
QX.0[3]	Digital/PWM Out
GNDCOM	External Isolated Out GND
24VCOM	External Isolated Out Vdc
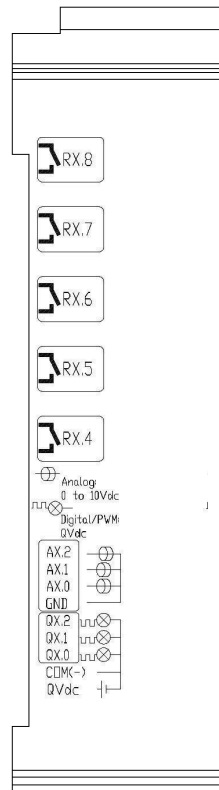
Relay Outputs

Analog Outputs

PWM/Digital Outputs

Top Zone

Led indicator I/Os state

X Zone	
RPI PLC Connector	Function
RX.3	Relay Out
RX.2	Relay Out
RX.1	Relay Out
GND	GND
IX.5	Analog/Digital Input
IX.4	Analog/Digital Input
IX.3	Analog/Digital Input
IX.2	Analog/Digital Input
IX.1/INT[4]	Interrupt **X** In
(-)IX.1	GND I0.1
IX.0/INT[4]	Interrupt **X** In
(-)IX.0	GND I0.0
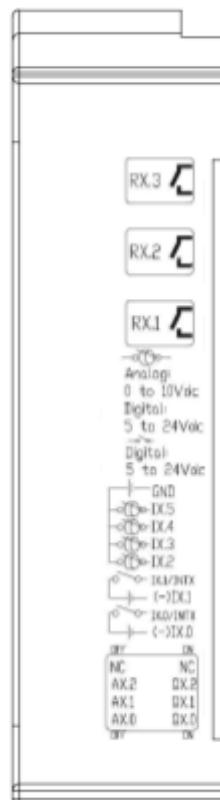
Relay Outputs

Analog Inputs

Interrupts/Digital Inputs (Isolated)

Configuration Switch
(See Section 8 to select the correct configuration for outputs)

---

[3] See Section 8 to select the correct configuration for outputs.
[4] See the Interrupt Equivalence Table in Section 8.5.

8.5 Interrupt equivalence table

See the Reference Table in order to know the specific device details.

ZONE	ANALOG/DIGITAL	RELAY	RASPBERRY PIN
A	0.5	0.0	GPIO13
A	0.6	0.1	GPIO12
B	1.5	1.0	GPIO27
B	1.6	1.1	GPIO5
C	2.5	2.0	GPIO26
C	2.6	2.1	GPIO4

*NOTE: It have to be taken into account that you can use the one linked to GPIO12 (I0.6/I0.1) or the OPTO_IN_1, and you can use the one linked to GPIO5 (I1.6/1.1) or the OPTO_IN_2.

9   Switch Configuration

9.1   Analog / Digital Zone

ZONE A		
SWITCH	ON	OFF
NC	-	-
Q0.7	Q0.7	A0.7
Q0.6	Q0.6	A0.6
Q0.5	Q0.5	A0.5

ZONE B		
SWITCH	ON	OFF
NC	-	-
Q1.7	Q1.7	A1.7
Q1.6	Q1.6	A1.6
Q1.5	Q1.5	A1.5

ZONE C		
SWITCH	ON	OFF
NC	-	-
Q2.7	Q2.7	A2.7
Q2.6	Q2.6	A2.6
Q2.5	Q2.5	A2.5

9.2   Relay Zone

ZONE A		
SWITCH	ON	OFF
NC	-	-
Q0.2	Q0.2	A0.2
Q0.1	Q0.1	A0.1
Q0.0	Q0.0	A0.0

ZONE B		
SWITCH	ON	OFF
NC	-	-
Q1.2	Q1.2	A1.2
Q1.1	Q1.1	A1.1
Q1.0	Q1.0	A1.0

ZONE C		
SWITCH	ON	OFF
NC	-	-
Q2.2	Q2.2	A2.2
Q2.1	Q2.1	A2.1
Q2.0	Q2.0	A2.0

For the Analog Shield if a switch is set to ON, it can only act as Digital Output. If it is set to OFF, it can only act as an Analog Output.

If it is desired to use a Digital Output the pin must be set to ON and the pin that will provide this digital output is represented with QX.X, being X any number of the tables above.

If it is desired to use an Analog Output the pin must be set to OFF and the pin that will provide this analog output is represented with AX.X, being X any number of the tables above.

10 I/O RASPBERRY PLC 3.3V pins

The RASPBERRY PLC FAMILY has some of the Raspberry PI 4 board pins available. These pins can be programmed according to Raspberry features such as I/O's operating at 3.3V or any additional features present in the pins (for example I2C communication in pins SCL and SDA). As these pins are directly connected to the Raspberry PI 4 board, they are not as well protected as the normal inputs. These pins are mainly meant to be used as prototyping.

The Raspberry board available pins are summarised in the table below. In order to access these pins some extra considerations must be taken in consideration.

PLC terminal	Raspberry pin
TX	GPIO 14
RX	GPIO 15
MISO	GPIO 9
MOSI	GPIO 10
SCK	GPIO 11
GPIO8	GPIO 8

*IMPORTANT: Do not connect the terminals in the chart above to voltages higher than 3.3V. These terminals provide direct access to the Raspberry board.

There are some special conditions depending on these 3.3V. Now it is going to be shown the considerations to operate with these pins.

10.1 Serial – RX/TX

The Serial protocol can work also as a 3.3V pin. These pins should be used only in case that all the 3.3V pins are already performing a function. If using both interfaces at the same time the Raspberry board will get blocked.

These pins are not established with a pull-up or a pull-down configuration by default. The Arduino board allows the pins to be set in a pull-up configuration. Otherwise, an external pull-up or pull-down circuit could be set.

10.2 SPI0 – MISO/MOSI/SCK

These pins can only work as 3.3V pins if the RS-485 protocol is not going to be used. As the RS-485 protocol uses the SPI to communicate with the Raspberry board, both behaviours cannot happen at the same time as the RS-485 would not work.

These pins are not established with a pull-up or a pull-down configuration by default. The Raspberry board allows the pins to be set either in a pull-up or pull-down configuration. Otherwise, an external pull-up or pull-down circuit could be set.

10.3 GPIO8

GPIO8 is a Raspberry PI 4 GPIO pin that can be set as an input or output. It does not have any default function, so it can be configured without any restrictions. The pin is powered at 3.3 volts; a higher voltage might be dangerous for the device.

This pin is not established with a pull-up or a pull-down configuration by default. If the pin must be used, it might require a pull-up or pull-down configuration for prevention. The Raspberry Pi 4 allows the pin to be set in both configurations; however, an external pull-up or pull-down circuit can be set in order to correctly work with this pin.

11 Digital inputs threshold detection

The Raspberry PLC inputs have a minimum voltage threshold to reliably detect the signal when used as digital input. The threshold value is different depending on the input type:

Input type	Threshold voltage (V)
Digital input	4.3
Analog input	4.1

12 Pinout time response

12.1 Inputs

C language:

Input type	Sampling frequency - Average	Sampling frequency - Highest read time
Digital Interrupt	943.40 kHz (1.06 μs)	9.39 kHz (106.46 μs)
Digital Opto-isolated	2.34 kHz(427.84 μs)	710.01 Hz (1409.05 μs)
Digital Non Opto-isolated	1.63 kHz (615.29 μs)	524.66 Hz (1905.69 μs)
Analog	1.62 kHz (618.17 μs)	624.11 Hz (1602.18 μs)
Direct pins	8.33 MHz (0.12 μs)	18.35 kHz (54.50 μs)

Python language:

Input type	Sampling frequency - Average	Sampling frequency - Highest read time
Digital Interrupt	374.53 kHz (2.67 μs)	12.83 kHz (77.94 μs)
Digital Opto-isolated	2.29 kHz (437.28 μs)	1.21 kHz (827.77 μs)
Digital Non Opto-isolated	1.59 kHz (628.29 μs)	0.63 kHz (1598.32 μs)
Analog	1.59 kHz (628.59 μs)	0.62 kHz (1603.84 μs)
Direct pins	20.95 kHz (47.73 μs)	17.59 kHz (56.85 μs)

12.2 Outputs

C language:

Output type	Maximum stable frequency	Signal Frequency - Maximum	Rise time	Fall time
Digital (QX.0-7)	46.75 Hz (51.8 %)	372.5 Hz (70.1 %)	8.48 μs	138.2 μs
Analog (AX.5-7)	499.7 mHz (49.5 %)	998.5 mHz (49.2 %)	159 ms	228.5 ms
Direct pins	61.08 kHz (50.0 %)	61.08 kHz (50.0 %)	692.9 ns	692.9 ns

Python language:

Output type	Maximum stable frequency	Signal Frequency - Maximum	Rise time	Fall time
Digital (QX.0-7)	46.59 Hz (51.6 %)	237.9 Hz (88.5%)	10.2 μs	99.59 μs
Analog (AX.5-7)	499.7 mHz (49.5 %)	998.5 mHz (49.2 %)	159 ms	228.5 ms
Direct pins	36.81 kHz (49.5%)	36.81 kHz (49.5%)	692.9 ns	692.9 ns

13 Equivalence Table

13.1 Pin-Out

PLC Pinout	RPI Pinout			RPI Pinout	PLC Pinout
-	3.3 V	**1**	**2**	5 V	-
SDA	GPIO2	**3**	**4**	5 V	-
SCL	GPIO3	**5**	**6**	GND	-
B3_INT1 (INT 31)[5] / INT ExpBoard 1	GPIO4	**7**	**8**	GPIO14	TX
-	GND	**9**	**10**	GPIO15	RX
CS SPI1 UART	GPIO17	**11**	**12**	GPIO18	CS SPI1 485
OPTO OUT 2 / B2_INT0[5]	GPIO27	**13**	**14**	GND	-
IRQ SPI1 485	GPIO22	**15**	**16**	GPIO23	UPS Control
-	3.3 V	**17**	**18**	GPIO24	UPS Control
MOSI 0	GPIO10	**19**	**20**	GND	-
MISO 0	GPIO9	**21**	**22**	GPIO25	-
SCLK 0	GPIO11	**23**	**24**	GPIO8	GPIO8 / INT UART ExpBoards / CS ExpBoard 2
-	GND	**25**	**26**	GPIO7	CS SPI0 ETH
-	GPIO0	**27**	**28**	GPIO1	-
OPTO IN 2 / B2_INT1[5]	GPIO5	**29**	**30**	GND	-
IRQ SPI0 ETH	GPIO6	**31**	**32**	GPIO12	OPTO IN 1 / B1_INT1[5]
OPTO OUT 1 / B1_INT0[5]	GPIO13	**33**	**34**	GND	-
MISO 1	GPIO19	**35**	**36**	GPIO16	CS ExpBoard 1
PRELAY / B3_INT0[5]	GPIO26	**37**	**38**	GPIO20	MOSI 1
-	GND	**39**	**40**	GPIO21	SCLK 1

---

[5] See the Interrupt Equivalence Table in Section 8.5.

13.1.1   Analog/Digital Devices Analog I/Os

Analog Inputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
I0.7	0x08	6
I0.8	0x08	0
I0.9	0x0A	6
I0.10	0x08	4
I0.11	0x0A	4
I0.12	0x08	2
Zone B		
I1.7	0x08	7
I1.8	0x08	5
I1.9	0x0A	7
I1.10	0x0A	1
I1.11	0x08	3
I1.12	0x0A	0
Zone C		
I2.7	0x08	1
I2.8	0x0A	3
I2.9	0x0A	2
I2.10	0x28	6
I2.11	0x0A	5
I2.12	0x28	7

Analog Outputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
A0.5	0x40	7
A0.6	0x40	2
A0.7	0x40	0
Zone B		
A1.5	0x40	8
A1.6	0x40	5
A1.7	0x40	1
Zone C		
A2.5	0x41	4
A2.6	0x40	4
A2.7	0x40	3

13.1.2   Analog/Digital Devices Digital I/Os

Digital Inputs			
PLC Pinout	Chip ADDR	Chip INDEX	GPIO
Zone A			
I0.0	0x20	4	-
I0.1	0x20	2	-
I0.2	0x20	3	-
I0.3	0x20	0	-
I0.4	0x20	1	-
I0.5	-	-	GPIO = 13
I0.6	-	-	GPIO = 12
Zone B			
I1.0	0x21	0	-
I1.1	0x20	6	-
I1.2	0x21	1	-
I1.3	0x20	5	-
I1.4	0x21	2	-
I1.5	-	-	GPIO = 27
I1.6	-	-	GPIO = 5
Zone C			
I2.0	0x21	7	-
I2.1	0x21	4	-
I2.2	0x21	6	-
I2.3	0x21	3	-
I2.4	0x21	5	-
I2.5	-	-	GPIO = 26
I2.6	-	-	GPIO = 4

Digital Outputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
Q0.0	0x40	C
Q0.1	0x40	B
Q0.2	0x40	F
Q0.3	0x40	E
Q0.4	0x40	6
Q0.5	0x40	7
Q0.6	0x40	2
Q0.7	0x40	0
Zone B		
Q1.0	0x40	A
Q1.1	0x41	1
Q1.2	0x40	9
Q1.3	0x41	0
Q1.4	0x40	D
Q1.5	0x40	8
Q1.6	0x40	5
Q1.7	0x40	1
Zone C		
Q2.0	0x41	7
Q2.1	0x41	6
Q2.2	0x41	3
Q2.3	0x41	5
Q2.4	0x41	2
Q2.5	0x41	4
Q2.6	0x40	4
Q2.7	0x40	3

13.1.3   Relay Devices Analog I/Os

Analog Inputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
I0.2	0x08	6
I0.3	0x08	0
I0.4	0x0A	6
I0.5	0x08	4
Zone B		
I1.2	0x08	7
I1.3	0x08	5
I1.4	0x0A	7
I1.5	0x0A	1
Zone C		
I2.2	0x08	1
I2.3	0x0A	3
I2.4	0x0A	2
I2.5	0x28	6

Analog Outputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
A0.0	0x40	7
A0.1	0x40	2
A0.2	0x40	0
Zone B		
A1.0	0x40	8
A1.1	0x40	5
A1.2	0x40	1
Zone C		
A2.0	0x41	4
A2.1	0x40	4
A2.2	0x40	3

13.1.4   Relay Devices Digital I/Os

Digital Inputs	
PLC Pinout	GPIO
Zone A	
I0.0	GPIO = 13
I0.1	GPIO = 12
Zone B	
I1.0	GPIO = 27
I1.1	GPIO = 5
Zone C	
I2.0	GPIO = 26
I2.1	GPIO = 4

Digital Outputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
Q0.0	0x40	7
Q0.1	0x40	2
Q0.2	0x40	0
Zone B		
Q1.0	0x40	8
Q1.1	0x40	5
Q1.2	0x40	1
Zone C		
Q2.0	0x41	4
Q2.1	0x40	4
Q2.2	0x40	3

13.1.5 Relay

Digital Outputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
R0.1	0x20	2
R0.2	0x20	4
R0.3	0x20	0
R0.4	0x20	3
R0.5	0x40	6
R0.6	0x40	E
R0.7	0x40	F
R0.8	0x40	B
Zone B		
R1.1	0x20	6
R1.2	0x21	0
R1.3	0x20	5
R1.4	0x21	1
R1.5	0x40	D
R1.6	0x41	0
R1.7	0x40	9
R1.8	0x41	1
Zone C		
R2.1	0x21	4
R2.2	0x21	7
R2.3	0x21	3
R2.4	0x21	6
R2.5	0x41	2
R2.6	0x41	5
R2.7	0x41	3
R2.8	0x41	6

NOTE: After a reboot/power disconnection and reconnection, the UPS will be activated and, until the device is fully initialised again (it will take a period from 1 to 6 seconds, depending on the opened processes that must be closed), the outputs will maintain their last activation state. Once the device is initialised again, all the outputs will turn into low state until they are modified again. It is very important to consider this behaviour for safety reasons.

14 Internal I2C and SPI Connections
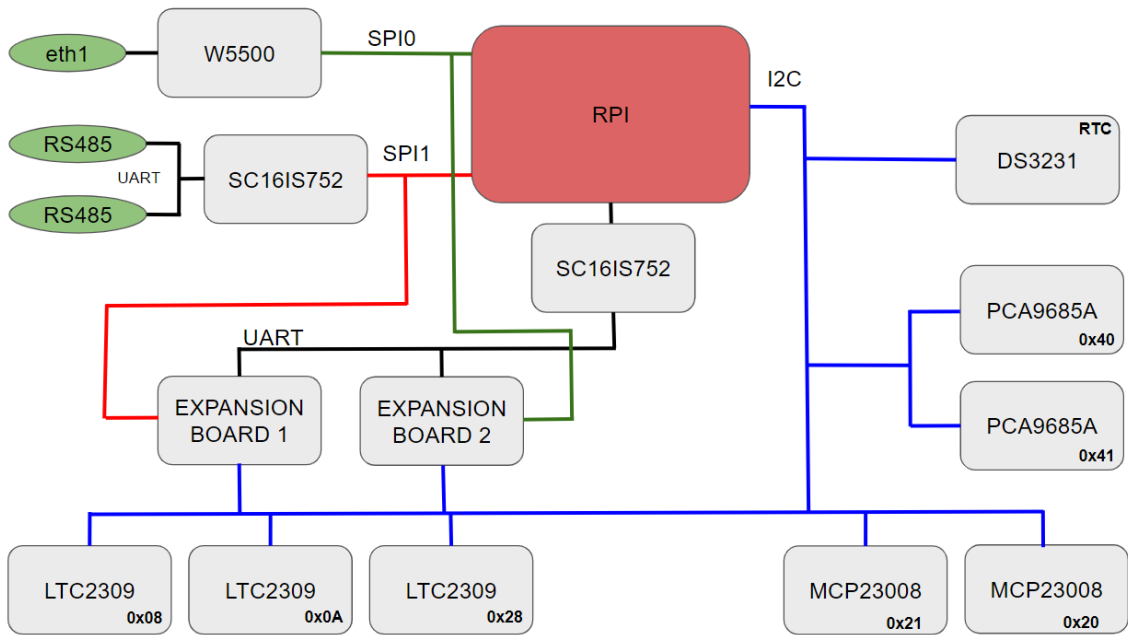
15 Software: Input & Output control

The IO's of the PLC can be managed through 3 different programming languages which are installed by default but, if not or in case of installing a fresh image, they can also be installed without any issue.

15.1 C++ Library

We have a C++ Library called rpiplc-lib that can be found on our GitHub repository. There you can also find the installation steps and the usage explanation among some practical examples.

rpi-plc C++ Library GitHub Repository

15.2 Python Library

We have a Python Library called rpiplc-python3-lib that can be found on our GitHub repository. There you can also find the installation steps and the usage explanation among some practical examples.

rpi-python3-plc Python Library GitHub Repository

15.3 Node-RED node

We have a Node-RED node called node-red-contrib-rpiplc-node that can be found on the Node-RED library repository. There you can also find the installation steps and the usage explanation among some practical examples.

node-red-contrib-rpi-plc-node Library Repository

16 A & B Zone Features: Communications & RTC & uSD

16.1 RS-485

Raspberry PLC is provided with two RS-485 channel communications. Both are half-duplex and are controlled via the SPI0 bus.

Both RS-485 ports can be configured with the following commands:

- Configure the RS-485(2) port using the command

```
stty (baudrate) -F /dev/ttySC2 raw -echo
```

- Configure the RS-485(1) port using the command:

```
stty (baudrate) -F /dev/ttySC3 raw -echo
```

These are the baud rates accepted for both ports:

4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200

Now that we can configure both channels we can start using these ports with the desired programming system.

For example, we can use the following bash script instruction to see the data coming from the RS-485(2). It will stay open waiting for data to read. Cancel it any time pressing Ctrl + C:

```
cat /dev/ttySC2
```

Or also use the following command to send a message through the RS-485(1) port:

```
echo -ne "Test message" > /dev/ttySC3
```

RS485 Noise/Error troubleshooting

If you are experiencing communication errors due to noise, you can add external failsafe circuitry to increase noise margin for applications in noisy environments. To do so, add the following resistors to each of the RS-485 lines:

- A+ : 2.2 kΩ pull up resistor to 5V.
- B- : 2.2 kΩ pull down resistor to GND.

The 5V and GND lines of the PLC can be used for this task.

16.2 I2C

The I2C protocol is meant to work in a pull-up configuration. A pull-up configuration means that when the pin is at rest (nothing connected to it) it always reads a HIGH value. In this case it reads 5V when nothing is connected. The pull-up configuration is established by default in these pins.

If it is meant to work as a GPIO at 5V, it has to be considered that they are pull-up inputs.

16.3 SPI

The Raspberry PLC pins used for the SPI bus are summarised in the table below. The MISO, MOSI and CLOCK pins are the same for all the connected devices, while the SS pin, conversely, will be unique for each one.

Function	PLC connection	Raspberry board pin
MISO 0	SO	GPIO 9
MOSI 0	SI	GPIO 10
CLOCK 0	SCK	GPIO 11
MISO 1	SO	GPIO19
MOSI 1	SI	GPIO20
CLOCK 1	SCK	GPIO21
Reset	Reset	Reset
SS	SCL/SDA/RX/TX	GPIO3/GPIO2/GPIO15/GPIO14

16.4 TTL

Raspberry PLC has one TTL port, RX/TX. TTL is accessed with the function Serial (pins GPIO14 and GPIO15 of the Raspberry PI 4).

16.5 Ethernet

Raspberry PLC has two Ethernet ports, eth1 and eth0. The Ethernet port controller is based on W5500 IC and it generates the eth1 interface, while the eth0 interface is automatically generated by the Raspberry PI 4.

In the RASPBERRY PLC FAMILY, W5500 IC communicates to the Raspberry PI 4 via SPI1 bus.

If you open a terminal window, and exec: **ip addr,** you will see all the network interfaces working on the Raspberry.

By default, we have a virtual interface with the ip address **10.10.10.20** called **eth0:1**. Likewise, eth1 has a virtual interface called **eth1:1** with the ip address **10.10.11.20**.

In case that you are interested in adding another interface or changing these two ip addresses, go to the directory where the interfaces are, with the following command:

```
cd /etc/network/interfaces.d/
```

There you will be able to see and edit each interface configuration.

Finally, in order to check the Ethernet connection of a specific interface, connect the cable of the Ethernet, and send 4 packets doing a ping to an ip address from the DNS server provided by Google (8.8.8.8) with the following instruction:

```
ping -c4 -I (interface name) 8.8.8.8
```

Example:

```
ping -c4 -I eth0 8.8.8.8
```

If the command exited and the result was: "4 packets transmitted, 4 received, 0% packet loss", the connection was successful!

NOTE: The ETH1 port is configured at 10BT Half-Duplex auto-negotiation disabled. 10BT is the maximum speed allowed by the hardware.

16.6 Wi-Fi

Raspberry PLC has Wi-Fi wireless connectivity by default. It uses the two most common frequencies; 2.4GHz and 5GHz. It uses the IEEE 802.11.b/g/n/ac bands.

To connect this PLC with our Wi-Fi network we must modify the wpa_supplicant file inside the Raspberry:

```
> sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

And we must configure the file with the Wi-Fi network configuration parameters (it can change depending on the specific configuration of each case):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB


network={
ssid="NETWORK SSID"
psk="NETWORK PASSWORD"
key_mgmt=WPA-PSK
}
```

16.7 BLE

Raspberry PLC has BLE wireless communication by default. It uses the 5.0 BLE version.

Connecting the Raspberry Pi bluetooth to Android bluetooth is possible following the next instructions:

Pairing mode on your Raspberry Pi PLC:

Type the following instruction:

```
sudo bluetoothctl
```

At the "[bluetooth]#" prompt, enter the following commands:

```
discoverable on
pairable on
agent on
default-agent
scan on
```

Wait for a message to appear showing the Android phone has been found:

```
[NEW] Device 12:23:34:45:56:67 devicename
```

Type "pair" with the mac address of your Android phone:

```
pair 12:23:34:45:56:67
```

On your Android phone and Raspberry Pi, confirm the passcode and then type the following instruction:

```
quit
```

Finally press enter to return to the command line.

Example of connecting mode on your Raspberry Pi BLE using an Android device:

To be able to get data from your Android phone, you must install a text-based modem control and terminal emulator program called minicom following the next instructions:

```
sudo apt update
sudo apt install minicom
```

Once the minicom is installed, you will have to open the serial port using an Android app on your mobile phone: go to the Play Store and install a "Serial Bluetooth Terminal" app.

Once it is installed, open it and go to Devices, click on the Raspberry Pi device, you should see the results on the Minicom terminal.

16.8 RTC

Raspberry PLC RTC Module is based on the DS3231 Chip. This chip works with the I2C protocol communication, so it is required to have enabled the I2C protocol. It is powered by a button battery (CR1216 or CR1220).

There are two types of date-time clocks in a Linux system:

The Hardware Clock: This clock is an independent hardware device, with its own power domain (battery, capacitor, etc), that operates when the machine is powered off, or even unplugged.

The System Clock: This clock is part of the Linux kernel and is driven by a timer interrupt. It has meaning only while Linux is running on the machine. The System Time is the number of seconds since 00:00:00 January 1, 1970, UTC. The System Time is not an integer, though. It has virtually infinite precision.

To read the Hardware Clock and print its time to standard output in the ISO 8601 format, type the following:

```
sudo hwclock -r
```

To set the Hardware Clock to the time given by the --date option, and update the timestamps in /etc/adjtime, exec:

```
sudo hwclock --set --date="2021-04-22 16:25"
```

For more information about "hwclock", exec the following command:

```
man hwclock
```

16.9 uSD

The micro-SD contains all the configurations to work properly with the Raspberry PLC. Consequently, the micro-SD should not be manipulated because without it the Raspberry PLC would not work correctly.

16.10 UPSberry

All Raspberry PLC Family devices include the UPSberry. It is a Plug & Play UPS Smart Shield that provides 5 seconds of electricity in front of a power supply cut off. This extra time of electricity ensures that the Raspberry follows a correct Shut Down process, avoiding any SD corruption issue. The recharge time of the UPS is less than 1 minute, and the reconnection time is less than 20 seconds. To know more technical specs about this device, you can consult the **UPSberry User Manual**.

NOTE: Do not confuse this UPS with the UPS models of other brands, the behaviour is different.

16.10.1 How to execute commands before the Power Off process

The management of the UPS is already handled through the overlays, but this can still be done.

The UPS has its own configuration file **rpishutdown** and service **rpishutdown-service.** The placement of both files can be consulted in the **Raspberry Pi PLC OS configuration** manual.

To execute specific commands before finishing the Power Off process, you must follow the next steps:

- Create the following path:

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

- Inside this folder, create a file called pre-poweroff:

```
sudo touch pre-poweroff
```

The file type can be anyone you want (Bash Scripts, Python, C++, etc.) but the most important thing is that it must be called **pre-poweroff** without any extension (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. Take into account that this code only can be things that do not depend on the external power supply, like creating a file or things directly related with the Raspberry. (Following this rule, you cannot do things such as enabling an output so it depends on the external power supply).

- Give the execution permissions to the file:

```
sudo chmod ugo-x pre-poweroff
```

Once finished these steps, the file will be executed before the Raspberry power off.

17 Additional family features: Expansion Boards

You can customise up to two additional communication expansions on your Raspberry PLC and prepare your custom-made project:

17.1 LTE Cat 1

The LTE Cat 1 expansion board uses the **SARA-R412M** module, which has the following features:

- **Model:** SARA-R412M
- **Type:** 2G GPRS, GSM/LTE
- **Key Features:** GSM Quad-band 850/1900, 900/1800 MHz, Maximum output power -8dBm, IPv4/IPv6, dual-stack
- **Applications:** Remote monitoring automation, asset tracking, surveillance and security, home automation systems, point of sales terminals etc.

In order to work with the Raspberry PLC GPRS module, you must do the following:

1. First of all, ensure that the [librpiplc](#) library is installed, as it is required for the process. If you already have a "test" folder in your home directory, the library should already be installed.

2. Now we must make sure that the config.txt file is set up correctly. To do this, first open /boot/firmware/config.txt with your preferred text editor, in this case we use nano:

   ```
   sudo nano /boot/firmware/config.txt
   ```

3. Use your arrow keys to reach the bottom of the file, and make sure you have these lines set up like this:

   ```
   [all]
   dtparam=spi=on
   gpio=8=pd
   dtoverlay=spi0-2cs,cs0_pin=7,cs1_pin=8
   dtoverlay=w5500,cs=0,int_pin=6,speed=10000000
   dtparam=i2c_arm=on
   dtoverlay=i2c-rtc,ds3231
   dtoverlay=sc16is752-spi1-rpiplc-v4,xtal=14745600
   enable_uart=1
   ```

4. Execute file setup-sara.sh which can be found [here](#), using these commands:

   ```
   wget
   https://apps.industrialshields.com/main/rpi/rpiplc_v6/setup-sara.sh
   chmod 755 setup-sara.sh
   sudo ./setup-sara.sh
   ```

If your SARA-R412M module is installed in the slot 2, you'll need to make adjustments to the setup-sara.sh file. You'd need to replace all occurrences of ttySC0 with ttySC1 and EXP1_RST with EXP1_RST_2.

The script asks for APN credentials and RAT (Radio Access Technology): LTE Cat M1 (4G), NB-IoT (Narrow Band) or GPRS/eGPRS.

It also 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.

About 50-120 seconds after executing setup-sara.sh, interface **ppp0** should be created. The process can be followed with:

```
journalctl -f
```

5. After finishing, you can check that the interface has been created with:

```
ifconfig ppp0
```

To ensure everything has gone flawless, you can ping to 8.8.8.8 like so:

```
ping 8.8.8.8 -c4 -I ppp0
```

6. To stop and disable the service, use the commands:

```
sudo systemctl stop pppd.service
sudo systemctl disable pppd.service
```

7. Finally, try to execute the created script newtork-setup.py with the desired RAT:

```
python /usr/local/bin/network-setup.py gprs | nb-iot | lte-m
```

17.2 LTE Cat 4

The LTE Cat 4 expansion board uses the **LARA-L6** module, which has the following features:

- **Model:** LARA-L6
- **Type:** LTE Cat 4 (FDD/TDD)
- **Key Features:** Multi-band LTE (FDD: 700 to 2600 MHz; TDD: 2300 to 2600 MHz), Maximum output power 23 dBm, downlink speeds up to 150 Mbps, uplink speeds up to 50 Mbps, GNSS support (GPS/GLONASS), IPv4/IPv6, dual-stack
- **Applications:** Industrial automation, smart metering, asset tracking, fleet management, security and surveillance, connected health, smart cities, etc.

In order to work with the Raspberry PLC LTE Cat 4 module, you must do the following:

1. First of all, ensure that the librpiplc library is installed, as it is required for the process. If you already have a "test" folder in your home directory, the library should already be installed.

2. Execute file setup-lara.sh which can be found here, using these commands:

```
wget
https://apps.industrialshields.com/main/rpi/rpiplc_v6/setup-lara.sh
chmod 755 setup-lara.sh
sudo ./setup-lara.sh
```

   If your LARA-L6 module is installed in the slot 2, you'll need to make adjustments to the setup-lara.sh file. You'd need to replace all occurrences of ttySC0 with ttySC1 and EXP1_RST with EXP1_RST_2.

   The script asks for your SIM card APN credentials (APN, ISP username, ISP password). They depend on the phone company of the card, you can find them on the Internet or contact directly the company if you do not have them.

   It also 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.

   About 50-120 seconds after executing setup-lara.sh, interface **ppp0** should be created. The process can be followed with:

```
journalctl -f
```

3. After finishing, you can check that the interface has been created with:

```
ifconfig ppp0
```

   To ensure everything has gone flawless, you can ping to 8.8.8.8 like so:

```
ping 8.8.8.8 -c4 -I ppp0
```

4. If there are issues with the interface, check that the **pppd** and **check-ppp0** services are active:

```
systemctl status pppd.service
systemctl status check-ppp0.service
```

5. If no successful responses are received, manually run the script created by the setup process with:

```
python3 /usr/local/bin/network-setup.py
```

17.3 CAN

The CAN expansion board uses the **MCP2515** microchip, which has the following features:

- **Model:** MCP2515
- **Type:** CAN V2.0B
- **Key Features:** Speed of 1Mb/s, receive buffers, masks and filters, data byte filtering on the first two data bytes, three transmit buffers with prioritization and abort features, high speed SPI interface (10MHz), etc.
- **Applications:** Communication with all kinds of CAN devices and the protocols that can be applied to this communication method.

17.4 LoRa

The LoRa expansion board can use the **RN2483** or **RN2903** microchip depending on the frequency needed, which have the following features:

- **Model:** RN2483 (for Europe/Asia), RN2903 (for North America/Australia)
- **Type:** LoRa
- **Key Features:** On-board LoRaWAN protocol stack, ASCII command interface over UART, Castellated SMT pads for easy and reliable PCB mounting, Environmentally friendly, RoHS compliant, Device Firmware Upgrade (DFU) over UART, etc.
- **Applications:** Automated Meter Reading, Home and Building Automation, Wireless Alarm and Security System, Industrial Monitoring and Control, Machine to Machine (M2M), Internet of Things (IoT), etc.

In order to work with the Raspberry PLC LoRa module, you must do the following:
1. Install our library **rpiplc-lib**, which can be found here.
2. Ensure that the */boot/config.txt* file ends with these lines (Remove the "Mandatory" comments):

```
[all]
```

```
dtparam=spi=on                                    <---- Mandatory
gpio=8=pd
dtoverlay=spi0-1cs,cs0_pin=7
dtoverlay=w5500,cs=0,int_pin=6
dtparam=i2c_arm=on
dtoverlay=i2c-rtc,ds3231
dtoverlay=sc16is752-spi1-rpiplc-v4,xtal=14745600  <---- Mandatory
enable_uart=1                                     <---- Mandatory
```

Otherwise, replace the previous configuration for this one and reset the PLC.

3. Set the reset pin to high to enable the LoRa module depending on the slot used.
    ○ If you installed librpiplc v3.0.0 use this, you'll need the model and version of your RPIPLC to replace [RPIPLC_Version] and [RPIPLC_Model]. You can find all RPIPLC models and verions [here](#).
        ■ For the slot 1, execute:

```
sudo ~/test/[RPIPLC_Version]/[RPIPLC_Model]/set-digital-output EXP1_RST
1
```

        ■ For the slot 2, execute:

```
sudo ~/test/[RPIPLC_Version]/[RPIPLC_Model]/set-digital-output EXP1_RST_2
1
```

    ○ If you are using an older version of librpiplc instead of v3.0.0, you can activate the LoRa module with this:
        ■ For the slot 1, execute:

```
sudo ./rpiplc-lib/test/set-digital-output EXP1_RST 1
```

        ■ For the slot 2, execute:

```
sudo ./rpiplc-lib/test/set-digital-output EXP1_RST_2 1
```

Note: The slot 1 is the closest to the ethernet connector and the slot 2 is the further one.

4. To open the port at 57600 baudrate (it is the module's default baudrate), execute:

```
stty 57600 -F [PORT] raw -echo
```

Note: The parameter PORT is **/dev/ttySC0** for slot 1 and **/dev/ttySC1** for slot 2.

5. To leave the port listening, execute:

```
cat [PORT]
```

6. To send a command, execute:

```
echo -e "[COMMAND]\r" > [PORT]
```

For instance, here are some useful commands:

- ○ **mac pause:** Pause the LoRaWAN stack functionality and allow transceiver
- ○ **radio rx 0**: Listen for data*.*
- ○ **radio tx ABC**: Send a message (ABC).
- ○ **radio set freq [FREQUENCY]**: Set the frequency of the radio transceiver.

You can also find further documentation on LoRa commands by following [this link](#).

17.5 GPS

The GPS expansion board uses the **L80** microchip, which has the following features:

- **Model:** L80-M39
- **Type:** GPS
- **Key Features:** GPS L1 1575.42 MHz C/A Code, 66 search channels, 22 simultaneous tracking channel, Max Update Rate up to 10 Hz, 1 Hz by default, Velocity Accuracy without aid: 0.1 m/s, Acceleration Accuracy without aid: 0.1 m/s², etc.
- **Applications:** High performance of MTK positioning engine to the industrial applications with top level of sensitivity, accuracy and TTFF in the industry with the minimum power consumption.

18 Build a Raspberry PLC Image

18.1 Build a Lite/Desktop Raspberry PLC Image from a Raspberry Pi OS 64-bit version

The Raspberry PLC family of products incorporates as standard a microSD card preloaded with Raspberry Pi OS Lite, a lightweight operating system without a graphical interface, designed for enhanced CPU performance.

Even so, if you want to install another operating system compatible with Raspberry Pi 4B, all the services, scripts and executables must be added to the new image to be able to operate with the additional communications, the I/Os of the PLC as well as the internal UPS, among other functionalities.

To install a new image, follow these steps:

1. Download the Raspberry Pi Imager from this URL: https://www.raspberrypi.org/software/
2. Download an operating system image from Raspberry Pi. You have two options for downloading the OS image:
   a. Download one of our pre-configured images (which already includes the necessary configurations): https://apps.industrialshields.com/main/rpi/images/RPI_PLC/
   b. Download one image from the official repository, with all the operating system images available here: https://www.raspberrypi.org/software/operating-systems/
3. Insert the microSD card into your PC using an adapter, then use the Raspberry Pi Imager to write the downloaded image to the card.
4. Once the image is written, insert the microSD card into your Raspberry PLC and proceed with the setup.

If you did not install one of our pre-configured images, follow these steps to add the required services, scripts, and executables for full functionality:

1. First, download the **install.sh** file by running the following command:

```
wget
https://apps.industrialshields.com/main/rpi/rpiplc_v6/install.sh
```
This bash script will automatize almost all the required processes.

2. Secondly, change the file permissions to make the script executable by running:

```
chmod 755 install.sh
```

3. Finally, run the script with the following command, and wait for it to complete.

```
sudo ./install.sh
```

19 Revision Table

Revision Number	Date	Changes
0	02/09/2024	First implementation
1	05/09/2024	Procedures for operating LTE Cat 1 and Cat 4 have been updated
2	25/09/2024	Added section 3.4 ("Isolation Precautions")
3	14/01/2025	Corrections on section 1.1 ("Reference Table") and section 2.1 ("General Specifications")

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