RASPBERRY PLC FAMILY

RASPBERRY PLC FAMILY

RASPBERRY PLC FAMILY Quick User Guide

Revised, September 2021

Preface

This User Guide is 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 can suffer irreversible damage.
- Maintenance must be performed by qualified personnel familiarized 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 authorized 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 encapsulated, there are supercapacitors if 25F which can be dangerous. Be careful with them.

Table of Contents

INDUSTRIAL SHIELDS

1 General Description RASPBERRY PLC FAMILY Product

1.1 Reference Table

Reference Table				
Model	RAM Memory			Accessory
	2GB RAM	4GB RAM	8GB RAM	with FAN
PLC Raspberry General Family				
Raspberry PLC Ethernet CPU (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000400	012003000000	012004000000	xxxxxxxxxxxxF
Raspberry PLC Ethernet 21 I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000600	012003000200	012004000200	xxxxxxxxxxxxF
Raspberry PLC Ethernet 42 I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000100	012003000400	012004000400	xxxxxxxxxxxxF
Raspberry PLC Ethernet 58 I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000300	012003000600	012004000600	xxxxxxxxxxxxF
Raspberry PLC Ethernet 19R I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000500	012003000100	012004000100	xxxxxxxxxxxxF
Raspberry PLC Ethernet 38R I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000700	012003000300	012004000300	xxxxxxxxxxxxF
Raspberry PLC Ethernet 57R I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000800	012003000500	012004000500	xxxxxxxxxxxxF
Raspberry PLC Ethernet 38AR I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000900	012003000700	012004000700	xxxxxxxxxxxxF
Raspberry PLC Ethernet 53ARR I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002001000	012003000800	012004000800	xxxxxxxxxxxxF
Raspberry PLC Ethernet 57AAR I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002001100	012003000900	012004000900	xxxxxxxxxxxxF
Raspberry PLC Ethernet 54ARA I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002001100	012003001000	012004001000	xxxxxxxxxxxxF
Raspberry PLC Ethernet 50RRA I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	012002000000	012003001100	012004001100	xxxxxxxxxxxxF
PLC Raspberry GPRS Family				
Raspberry PLC Ethernet & GPRS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000000	016003000000	016004000000	xxxxxxxxxxxxF
Raspberry PLC & GPRS 21 I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000200	016003000200	016004000200	xxxxxxxxxxxxF
Raspberry PLC & GPRS 42 I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000400	016003000400	016004000400	xxxxxxxxxxxxF
Raspberry PLC & GPRS 58 I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000600	016003000600	016004000600	xxxxxxxxxxxxF
Raspberry PLC & GPRS 19R I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000100	016003000100	016004000100	xxxxxxxxxxxxF
Raspberry PLC & GPRS 38R I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000300	016003000300	016004000300	xxxxxxxxxxxxF
Raspberry PLC & GPRS 57R I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000500	016003000500	016004000500	xxxxxxxxxxxxF
Raspberry PLC & GPRS 38AR I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000700	016003000700	016004000700	xxxxxxxxxxxxF
Raspberry PLC & GPRS 53ARR I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000800	016003000800	016004000800	xxxxxxxxxxxxF
Raspberry PLC & GPRS 57AAR I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002000900	016003000900	016004000900	xxxxxxxxxxxxF
Raspberry PLC & GPRS 54ARA I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002001000	016003001000	016004001000	xxxxxxxxxxxxF
Raspberry PLC & GPRS 50RRA I/Os Analog/Digital PLUS (Raspberry Pi 4B 2GB RAM Included + 8GB pSLC SIM W/Linux)	016002001100	016003001100	016004001100	xxxxxxxxxxxxF

1.2   Zones Table

Model	Zones Table			
	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	Measures Table			
	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 this nomenclature.

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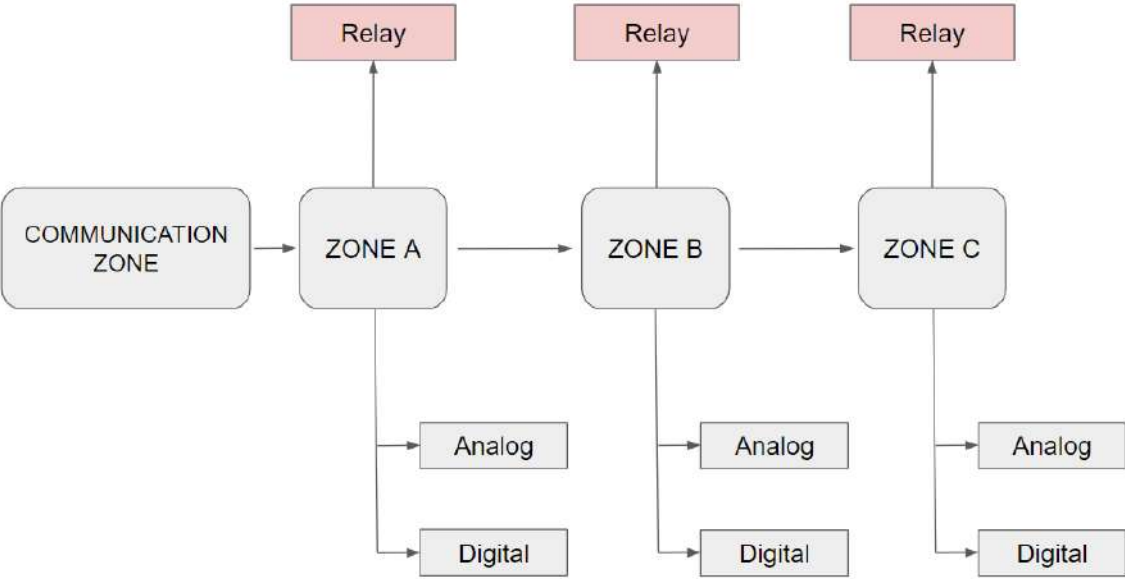
This is the possible zones scheme, whether the zone is Analog/Digital or Relay depends on your device, you can consult it on the Zones Table. The distribution of the different features that provide the Raspberry PLC Family are described below.

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 (1X) CAN (x2) µHDMI
Analog Shield	13 Inputs: 13 Digital inputs, 6 of which can work as Analog Input 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 11 Outputs: 8 Relay Outputs 3 Digital Outputs, 3 of which can work as Analog Output	

1.7   Mechanical dimension

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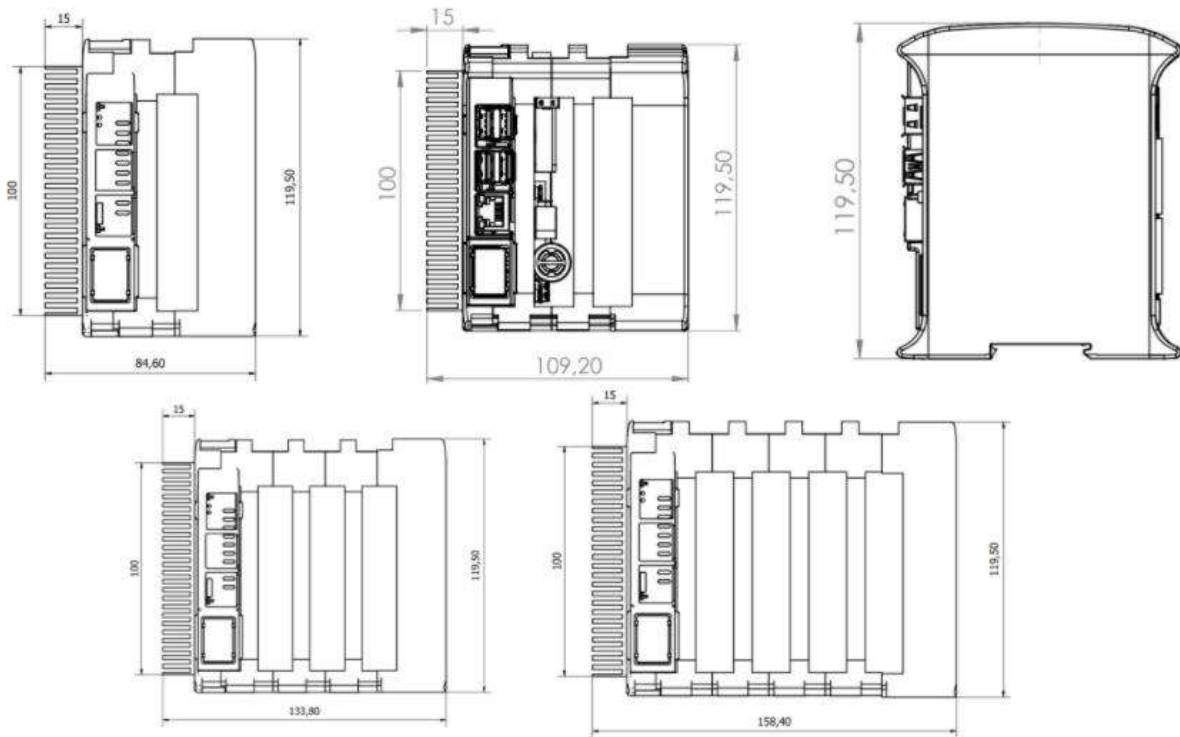
1.8 General Features

CONNECTABLE PLC RASPBERRY 24 VCC		
MODEL TYPE	**Controller General Specifications**	**Information**
Input Voltage	12 to 24 Vdc	Fuse Protection (2.5 A) 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 – CAN - μSD - RTC	MAX485 – W5500
Digital GPIO25 (3.3 V)	3.3 V ---	
An/Dig Input 11bits (0-10Vcc)	0 to 10Vdc Rated Voltage: 10Vdc 5 to 24Vdc I min: 2 to 12 mA Galvanic Isolation Rated Voltage: 24 Vdc ---	
Digital Isolated Input (24Vcc)	5 to 24Vdc I min: 2 to 12 mA Galvanic Isolation Rated Voltage: 24 Vdc ---	
* Interrupt isolated Input HS (24Vcc)	5 to 24Vdc I min: 2 to 12 mA Galvanic Isolation Rated Voltage: 24Vdc ---	
Analog Output 12 bits (0 - 10 Vcc)	0 to 10 Vdc I max: 20 mA Separated PCB Ground ---	
Digital Isolated Output (24 Vcc)	5 to 24 Vdc I max: 70 mA Galvanic Isolation Diode Protected for Relay ---	I max 24 Vdc: 410 mA
Relay Output (220 Vac)	24 Vdc / 220 Vac I max: 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 ▬ ▬ ▬	14
Expandability	I2C: 127 elements – Serial Port RS485	
* By using this type of signal can no longer use Digital signal (24Vdc)		

2 Technical Specifications:

2.1 General Specifications:
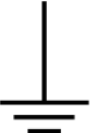
Item		RASPBERRY PLC
Power supply voltage	**DC power supply**	12 to 24Vdc ▬ ▬ ▬
Operating voltage range	**DC power supply**	11.4 to 25.4Vdc ▬ ▬ ▬
Power consumption	**DC power supply**	30 W max.
External power supply	**Power supply voltage**	24 Vdc ▬ ▬ ▬
	Power supply output capacity	700 Ma
Insulation resistance		20MΩ min.at 500Vdc between the AC terminals and the protective earth terminal.
Dielectric strength		2.300 VAC at 50/60 Hz for one minute with a leakage current of 10mA max. Between all the external AC terminals and the protective ground terminal.
Shock resistance		80m/s^2 in the X, Y and Z direction 2 times each.
Ambient temperature (operating)		0º to 50ºC with Raspberry OS Lite / 0º to 40º with Raspberry OS Desktop
Ambient humidity (operating)		10% to 90% (no condensation)
Ambient environment (operating)		With no corrosive gas
Ambient temperature (storage)		-20º to 60ºC
Power supply holding time		2ms 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 : Python, C++, etc.
Program capacity (SRAM)	2 – 4 - 8 GB
EEPROM	4 MB/512 KB
Clock Speed	1.5 GHz
CPU	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

2.3 Symbology

Table that includes all the symbology that is 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.
	Medical Devices Directive 93/42/EEC	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.

4   How to connect PLC to power supply

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

- 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.
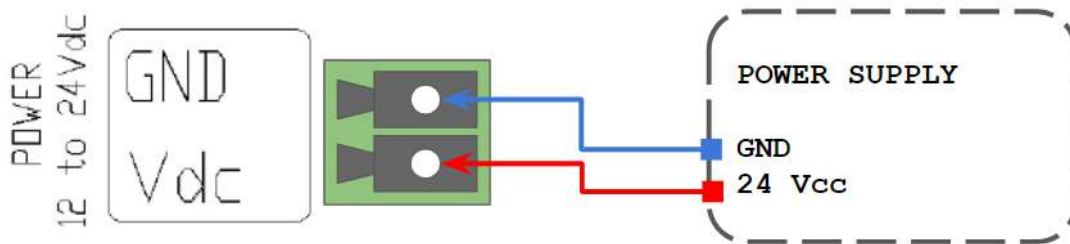
- Suggested power suppliers

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.
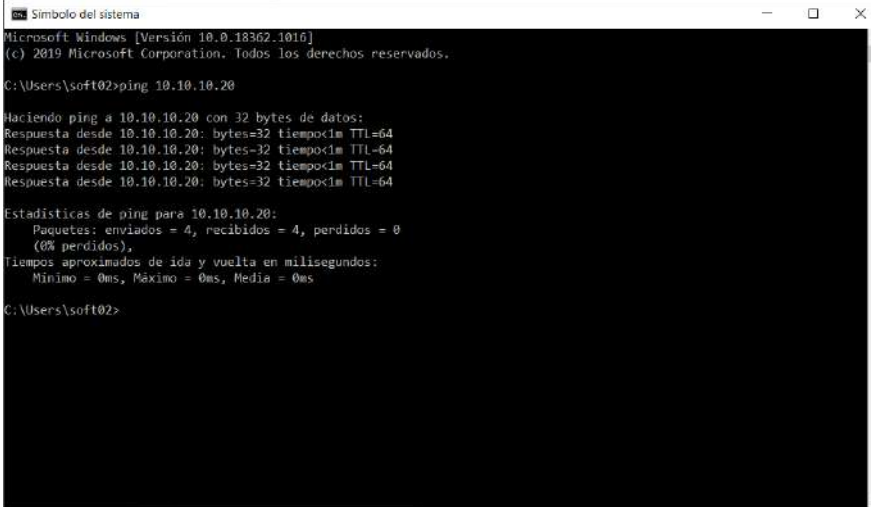
⚠️ **Warning:**

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 it. 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***, the default user is **pi** and the password is **raspberry**. For connecting to it, you must change your local address for being 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 in 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 raspberrian 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.

For accessing to the Raspberry PLC we will run the command *ssh* shown below:

```
$ssh pi@10.10.10.20
```

The connection will start, but first the password provided before will be required. If the username and password are correct, you can start the SSH session.

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, will be a 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: it is used to list the contents of a file on the standard output.
- mkdir: use mkdir command to make 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 closing the connection, just type *exit* in the Raspberry terminal.

5.1.2 **Windows**

For the 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

For connecting to it, you must change your local address for being in the same local network as the Raspberry. You can use the *ping* command with the address 10.10.10.20 as 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 the option *Session.*
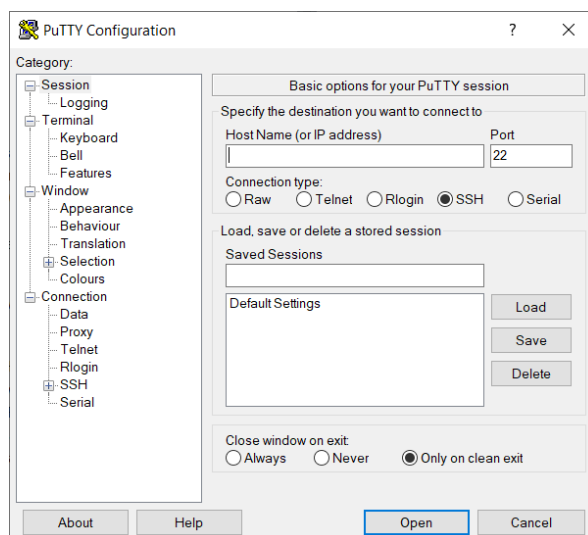
For setting up PuTTY the Raspberry IP must be entered on the *HostName* field and make sure that the port 22 is being used and the *SSH* option is selected on the *Connection Type*. After all the settings have been done, click on the *Open* button to run the program. For a faster start in subsequent sessions, before run 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).
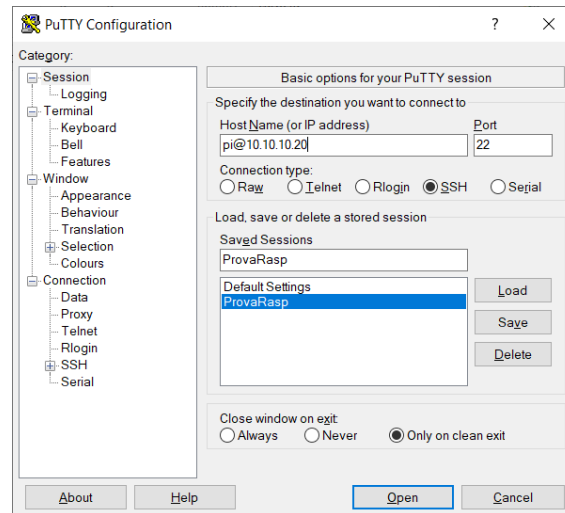
When you start the connection, the terminal window will be opened. Enter your username **pi** and press *Enter*, then enter your password ***raspberry***. If the username and password are correct, you can start the SSH session.

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, will be a 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: it is used to list the contents of a file on the standard output.
- mkdir: use mkdir command to make a new directory
- rm: is used to delete directories and the contents within them.
- touch: allows you to create a blank new file.

For closing the connection, just type ***exit*** in the Raspberry terminal.

6 How to change the IP

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

```
$ip a
```

All the PC's interfaces will be shown and we will have to look for the one named *enp* as will be the Ethernet one. In order to change it, we will use the "**ifconfig**" program, install if not by default in the system. 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 have to be called being administrator, to be able to perform the command below, we need to use su or sudo command.

```
$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 it. In the Control Panel, you can 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.
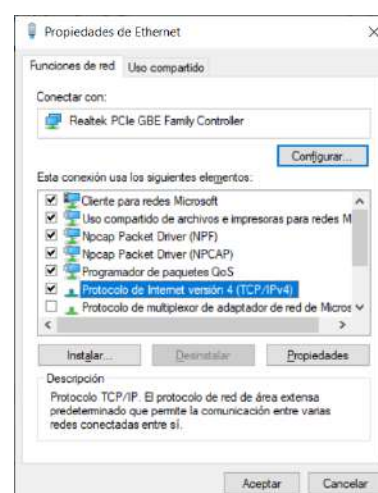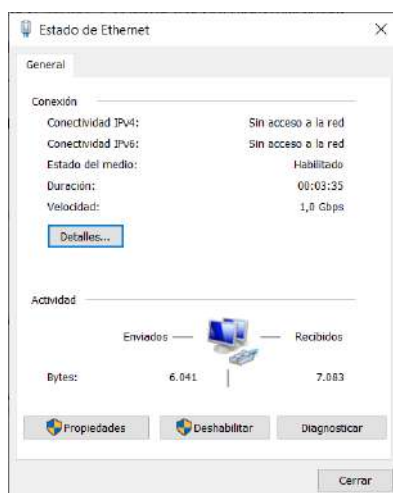
Finally, a window will be displayed where we will be able to change the IP address. Choose the second option and enter the following IP as the following image. Then choose **Accept** button and in the **Proprieties of Ethernet** also. Finally close the window and the IP will have been changed. In order to check it, click on the network details or run the **ipconfig** 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 built-in SCP, allowing to copy files using the scp command. However, Windows systems need to use an external app for being able to use this protocol. We recommend working with WinSCP.

Some directories of the Raspberry Pi PLC have administrator permits, which will not allow to transfer file 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 be opened for logging 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.

On this example, we will be using the default options for the protocol (SFTP) and the port (22). The Raspberry Pi PLC IP by default is the **10.10.10.20**, the user is **pi** and the password **raspberry**. After setting all the requirements, press the connect button and the connection will start.
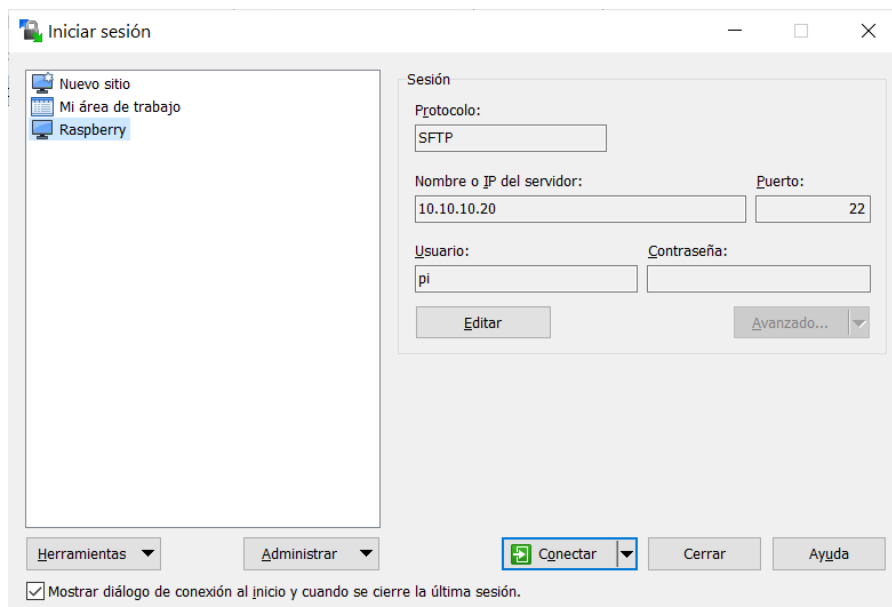
A new window will be opened, seeing the PC files on the left and the Raspberry Pi PLC on the right of the screen. There you can search on the different directories and files and for coping one file from one host no another, just click and drag from one place to the other.
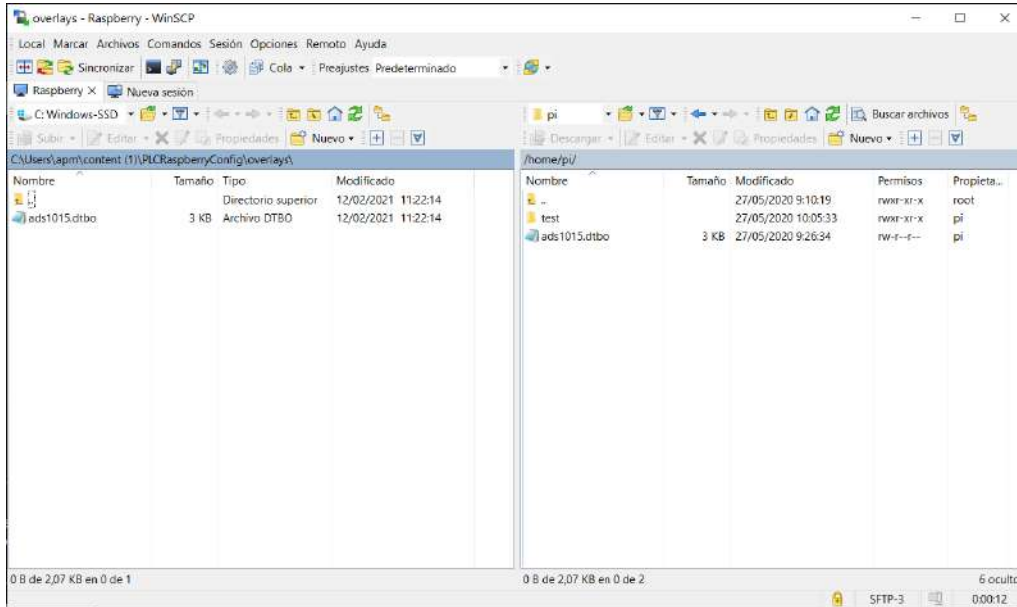
7.2   Linux

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

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

To copy a file from our personal computer to the home directory of the Raspberry Pi PLC we will 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	01200X000000	0	1	0	0	0
21+	01200X000200	6	7	5	3	0
42+	01200X000400	12	14	10	6	0
58+	01200X000600	16	21	15	9	0
19R	01200X000100	4	2	0	3	8
38R	01200X000300	8	4	0	6	16
57R	01200X000500	12	6	0	9	24
38AR	01200X000700	10	9	5	6	8
53ARR	01200X000800	14	11	5	9	16
57AAR	01200X000900	16	16	10	9	8
54ARA	01200X001000	16	16	10	9	8
50RRA	01200X001100	14	11	5	9	16

8.2   0 Zone connection (Communications)

Base (common unit)		
0 Zone		
PLC Connector	*Raspberry Pin*	Function
A+	-	RS485(2)
B-	-	RS485(2)
GND	-	GND
B-	-	RS485(1)
A+	-	RS485(2)
GND	-	GND
SDA	*GPIO2*	I2C
SCL	*GPIO3*	I2C
5V	-	Power Supply
3.3V	-	Power Supply
GND	-	GND
GPIO25	*GPIO25*	RASPBERRY Pin
5V	-	Power Supply
GND	-	GND
SCK	*GPIO11*	SPI/SS
SI	*GPIO10*	SPI/SS
SO	*GPIO9*	SPI/SS
NC	-	Not Connected
RX	*GPIO15*	RX
TX	*GPIO14*	TX
Vdc	-	Power Supply
GND	-	GND

Communication Pins

Power supply connectors
(24Vdc – GND)
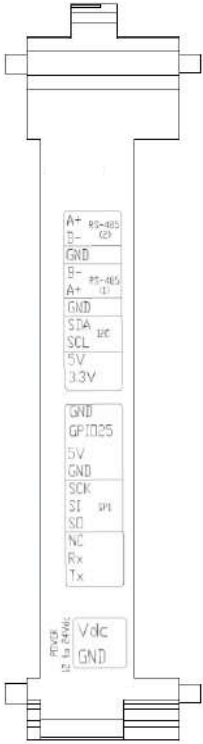
SS: Chip Select pins. These pins can act as TTL, so they can work for the Chip Select pin of any device.

Right Side

ETHERNET 0

USB 3.0 (x2)

USB (x2)

Left Side

Micro SD

Upper Side

POWER INDICATOR

RTC BATTERY

CAN BUS

ETHERNET 1

Type-C (Power Only)

HDMI 0

HDMI 1

AUX
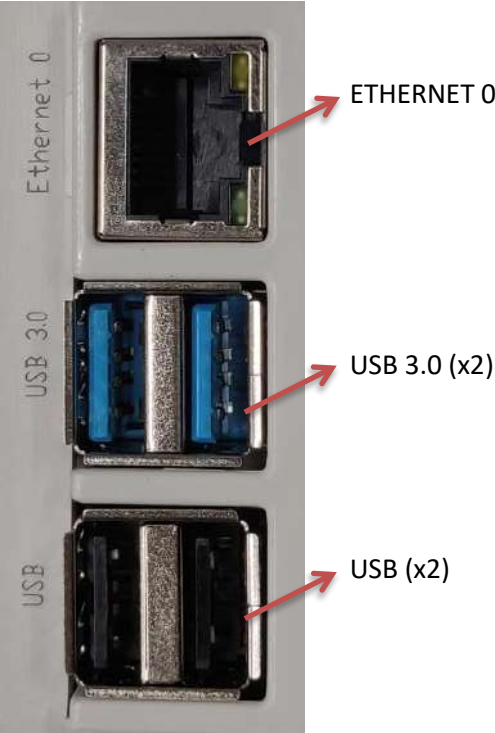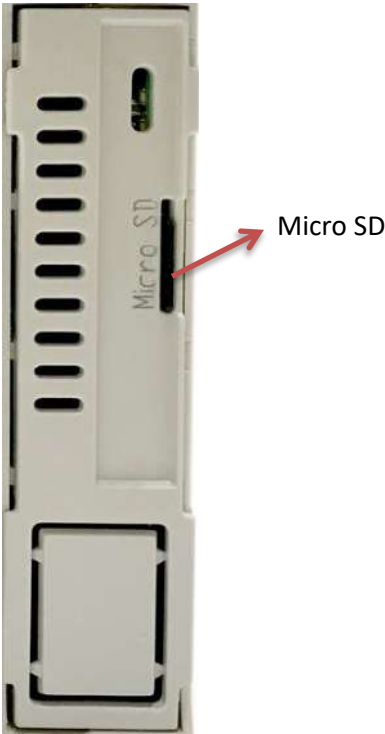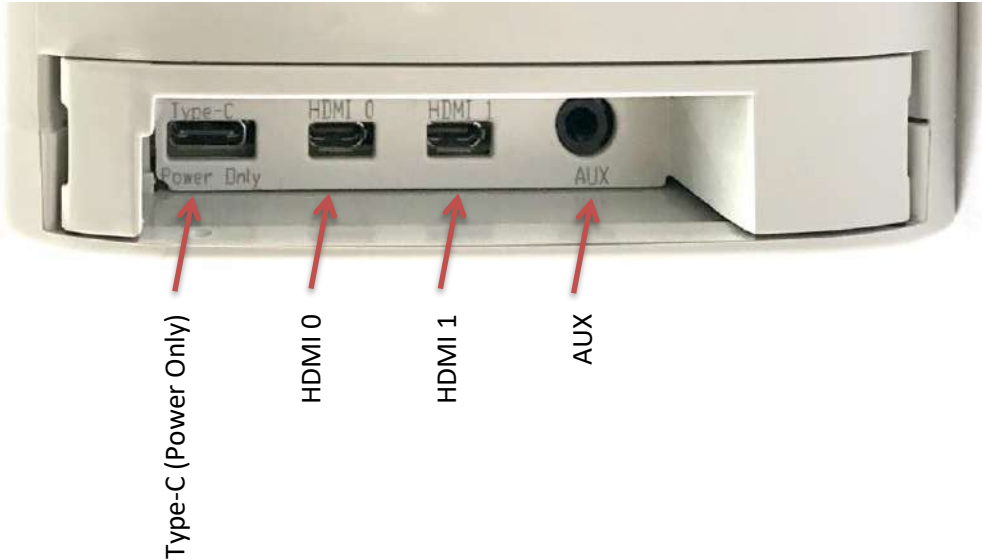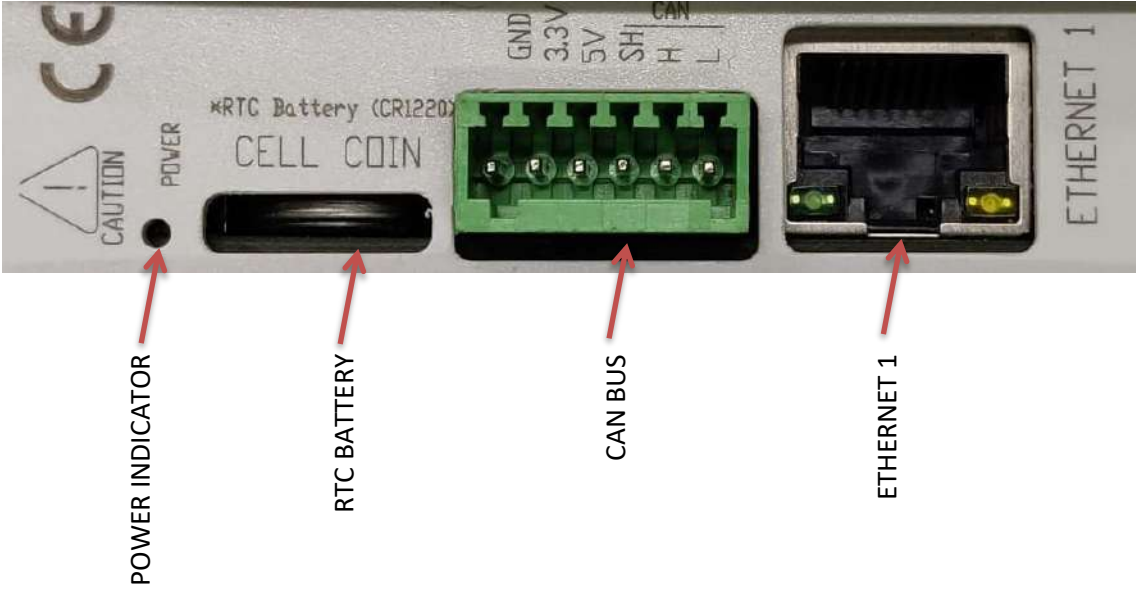
8.3   Analog/Digital Zone connection (I/Os)

X Zone	
Raspberry PLC Connector	Function
IX.12	Analog / Digital In
IX.11	Analog / Digital In
IX.10	Analog / Digital In
IX.9	Analog / Digita 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
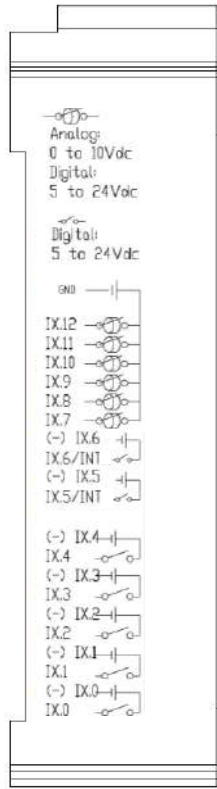
Analog / Digital Inputs

Digital / Interrupt Inputs (isolated)

Digital Inputs (isolated)

Top Zone
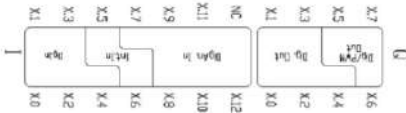
Led indicator I/Os state

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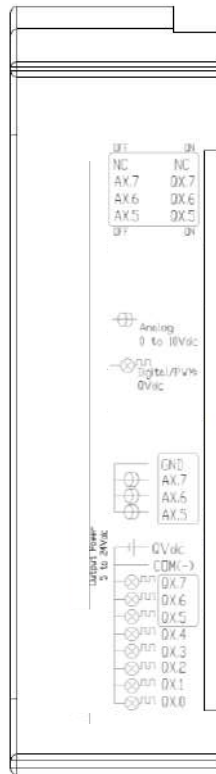
Configuration Switch
(See Section 8 to select the correct configuration for outputs)

Analog Outputs

Voltage Supply/Reference for

Digital/PWM Outputs (isolated)

[1] See the Interrupt Equivalence Table in Section 7.4.
[2] See Section 8 to select the correct configuration for outputs.

8.4   Relay Zone connection (I/Os)

X Zone	
M-Duino 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
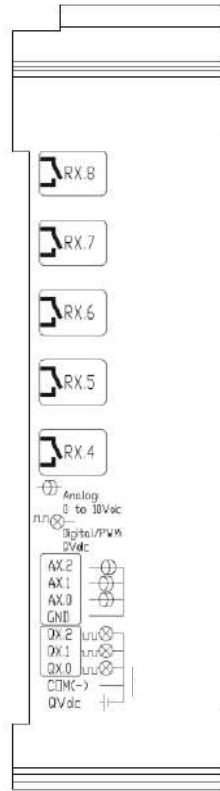
Top Zone

Led indicator I/Os state

C Zone	
M-Duino 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)

---
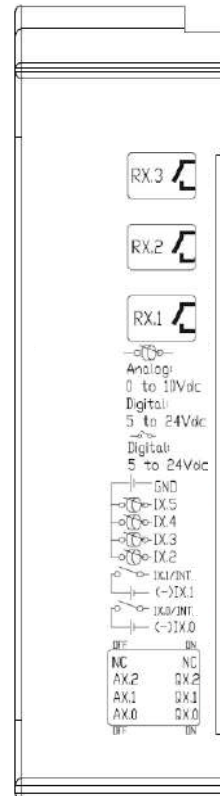
[3] See Section 8 to select the correct configuration for outputs.
[4] See the Interrupt Equivalence Table in Section 7.4.

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	GPIO4
C	2.5	2.0	GPIO17
C	2.6	2.1	GPIO16

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
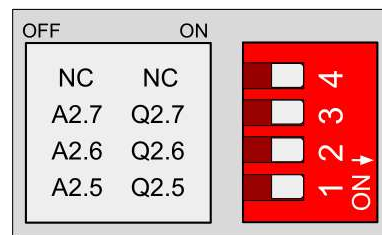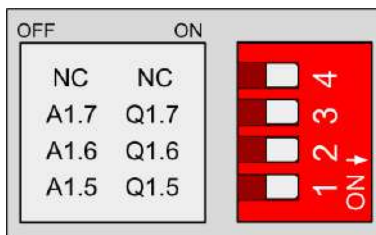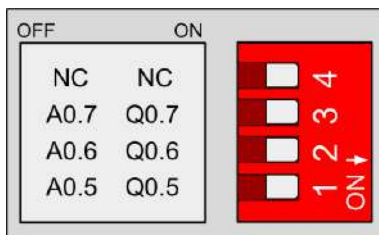
9.2   Relay Zone

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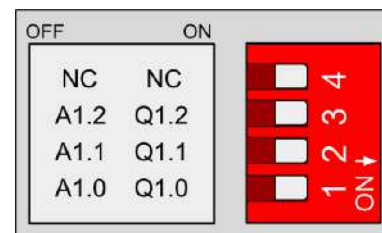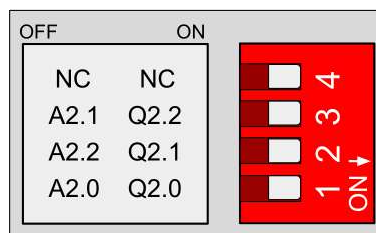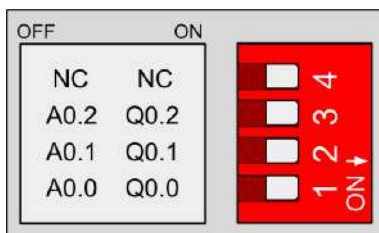
For the Analog Shield if a switch is set to ON, it can only act as Digital Output. If it is set to OFF, it can only act as an Analog Output.

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 summarized in the table below. In order to access to these pins some extra considerations must be taken in consideration.

PLC terminal	Raspberry pin
TX	GPIO 14
RX	GPIO 15
MISO 0	GPIO 9
MOSI 0	GPIO 10
SCLK 0	GPIO 11
GPIO25	GPIO 25

*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 a 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 GPIO25

GPIO25 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 has to 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, it can be set an external pull-up or pull-down circuit in order to correctly work with this pin.

11 Equivalence Table

11.1 Pin-Out

Raspberry Pinout	PLC Pinout
NC	-
GPIO2	SDA
GPIO3	SCL
GPIO4	INT21
GND	-
GPIO17	INT30
GPIO27	INT20
GPIO22	IRQ SPI 485
NC	-
GPIO10	MOSI 0
GPIO9	MISO 0
GPIO11	SCLK 0
GND	-
GPIO 0	-
GPIO5	IRQ SPI CAN
GPIO6	IRQ SPI ETH
GPIO13	INT10
GPIO19	MISO 1
GPIO26	FAN CONTROL
GND	-

Raspberry Pinout	PLC Pinout
5V	-
5V	-
GND	-
GPIO14	TX
GPIO15	RX
GPIO18	CS SPI1 485
GND	-
GPIO23	UPS CONTROL FROM RASPI
GPIO24	UPS CONTROL TO RASPI
GND	-
GPIO25	GPIO25
GPIO8	CS SPI0 CAN
GPIO7	CS SPI0 ETH
GPIO1	-
GND	-
GPIO12	INT11
GND	-
GPIO16	INT31
GPIO 20	MOSI 1
GPIO21	SCLK 1

11.1.1 Analog/Digital Devices Analog I/Os

Analog Inputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
I0.7	0x4a	0
I0.8	0x4a	1
I0.9	0x4b	0
I0.10	0x48	2
I0.11	0x48	0
I0.12	0x48	1
Zone B		
I1.7	0x49	0
I1.8	0x4a	3
I1.9	0x4b	2
I1.10	0x4b	3
I1.11	0x4a	2
I1.12	0x49	1
Zone C		
I2.7	0x49	3
I2.8	0x49	2
I2.9	0x48	3
I2.10	0x4b	1

Analog Outputs		
PLC Pinout	Chip ADDR	Chip INDEX
Zone A		
A0.5	0x40	10
A0.6	0x40	1
A0.7	0x40	0
Zone B		
A1.5	0x40	3
A1.6	0x40	5
A1.7	0x40	8
Zone C		
A2.5	0x41	2
A2.6	0x41	1
A2.7	0x41	0

11.1.2 Digital I/Os

Digital Inputs			
PLC Pinout	**Chip ADDR**	**Chip INDEX**	**GPIO**
Zone A			
I0.0	ADDR = 0x21	5	-
I0.1	ADDR = 0x21	3	-
I0.2	ADDR = 0x21	2	-
I0.3	ADDR = 0x21	1	-
I0.4	ADDR = 0x21	0	-
I0.5	-	-	GPIO = 13
I0.6	-	-	GPIO = 12
Zone B			
I1.0	ADDR = 0x20	2	-
I1.1	ADDR = 0x20	1	-
I1.2	ADDR = 0x20	0	-
I1.3	ADDR = 0x21	7	-
I1.4	ADDR = 0x21	6	-
I1.5	-	-	GPIO = 27
I1.6	-	-	GPIO = 4
Zone C			
I2.0	ADDR = 0x20	6	-
I2.1	ADDR = 0x20	5	-
I2.2	ADDR = 0x20	7	-
I2.3	ADDR = 0x20	4	-
I2.4	ADDR = 0x20	3	-
I2.5	-	-	GPIO = 17
I2.6	-	-	GPIO = 16

Digital Outputs		
PLC Pinout	**Chip ADDR**	**Chip INDEX**
Zone A		
Q0.0	0x40	15
Q0.1	0x40	14
Q0.2	0x40	13
Q0.3	0x40	12
Q0.4	0x40	11
Q0.5	0x40	10
Q0.6	0x40	1
Q0.7	0x40	0
Zone B		
Q1.0	0x40	2
Q1.1	0x40	9
Q1.2	0x40	6
Q1.3	0x40	4
Q1.4	0x40	7
Q1.5	0x40	3
Q1.6	0x40	5
Q1.7	0x40	8
Zone C		
Q2.0	0x41	6
Q2.1	0x41	7
Q2.2	0x41	5
Q2.3	0x41	4
Q2.4	0x41	3
Q2.5	0x41	2
Q2.6	0x41	1
Q2.7	0x41	0

11.1.3 Relay Devices Analog I/Os

Analog Inputs		
PLC Pinout	**Chip ADDR**	**Chip INDEX**
Zone A		
I0.2	0x4a	0
I0.3	0x4a	1
I0.4	0x4b	0
I0.5	0x48	2
Zone B		
I1.2	0x49	0
I1.3	0x4a	3
I1.4	0x4b	2
I1.5	0x4b	3
Zone C		
I2.2	0x49	3
I2.3	0x49	2
I2.4	0x48	3
I2.5	0x4b	1

Analog Outputs		
PLC Pinout	**Chip ADDR**	**Chip INDEX**
Zone A		
A0.0	0x40	10
A0.1	0x40	1
A0.2	0x40	0
Zone B		
A1.0	0x40	3
A1.1	0x40	5
A1.2	0x40	8
Zone C		
A2.0	0x41	2
A2.1	0x41	1
A2.2	0x41	0

11.1.4 Digital I/Os

Digital Inputs	
PLC Pinout	**GPIO**
Zone A	
I0.0	13
I0.1	12
Zone B	
I1.0	27
I1.1	4
Zone C	
I2.0	17
I2.1	16

Digital Outputs		
PLC Pinout	**Chip ADDR**	**Chip INDEX**
Zone A		
Q0.0	0x40	10
Q0.1	0x40	1
Q0.2	0x40	0
Zone B		
Q1.0	0x40	3
Q1.1	0x40	5
Q1.2	0x40	8
Zone C		
Q2.0	0x41	2
Q2.1	0x41	1
Q2.2	0x41	0

11.1.5 Relay

Relay		
PLC Pinout	**Chip ADDR**	**Chip INDEX**
Zone A		
R0.1	0x21	3
R0.2	0x21	5
R0.3	0x21	1
R0.4	0x21	2
R0.5	0x40	11
R0.6	0x40	12
R0.7	0x40	13
R0.8	0x40	14
Zone B		
R1.1	0x20	1
R1.2	0x20	2
R1.3	0x21	7
R1.4	0x20	0
R1.5	0x40	7
R1.6	0x40	4
R1.7	0x40	6
R1.8	0x40	9
Zone C		
R2.1	0x20	5
R2.2	0x20	6
R2.3	0x20	4
R2.4	0x20	7
R2.5	0x41	3
R2.6	0x41	4
R2.7	0x41	5
R2.8	0x41	7

12 Internal I2C and SPI Connections
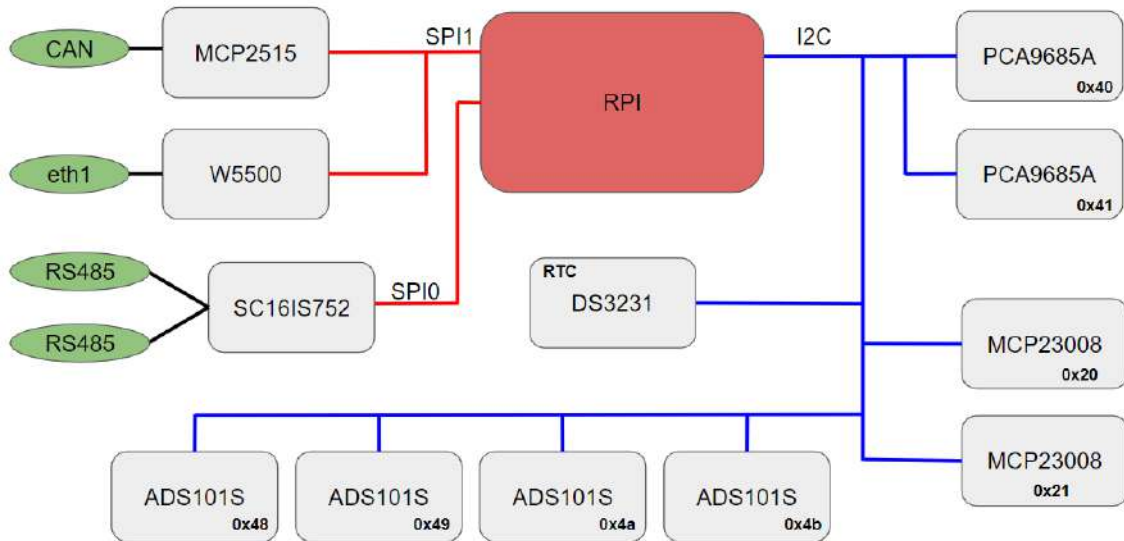
13 Input & Output control

For the Input & Output control of the device there are two commands for working with both digital and analogic pins. These functions are for setting the value or getting its actual information.

13.1 Scripts Location

All the inputs and outputs scripts must be executed from the correct path. It depends on the shield type of the I/O executed. To be sure of your device's shields, you can consult the section 1.2 Zones Table to ensure it. For example, the Raspberry PLC 57 AAR has the first shield Analog/Digital, the second Analog/Digital and the third Relay. In function of the shield of the I/O that you need to activate, you must execute the scripts from a specific path:

13.1.1 Analog/Digital Shield

cd /home/pi/test/analog

13.1.2 Relay Shield

cd /home/pi/test/relay

13.1.3 Generic Shield

cd /home/pi/test/generic

13.2 Set Value

The *set* function will initialize the pin. We will provide the pin with which we are going to work and the value that will be set. If we are working as the **analogical option**, the value will work on a 12-bit resolution range, returning a value from 0 (0 Vdc) to 4095 (10 Vdc). Otherwise, for the **digital option** a logical *1* will turn on the pin while a *0* will stop it (considering that they work on a 5-12/24 Vdc range). In the case that we want to control a **relay option**, it will work out as a digital option, where a logical 1 will turn on the device and the logical 0 will shut down it.

By default, if not value option is provided it will be initialized as a 1 for the Digital and for the Relay or a 50% for the Analogic. If any other options are chosen, an error code will warn us. In order to call the function, we will do the following:

```
./set-digital-output <output> <value>
./set-analog-output <output> <value>
./set-relay <output> <value>
```

NOTE: You may need elevated permissions to execute these commands the first time, make sure you run them as *super user* ($sudo).

There is also the **PWM option** (that must be executed from the set-analog-output script)**,** that configures the output to work as PWM and you must specify as parameters the digital output and the width of the pulse. The width of the pulse is the high time period of the duty cycle and it has a range from 0 to 4095 (12 bits). For example, if you want a high time period of 25% you must put 1024 and if you want a high time period of 100% you must put 4095.

```
./set-analog-output <output> <pulse width value>
```

The PWM parameters of the script do not have to be modified to ensure a correct behaviour of the PWM.

NOTE: See the Section 8 to know which outputs are PWM supported.

⚠️ Warning:

There are some pins that can work as digital or analogue. In this case, if we have used these pins before in either digital or analogic and we want to switch its mode, we must call the set function providing *stop* to the value option; otherwise, there will be a system error. If a reboot is done, it is not necessary to do it.

The pins which can operate with both Analog/Digital configurations are:

- 0.5
- 0.6
- 0.7

- 1.5
- 1.6
- 1.7

- 2.5
- 2.6
- 2.7

```
./set-digital-output Q0.5 1
./set-digital-output Q0.5 stop
./set-analog-output A0.5 2048
```

As for the relay option, we can call the function as the following example:

```
./set-relay R0.5 1
```

Respect the PWM, all the digital output pins can work as PWM, so we must also consider the stop function in case that they are running as digital or analog (in the case of these ones that can work as analog or digital).

```
./set-analog-output A0.5 2048
./set-digital-output A0.5 stop
./set-analog-output Q0.5 4095
```

Or:

```
./set-digital-output Q0.5 1
./set-digital-output Q0.5 stop
./set-analog-output Q0.5 4095
```

13.3 Get Value

The ***get-[option]-input*** function will show the value of the selected input pin. It will only be provided the pin with which we are going to work. For the **Digital Option**, input pins go from Ix.0 to Ix.6, while the **Analogic Option** pins go from Ix.7 to Ix.12. The digital function can return a 0 or 1 (considering that they work on a 5-12/24 Vdc range). The analog function works on a 11-bit resolution range, returning a value from 0 (0 Vdc) to 2047 (10 Vdc).

In order to call the function, we will do the following:

```
./get-digital-input <input>
./get-analog-input <input>
```

As for the **get-temp** function, it will return the actual temperature of the device. Any parameter must be passed, as calling the function itself will return the temperature in milli-degrees.

In order to call the function, we will do the following:

```
./get-temp
```

13.4 Fan Functions

13.4.1 Start Fan

The **start-fan** function will activate the fan of the Raspberry PLC on the moment that the command is called. Consider that, if there is any temperature control system by default, the start-fan function could disable it.

```
./start-fan
```

13.4.2 Stop Fan

The **stop-fan** function will deactivate the fan of the Raspberry PLC on the moment that the command is called. Consider that, if there is any fan control system by default, the stop-fan function could disable it.

```
./stop-fan
```

⚠️ **Warning:**

Note that calling these functions can alter the behaviour of the temperature control system in the event of any.

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

14.1 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/ttySC0
```

➔ Configure the RS-485(1) por using the command:

```
stty (baudrate) -F /dev/ttySC1
```

This are the baudrates 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 opened waiting for data to read. Cancel it any time pressing Ctrl + C:

```
cat /dev/ttySC0
```

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

```
echo "Test message" > /dev/ttySC1
```

14.2 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 3.3V when nothing is connected. The pull-up configuration is established by default in these pins.
If it is meant to work them as a GPIO at 3.3V, it has to be considered that they are pull-up inputs.

14.3 SPI

The Raspberry PLC pins used for the SPI bus are summarized 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	GPIO2/GPIO3/GPIO15/GPIO14

14.4 TTL

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

14.5 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 windows, 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, the 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/ and 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!

14.6 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
}
```

14.7 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, then type the following instruction:

```
quit
```

and 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 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.

14.8 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.
```

14.9 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.

14.1 UPS Shield

All Raspberry PLC Family devices include the UPS Shield. 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 provides that the Raspberry follow a correct Shut Down process, avoiding any SD corruption issue. The recharge time of the UPS is less than 1 minute, and the reconnecting

time is less that 20 seconds. To know more technical specs about this device, you can consult the **Raspberry PI UPS and RTC Shield User Manual**.

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

14.1.1 How to execute commands before the Power Off process

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 this path inside the /etc folder:

sudo mkdir -p rpishutdown/hooks/

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

sudo nano 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-poweroffs

Following these steps, the file must be executed before the Raspberry Power Off.

15 Additional family features

15.1 GPRS Family

The GPRS Raspberry Pi PLC family works using the Adafruit SIM800L module, the integrated one for the use of GPRS/GSM in this PLC. The internal connections between the SIM800L and the Raspberry Pi 4 are the following:

RASPBERRY PI 4	SIM800L PINOUT
5 VDC	Vcc
GND	GND
GPIO14 (TXD0)	RxD
GPIO15 (RXD0)	TxD
GPIO16 (RXD0)	RST

In order to work with the Raspberry PLC GPRS module, we have to enable the hardware serial while the console must be disabled. For doing this, we will be executing the **raspi-config** interface.

```
sudo raspi-config
```

A new window will be open with different configuration options of the Raspberry. Once there, we will have to choose the option **interfacing options** and then the P6 option, **serial** for enabling or disabling the serial connection. Then it will be asked as if we would like a **login shell**, which we will have to reject, choosing to not work with it. After it, a new question regarding about enabling the **serial port hardware** will be shown. We must accept it.

If we have done the steps satisfactorily, a message saying that the serial login shell is disabled and the serial interface is enabled will be shown.

We are going to work with the link layer protocol PPP (Point-to-Point). By default, Raspberry Pi PLC will have the tool preinstalled and his dependencies, if not install **ppp** and the newest **libcap0** version using the apt-get install command.

For doing the first configurations with the GPRS module we will have to modify the following files located on their corresponding directories.

- **ppp-secrets**: /etc/ppp/pap-secrets
- **gprs-connect**: /etc/ppp/gprs-connect
- **gprs-disconnect**: /etc/ppp/gprs-disconnect
- **gprs**: /etc/ppp/peers/gprs
- **ppp0**: /etc/network/interfaces.d/ppp0

The files GPRS-connect and GPRS-disconnect will be tests used for doing the basics about the communications, getting an IP for a future use. As you can see on them, on the left side of the lines there is the response that the GPRS will send after getting the order specified on the right side of the line. For example, **AT+CGDCONT=1,"IP","YOUR-APN-HERE","0.0.0.0",0,0** will expect an **OK**.

The file that would be executing the whole process is the GPRS. On this example, in the GPRS file is called the gprs-connect, if different functions are wished to be implemented, they should be configured there.

In some of these files we must apply some changes in order to work with the SIM. The files have some sections where will be found **"YOUR-USER-HERE"**, **"YOUR-PASSWORD-HERE"** or **"YOUR-APN-HERE"** option, on which the user must configure his username, password and the APN that it is using. Some of the files will require modifying them with the superuser option *sudo*. The options that we will have to change are:

- **/etc/ppp/pap-secrets**: User and Password
- **/etc/ppp/gprs-connect**: APN
- **/etc/ppp/peers/gprs**: User

After setting this configuration, a reboot of the module must be set in order to apply the changes.

```
> sudo reboot
```

If there is a problem during the process, doing a manual reset to the GPRS module should solve the issue. For doing the reset we will have to execute the following commands:

```
echo 16 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio16/direction
echo 1 > /sys/class/gpio/gpio16/value
echo 0 > /sys/class/gpio/gpio16/value
echo 1 > /sys/class/gpio/gpio16/value
echo 16 > /sys/class/gpio/unexport
```

15.2 Build a Desktop Raspberry PLC Image from a Raspberry Pi OS 32-bit version

The Raspberry PLC family of products incorporates as standard a microSD card with the Raspberry Pi OS Lite operating system without a graphical interface for better 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.

Here are the steps to do it easily once the new image is installed:

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 from the Raspberry Pi Imager.
4. Once written, introduce the microSD card in your Raspberry PLC and do the setup.
5. Create a file in your Raspberry PLC called install.sh, and copy the following content:

https://apps.industrialshields.com/main/rpi/rpiplc/install.sh

6. Run this command to change the permissions:

```
chmod 755 install.sh
```

7. Finally, run it as a superuser (You will need to introduce the User and Password to enable the instruction):

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

After the reboot is done, the Raspberry Pi image will be now a Raspberry PLC image with all the required services, bash scripts and files.

15.3 Build a Desktop Raspberry PLC Image from a Raspberry Pi OS 64-bit version

If you want to install a 64-bit Linux image on the Raspberry PLC, you must take into account some considerations. First of all, there are not many 64-bit Linux images supported by Raspberry Pi 4. Although that, there is an official Raspberry repository with 64-bit images that are in beta test version:

This one for the Lite version:

https://downloads.raspberrypi.org/raspios_lite_arm64/images/

This one for the Desktop version:

https://downloads.raspberrypi.org/raspios_arm64/images/

The installation procedure of the image is the same previously commented, using the Raspberry Pi Imager Tool:

https://www.raspberrypi.org/downloads/

Once the image is installed on the microSD card, some steps must be followed in order to install all the services, scripts and executables which 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. The procedure here is a little more complex than the previously one:

1. First, we are going to follow the steps shown on the section 15.2 respect the execution of the **install.sh** file. This bash script will automatize almost all the required processes.

2. We will consult the specific path of the 64-bit compiled files for Raspberry image on our repository:

   https://apps.industrialshields.com/main/rpi64/

   There, we can find the correspondent services files about the hw-config and rpishutdown. We will only need to download the files called like the services: **hw-config** and **rpishutdown**, and we are going to move them to the path **/usr/local/bin/.** This path is usually created by default so, in case that it is not created, we are going to execute this command:

   sudo mkdir -p /usr/local/bin/

   Once we did this, we are going to execute these commands in the following order:

   cd /usr/local/bin/

   sudo chmod ugo+x rpishutdown

   sudo chmod ugo+x temperature-control

   sudo chmod ugo+x hw-config

3. We are going to activate the services because, when the system restarts, they must run automatically:

   sudo systemctl enable hw-config.service

   sudo systemctl enable rpishutdown.service

   sudo systemctl enable temperature-control.service

   To start the services at this moment:

   sudo systemctl start hw-config.service

   sudo systemctl start rpishutdown.service

   sudo systemctl start temperature-control.service

If it is needed, we can stop the services:

sudo systemctl stop hw-config.service

sudo systemctl stop rpishutdown.service

sudo systemctl stop temperature-control.service

16 **Revision Table**

Revision Number	Date	Changes
0	30/10/2020	First implementation
1	09/12/20	Change on the IOs table
2	03/02/21	Change on Section 12
3	05/02/21	Change on the screen printing of section 7.1
4	10/02/21	Change on the PWM outputs
5	11/02/21	Added Section 14 "Additional Family Features"
6	15/02/21	Added Section 7 "How to transfer files"
7	02/03/21	I/Os table added and changes on size and communications
8	29/04/21	Added communication examples
9	28/06/21	Change on Section 8.1
10	01/07/21	Added elevated permission info in RTC and I/O
11	28/07/21	Change on Section 1.3
12	09/08/21	Change on Section 8.2
13	19/08/21	Analog I/Os resolution updated
14	08/09/21	Sections 15.2 and 15.3 added

About Industrial Shields:

Direction: Fàbrica del Pont, 1-11

Zip/Postal Code: 08272

City: Sant Fruitós de Bages (Barcelona)

Country: Spain

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

Mail: industrialshields@industrialshields.com