



INDUSTRIAL SHIELDS

Touchberry 7" & Tinkertouch 7" User Guide

10 Configurable I/Os – RS485 – RS232 –
UPS Included



Touchberry 7" & TinkerTouch 7" User Guide

Revised December 2022

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 operating, to functions, and to the technical data of the signal modules, power supply modules and interface modules.

Intended Audience

This User Guide is intended for the following audience:

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



Warnings:

- Unused pins should not be connected. Ignoring the directive may damage the controller.
- Improper use of this product may severely damage the Panel PC.
- Refer to the Panels User Guide regarding wiring considerations.
- Before using this product, it is the responsibility of the user to read the product's User Guide and all accompanying documentation.
- 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.
- Care should be taken when servicing electrostatic sensitive components. The manufacturer's recommendations for these components should be followed.
- The Panel PC family are Open Type panels. It is required that you install Panel PC in a housing, cabinet, or electric control room. Entry to the housing, cabinet, or electric

control room should be limited to authorized personnel. Failure to follow these installation requirements could result in severe personal injury and/or property damage. Always follow these requirements when installing Panel PC.

- In case of installation or maintenance of the TinkerTouch 7" or Touchberry 7" please follow the instructions marked in the Installation and Maintenance section.
- 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.



Avertissements:

- Les broches non utilisées ne doivent pas être connectées. Ignorer la directive peut endommager le contrôleur.
- Une utilisation incorrecte de ce produit peut endommager gravement le contrôleur.
- Reportez-vous au Guide de l'utilisateur du Panel PC pour les considérations de câblage.
- Avant d'utiliser ce produit, il incombe à l'utilisateur de lire le Guide de l'utilisateur du produit et la documentation qui l'accompagne.
- La maintenance doit être effectuée par personnel qualifié familiarisé avec la fabrication, le fonctionnement et les dangers liés au contrôleur.
- La maintenance doit être effectuée avec l'équipement hors service et déconnectée de toutes les sources d'alimentation.
- Faites attention lors de l'entretien des composants sensibles à l'électricité statique. Les recommandations du fabricant pour ces composants doivent être suivies.
- Les automates de la famille Panel PC sont des contrôleurs de type ouvert. Il est nécessaire d'installer l'automate Panel PC dans un boîtier, une armoire ou une salle de contrôle électrique. L'accès au boîtier, à l'armoire ou à la salle de commande électrique doit être limité au personnel autorisé. Le non-respect de ces exigences d'installation peut entraîner des blessures graves et/ou des dommages matériels importants. Respectez toujours ces exigences lors de l'installation des automates de la TinkerTouch 7" et Touchberry 7".
- En cas d'installation ou de maintenance du TinkerTouch 7" et Touchberry 7", veuillez suivre les instructions indiquées dans la section Installation et Maintenance.
- Ne débranchez pas l'équipement en présence d'une atmosphère inflammable ou combustible. La déconnexion de l'équipement en présence d'une atmosphère inflammable ou combustible peut provoquer un incendie ou une explosion pouvant entraîner la mort, des blessures graves et/ou des dommages matériels.

Application Considerations and Warranty

Read and Understand this Manual

Please read and understand this manual before using the product. Please consult your comments or questions to Industrial Shields before using the product.

Application Consideration

THE PRODUCTS CONTAINED IN THIS DOCUMENT ARE NOT SAFETY RATED. THEY SHOULD NOT BE RELIED UPON AS A SAFETY COMPONENT OR PROTECTIVE DEVICE FOR ENSURING SAFETY OF PERSONS, AS THEY ARE NOT RATED OR DESIGNED FOR SUCH PURPOSES.

Please know and observe all prohibitions of use applicable to the products.

FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, NEVER USE THE INDUSTRIAL SHIELDS PRODUCTS.

NEVER USE THE INDUSTRIAL SHIELDS PRODUCTS BEFORE THEY ARE PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Industrial Shields shall not be responsible for conformity with any codes, regulations or standards that apply to the combination of products in the customer's application or use of the product.

The following are some examples of applications for which particular attention must be given. This is not intended to be an exhaustive list of all possible uses of the products, nor is it intended to imply that the uses may be suitable for the products:

- Systems, machines, and equipment that could present a risk to life or property.
- Nuclear energy control systems, combustion systems, railroad systems, aviation systems, medical equipment, amusement machines, vehicles, safety equipment, and installation subject to separate industry or government regulations.
- WARNING: On outdoor operations, locations involving potential chemical contamination or electrical interference, conditions or uses not described in this document please contact Industrial Shields for further information.

At the customer's request, INDUSTRIAL SHIELDS will provide applicable third-party certification documents identifying ratings and limitations of use that apply to the products. This information by itself is not sufficient for a complete determination of the suitability of the products in combination with the system, machine, end product, or other application or use.

Intended use or of Industrial Shields products

Consider the following:

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

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

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

Disclaimers

Weights and Dimensions

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

Performance Data

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

Change in Specifications

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

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

Errors and Omissions

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

The control and drive components of an Industrial Shields Panel PC are approved for industrial and commercial use in industrial line supplies. Their use in public line supplies requires a different configuration and/or additional measures. These components may only be operated in closed housings or in higher-level control cabinets with protective covers that are closed, and when all of the protective devices are used. These components may only be handled by qualified and trained technical personnel who are knowledgeable and observe all of the safety information and instructions on the components and in the associated technical user documentation. When carrying out a risk assessment of a machine in accordance with the EU Machinery Directive, the machine manufacturer must consider the following residual risks associated with the control and drive components of a PDS.

1. Unintentional movements of driven machine components during commissioning, operation, maintenance, and repairs caused by, for example: – Hardware defects and/or software errors in the sensors, controllers, actuators, and connection technology – Response times of the controller and drive – Operating and/or ambient conditions not within the scope of the specification – Condensation / conductive contamination – Parameterization, programming, cabling, and installation errors – Use of radio devices / cellular phones in the immediate vicinity of the controller – External influences / damage.
2. Exceptional temperatures as well as emissions of noise, particles, or gas caused by, for example: – Component malfunctions – Software errors – Operating and/or ambient conditions not within the scope of the specification – External influences / damage.
3. Hazardous shock voltages caused by, for example: – Component malfunctions – Influence of electrostatic charging – Induction of voltages in moving motors – Operating and/or ambient conditions not within the scope of the specification – Condensation / conductive contamination – External influences / damage
4. Electrical, magnetic and electromagnetic fields generated in operation that can pose a risk to people with a pacemaker, implants or metal replacement joints, etc. if they are too close.
5. Release of environmental pollutants or emissions as a result of improper operation of the system and/or failure to dispose of components safely and correctly.

Warranty and Limitations of Liability

Warranty

Industrial Shields's exclusive warranty is that the products are free from defects in materials and workmanship for a period of one year (or other period if specified) from date of sale by Industrial Shields.

INDUSTRIAL SHIELDS MAKES NO REPRESENTATION OR WARRANTY, EXPRESSED OR IMPLIED, REGARDING MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR PARTICULAR PURPOSE OF THE PRODUCTS. ANY BUYER OR USER ACKNOWLEDGES THAT THE BUYER OR USER ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. INDUSTRIAL SHIELDS DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED

Limitations of Liability

INDUSTRIAL SHIELDS SHALL NOT BE RESPONSIBLE FOR SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR COMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED ON CONTRACT, WARRANTY, NEGLIGENCE, OR STRICT LIABILITY.

IN NO EVENT SHALL INDUSTRIAL SHIELDS BE RESPONISBLE FOR WARRANTY, REPAIR OR OTHER CLAIMS REGARDING THE PRODUCTS UNLESS INDUSTRIAL SHIELDS'S ANALYSIS CONFIRMS THAT THE PRODUCTS WERE PROPERLY HANDLED, STORED, INSTALLED, AND MAINTAINED AND NOT SUBJECT TO CONTAMINATION, ABUSE, MISUSE, OR INAPPROPRIATE MODIFICATION OR REPAIR.

Table of Contents

1	General Description Touchberry 7" & Tinkertouch 7"	10
1.1	Touchberry 7" & Tinkertouch 7" pinout.....	10
1.2	Mechanical dimension	11
1.3	Touchberry 7" & Tinkertouch 7" pin layout:	12
1.4	Touchberry 7" & Tinkertouch 7" Jumpers Operation:	14
2	General Jumpers configuration (General Settings)	15
2.1	General Jumper Configuration. Left zone (General settings):.....	16
2.2.	General Jumper Configuration. Right zone (I/Os settings):.....	17
2.3.	General Jumper Configuration. RS-232 & RS-485 configuration (I/Os settings):	19
2.3.1	For Touchberry 7"	19
2.3.2	For TinkerTouch 7"	20
3	Symbology	21
4	How to connect the touch Screen of the Touchberry 7"/Tinkertouch 7"	22
5	How to connect the Touchberry 7"/Tinkertouch 7" to an external Power Supply ..	22
6	Communications & RTC & uSD	24
6.1	RS-485	24
6.2	I2C.....	24
6.3	SPI.....	25
6.4	TTL	25
6.5	Ethernet.....	25
6.6	Wi-Fi	26
6.7	BLE.....	26
6.8	RTC.....	28
6.9	uSD	28
6.1	0 UPS Shield.....	28
6.10.1	How to execute commands before the Power Off process	29
7	Hardware TEST	29
7.1	Touchberry 7" hardware test scripts	30
7.1.1	Common library TEST	30
7.1.2	IOs TEST	33
7.1.3	RS-232 TEST	33

7.1.4	TTL TEST	34
7.1.5	RS-485 TEST	35
7.1.6	fan Test.....	36
7.2	Tinkertouch 7" hardware test scripts	36
7.2.1	Common library TEST	37
7.2.2	IOs TEST	40
7.2.3	RS-232 TEST	41
7.2.4	TTL TEST.....	41
7.2.5	RS-485 TEST	42
7.2.6	fan Test.....	43
8	Installation and Maintenance	44
9	Revision Table	47

1 General Description Touchberry 7" & Tinkertouch 7"

1.1 Touchberry 7" & Tinkertouch 7" pinout

A compact Panel PC in Open Source Hardware technology. With different Input/Outputs units and a screen with a resolution of 800x480 pixels.

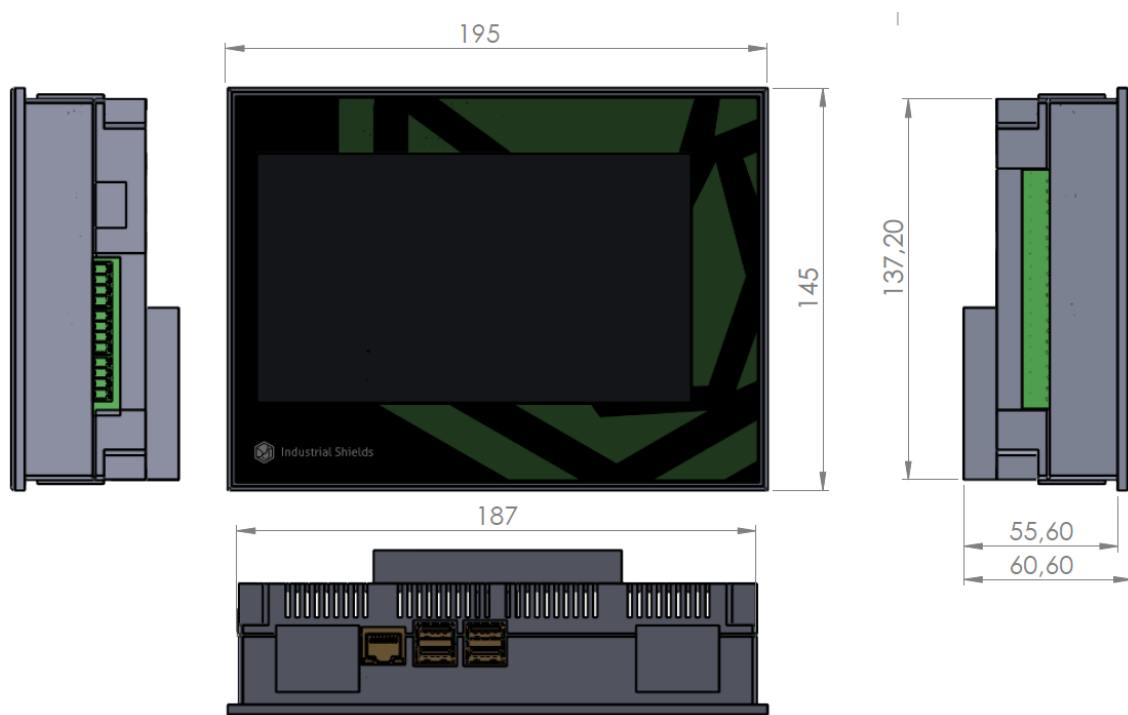
Next it is showed a table with the connection between the Raspberry Pi/Tinkerboard pinout and the Touchberry 7" & Tinkertouch 7" Panel PC Pinout.



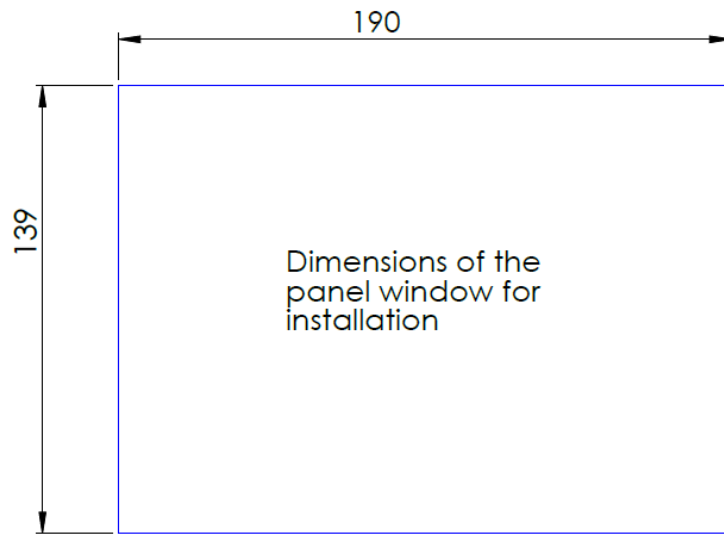
TinkerTouch 7" / Touchberry 7" Pinout	Raspberry PI 4B GPIO	Tinker board GPIO
---	-------------------------	----------------------

11	GPIO17	164
12	GPIO18	184
13	GPIO27	166
16	GPIO23	162
18	GPIO24	163
22	GPIO25	171
24	GPIO8	255
26	GPIO7	251
29	GPIO5	165
31	GPIO6	168
32	GPIO12	239
33	GPIO13	238
35	GPIO19	185
36	GPIO16	223
37	GPIO26	224
38	GPIO20	187
40	GPIO21	188

1.2 Mechanical dimension



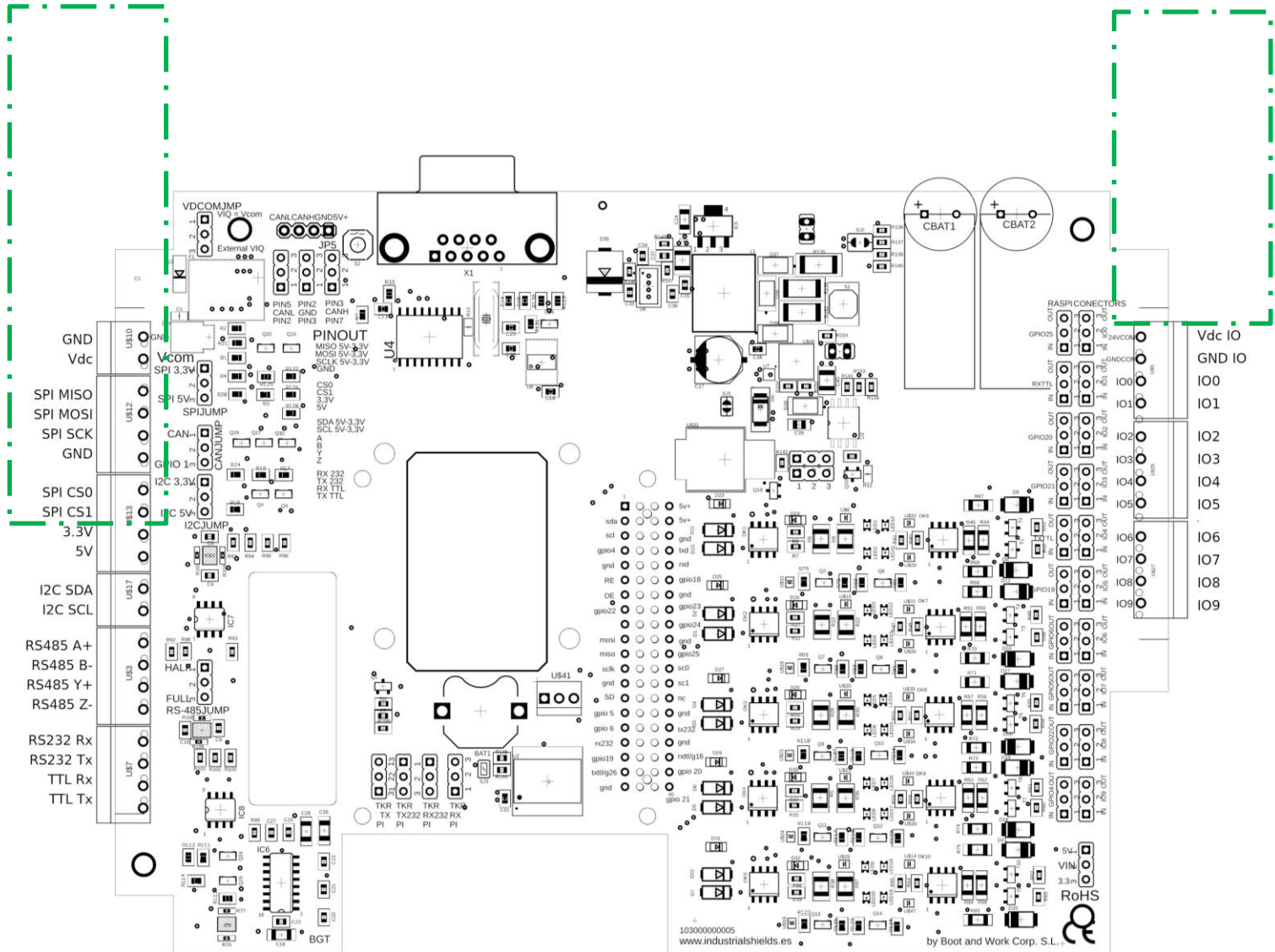
Panel Touch 7" is designed to be placed build-in an electrical box. Next it is showed the dimensions to be installed properly:



1.3 Touchberry 7" & Tinkertouch 7" pin layout:

The following image shows the distribution of the Panel PC pins (in the image you can see the Panel PC from the back).

On the right side are the GPIO pins of the Panel PC while the set of pins on the left corresponds to the communication and power pins of the equipment.

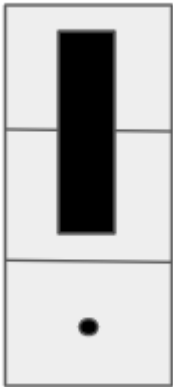


1.4 Touchberry 7" & Tinkertouch 7" Jumpers Operation:

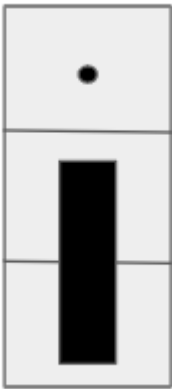
The configuration of the jumpers is explained below.

In each configuration the different positions are explained and here you can find a graphic reference to the configuration tables of section 2:

Example Jumper 1	Example Jumper 2
VIO = VCOM	VIO = VCOM
Enable 1	Enable 1
VIO external	VIO external



Example Jumper 1



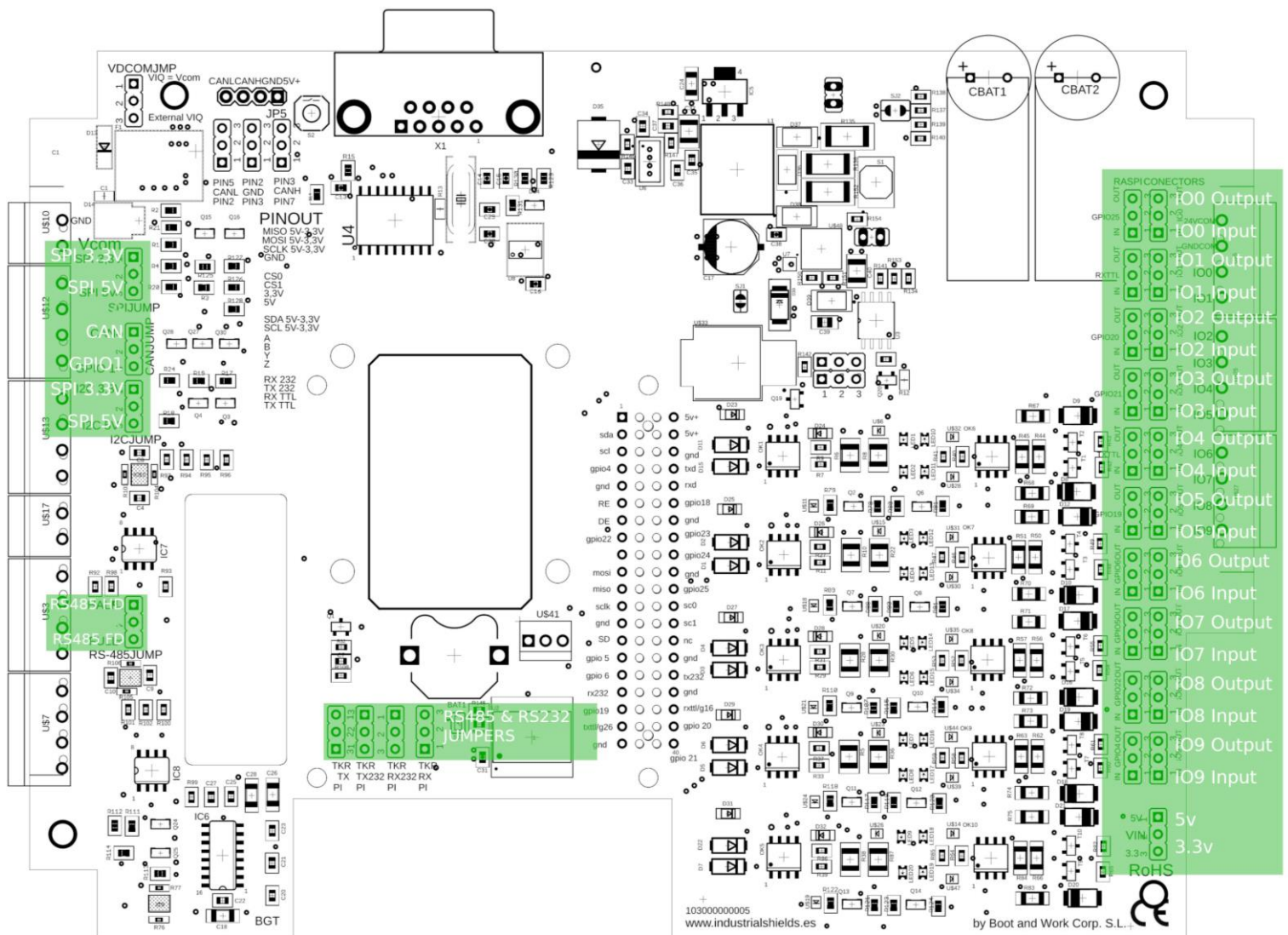
Example Jumper 2

Here you are selecting between the two possible configurations that the jumper allows you.

Using the Jumper Block you can select in which pins are you allowing the electricity to flow.

2 General Jumpers configuration (General Settings)

In the following image, you can see the position of the jumpers on the board located inside the Panel PC:



2.1 General Jumper Configuration. Left zone (General settings):

Named from top to bottom:

Left Jumper 3
SPI 3.3 V
Enable 3
SPI 5V

Left Jumper 4 enables SPI at 3.3V or 5V. Typical configurations showed below:

Left Jumper 3
SPI 3.3 V
Enable 3
SPI 5V

This configuration enables SPI at 3.3V.

Left Jumper 3
SPI 3.3 V
Enable 3
SPI 5V

This configuration enables SPI at 5V.

Left Jumper 4
I2C 3.3 V
Enable 4
I2C 5V

Left Jumper 4 enables I2C at 3.3V or 5V. Typical configurations showed below:

Left Jumper 4
I2C 3.3 V
Enable 4
I2C 5V

This configuration enables I2C at 3.3V.

Left Jumper 4
I2C 3.3 V
Enable 4
I2C 5V

This configuration enables I2C at 5V.

Left Jumper 5
RS-485 Half Duplex
Enable 5
RS-485 Full Duplex

Left Jumper 5 enables RS-485 Half Duplex or Full Duplex. Typical configuration showed below:

Left Jumper 5
RS-485 Half Duplex
Enable 5
RS-485 Full Duplex

This configuration enables RS-485 HD.

Left Jumper 5
RS-485 Half Duplex
Enable 5
RS-485 Full Duplex

This configuration enables RS-485 FD.

2.2. General Jumper Configuration. Right zone (I/Os settings):

By default, IO0 through IO4 are configured as Inputs, while IO5 through IO9 are configured as Outputs

Named from top to bottom.

Right Jumper 1	
Left	Right
OUT	OUT
GPI025	IO0
IN	IN

Right Jumper 2	
Left	Right
OUT	OUT
GPI016	IO1
IN	IN

Right Jumper 3	
Left	Right
OUT	OUT
GPI020	IO2
IN	IN

Right Jumper 4	
Left	Right
OUT	OUT
GPI021	IO3
IN	IN

Right Jumper 5	
Left	Right
OUT	OUT
GPI026	IO4
IN	IN

Right Jumper 6	
Left	Right
OUT	OUT
GPI019	IO5
IN	IN

Right Jumper 7	
Left	Right
OUT	OUT
GPI006	IO6
IN	IN

Right Jumper 8	
Left	Right
OUT	OUT
GPI005	IO7
IN	IN

Right Jumper 9	
Left	Right
OUT	OUT
GPI022	IO8
IN	IN

Right Jumper 10	
Left	Right
OUT	OUT
GPI004	IO9
IN	IN

From Right Jumper 1 to Right Jumper 10, the functionality is the same. You can enable IOs as Outputs or Inputs. Typical example with Right Jumper 1:

Right Jumper 1	
Left	Right
OUT	OUT
GPI025	IO0
IN	IN

This configuration enables IO0 as an Output.

Right Jumper 1	
Left	Right
OUT	OUT
GPI025	IO0
IN	IN

This configuration enables IO0 as an Input.

Right Jumper 11
Not Connected
VIN
VIN 3.3V

Right Jumper 11



WARNING: Touchberry 7” & Tinkertouch 7” ONLY HANDLES 3.3Vdc Vin. It comes with the 3.3Vdc Vin configuration by default, do not touch it! Do not connect Vin at NC or 5Vdc!

Right Jumper 11
VIN 5V
VIN
VIN 3.3V

This configuration enables Vin at 3.3V

2.3. General Jumper Configuration. RS-232 & RS-485 configuration (I/Os settings):

2.3.1 For Touchberry 7”

RS-232 Configuration:

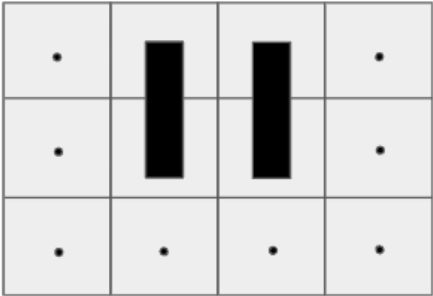
.	.	.	.
.			.
.			.

RS-485 Configuration:

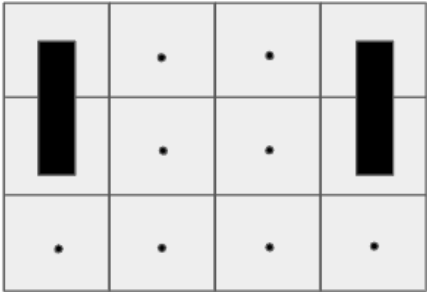
.	.	.	.
	.	.	
	.	.	

2.3.2 For TinkerTouch 7”

RS-232 Configuration:

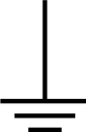


RS-485 Configuration:



3 Symbology

Table that includes all the symbology that is used in the serigraph of the Tinkertouch 7" and Touchberry 7":

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

4 How to connect the touch Screen of the Touchberry 7"/Tinkertouch 7"

- Connect USB port from the Panel PC to one of the empty USB ports.

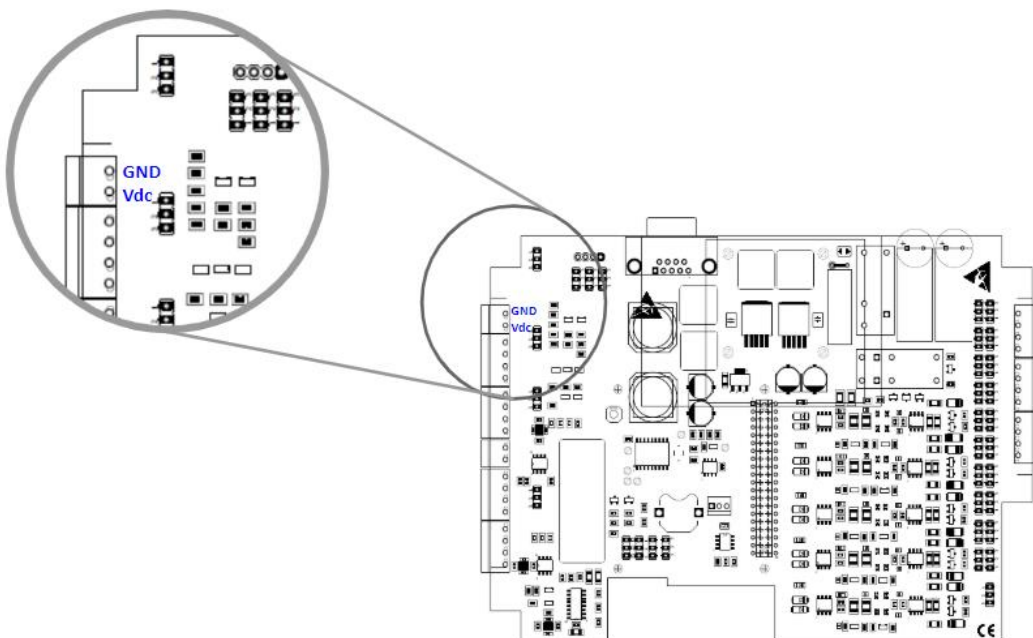
NOTE:

Panel PCs 7" use USB-USB cable.

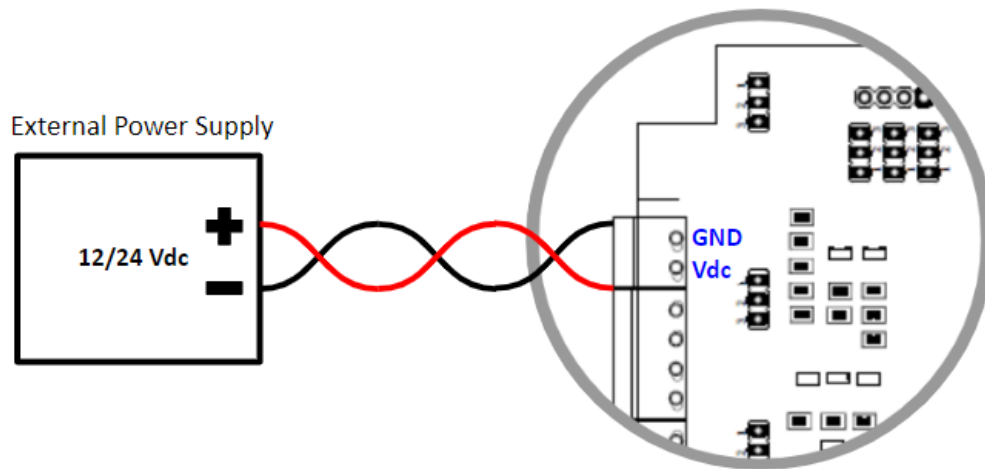


5 How to connect the Touchberry 7"/Tinkertouch 7" to an external Power Supply

- Touchberry 7"/Tinkertouch 7" is 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**.



Find attached a picture regarding the position of the external power inputs:



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.

6 Communications & RTC & uSD

6.1 RS-485

This device is provided with RS-485 full-duplex communication.

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) port using the command:

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

These 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
```

6.2 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.

6.3 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

6.4 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).

6.5 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!

6.6 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
}
```

6.7 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.

6.8 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.
```

6.9 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.

6.10 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 than 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.

6.10.1 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.

7 Hardware TEST

All the scripts showed in this section are installed on the Touchberry 7" & Tinkertouch 7".

7.1 Touchberry 7" hardware test scripts

In order to test all the scripts in your 7" Panel PC, you must activate before the SSH.

To activate it, you must go to:

Preferences -> Raspberry Configuration -> Activate SSH

SPI and I2C communications are disabled by default.

If you want to enable SPI and I2C communications, add the following lines to the /boot/config.txt file:

```
enable_uart=1
dtparam=spi=ON
dtparam=i2c_arm=ON
```

7.1.1 Common library TEST

There is a common library for all the test called *common.sh*. It defines common functions and pinout mappings to access them from the test scripts.

common.sh

```
#!/bin/bash

IOS="25 12 16 20 21 26 19 13 6 5"
SPI_CS0="8"
SPI_CS1="7"
INPUTS="${IOS}"
OUTPUTS="${IOS}"
FAN="18"

SERIAL_PORT="/dev/ttyS0"
RS485_PORT="${SERIAL_PORT}"
RS485_DE="27"
RS485_RE="17"
RS232_PORT="${SERIAL_PORT}"
TTL_PORT="${SERIAL_PORT}"

enable_io() {
    if [ ! -d "/sys/class/gpio/gpio${1}" ]; then
        echo "${1}" > "/sys/class/gpio/export"
    fi
    echo "${2}" > "/sys/class/gpio/gpio${1}/direction"
}

disable_io() {
    if [ -d "/sys/class/gpio/gpio${1}" ]; then
        echo "in" > "/sys/class/gpio/gpio${1}/direction"
        echo "${1}" > "/sys/class/gpio/unexport"
    fi
}

enable_input() {
```

```

    enable_io "${1}" "in"
}

disable_input() {
    disable_io "${1}"
}

read_input() {
    echo -n "ReadInput: gpio${1} = "
    cat "/sys/class/gpio/gpio${1}/value"
}

enable_output() {
    enable_io "${1}" "out"
}

disable_output() {
    disable_io "${1}"
}

set_output() {
    echo "1" > "/sys/class/gpio/gpio${1}/value"
}

clear_output() {
    echo "0" > "/sys/class/gpio/gpio${1}/value"
}

enable_serial() {
    DEV="${1}"
    RATE="${2}"
    SERIAL_RATE="${RATE}"
    SERIAL_DATABITS="8"
    if [ -n "${3}" ]; then
        DATABITS="cs${3}"
        SERIAL_DATABITS="${3}"
    fi
    if [ "${4}" = "even" ]; then
        PARITY="parenb -parodd"
        SERIAL_DATABITS=$((SERIAL_DATABITS + 1))
    elif [ "${4}" = "odd" ]; then
        PARITY="parenb parodd"
        SERIAL_DATABITS=$((SERIAL_DATABITS + 1))
    fi
    if [ "${5}" = "2" ]; then
        STOPBITS="cstopb"
        SERIAL_DATABITS=$((SERIAL_DATABITS + 2))
    else
        SERIAL_DATABITS=$((SERIAL_DATABITS + 1))
    fi
    SERIAL_DATABITS=$((SERIAL_DATABITS + 1)) # Start bit
    stty -F "${DEV}" ${RATE} ${DATABITS} ${PARITY} ${STOPBITS} ignpar ignbrk -brkint -
    icrnl -imaxbel -opost -onlcr -isig -icanon -iexten -echo -echoe -echok -echoctl -
    echoke noflsh -ixon -crtcts
}

send_serial() {
    echo -ne ${2} > ${1}
    SERIAL_DATALEN=$(echo ${2} | wc -c)
}

flush_serial() {
    T=$(echo "scale=3;${SERIAL_DATABITS} * ${SERIAL_DATALEN} / ${SERIAL_RATE}" | bc)
    sleep ${T}
}

```

```

receive_serial_byte() {
    dd if="${1}" bs=1 count=1 2> /dev/null
}

receive_serial() {
    dd if="${1}" bs=1 2> /dev/null
}

enable_rs485() {
    enable_serial "${RS485_PORT}" ${@}
    enable_output "${RS485_RE}"
    enable_output "${RS485_DE}"
    clear_output "${RS485_RE}"
    clear_output "${RS485_DE}"
}

disable_rs485() {
    disable_output "${RS485_RE}"
    disable_output "${RS485_DE}"
}

send_rs485_full() {
    set_output "${RS485_DE}"
    send_serial "${RS485_PORT}" "${1}"
    flush_serial
    clear_output "${RS485_DE}"
}

send_rs485_half() {
    set_output "${RS485_RE}"
    set_output "${RS485_DE}"
    send_serial "${RS485_PORT}" "${1}"
    flush_serial
    clear_output "${RS485_DE}"
    clear_output "${RS485_RE}"
}

receive_rs485_byte() {
    receive_serial_byte "${RS485_PORT}"
}

receive_rs485() {
    receive_serial "${RS485_PORT}"
}

enable_rs232() {
    disable_rs485
    enable_serial "${RS232_PORT}" ${@}
}

send_rs232() {
    send_serial "${RS232_PORT}" "${1}"
}

receive_rs232_byte() {
    receive_serial_byte "${RS232_PORT}"
}

receive_rs232() {
    receive_serial "${RS232_PORT}"
}

enable_ttl() {
    disable_rs485
    enable_serial "${TTL_PORT}" ${@}
}

```

```

send_ttl() {
    send_serial "${TTL_PORT}" "${1}"
}

receive_ttl_byte() {
    receive_serial_byte "${TTL_PORT}"
}

receive_ttl() {
    receive_serial "${TTL_PORT}"
}

```

7.1.2 IOs TEST

There are two scripts in order to control the inputs and outputs of the TouchScreen 7”, get-digital-value and set-digital value. They are located on the directory **/home/pi/test**.

7.1.2.1 Outputs

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. A logical 1 will turn on the pin while a 0 will stop it.

By default, if not value option is provided it will be initialized as a 1 for the Digital and f. 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>
```

Example:

```
./set-digital-output IO5 1
```

7.1.2.2 Inputs

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. Input pins go from Ix.0 to Ix.9. In order to call the function, we will do the following:

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

Example:

```
./get-digital-output IO5
```

7.1.3 RS-232 TEST

The *rs232-periodic-sender.sh* script periodically sends a message through the **RS-232** port periodically.

The *rs232-receiver.sh* script prints the received messages from the RS-232 port. It is possible to connect the RS-232 RX pin to the RS-232 TX pin and run both scripts at the same time.

rs232-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello RS-232!\n"
PERIOD=1

enable_rs232 9600
while true; do
    echo "Send \"${DATA}\" using RS-232"
    send_rs232 "${DATA}"
    sleep "${PERIOD}"
done
```

rs232-receiver.sh

```
#!/bin/bash

. common.sh

enable_rs232 9600
echo "Receiving using RS-232 ..."
receive_rs232
```

7.1.4 TTL TEST

The *ttn-periodic-sender.sh* script periodically sends a message through the **TTL serial** port.

The *ttn-receiver.sh* script prints the received messages from the TTL serial port.

It is possible to connect the TTL RX pin to the TTL TX pin and run both scripts at the same time.

ttn-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello TTL!\n"
PERIOD=1

enable_ttn 9600
while true; do
    echo "Send \"${DATA}\" using TTL"
    send_ttn "${DATA}"
    sleep "${PERIOD}"
done
```

ttn-receiver.sh

```
#!/bin/bash

. common.sh

enable_ttn 9600
```

```
echo "Receiving using TTL ..."
receive_ttl
```

7.1.5 RS-485 TEST

The *rs485-hd-periodic-sender.sh* script periodically sends a message through the **RS-485 port**, using the RS-485 half duplex mode. It sends the data through the A and B pins.

The *rs485-fd-periodic-sender.sh* script sends a message through the **RS-485 port** periodically, using the RS-485 full duplex mode. It sends the data through the Y and Z pins. In this mode, the A and B pins are used to receive data.

The *rs485-receiver.sh* script prints the received messages from the RS-485 port, in both half duplex and full duplex modes.

In the half duplex mode it is required another device to receive data from and send data to the panel PC.

In the full duplex mode it is not required an additional device and it is possible to connect the A/B pins to the Y/Z ones. In this case it is possible to run the send and receive scripts at the same time.

rs485-hd-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello RS-485!\n"
PERIOD=1

enable_rs485 9600
while true; do
    echo "Send \"${DATA}\" using RS-485 half duplex"
    send_rs485_half "${DATA}"
    sleep "${PERIOD}"
done
```

rs485-fd-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello RS-485!\n"
PERIOD=1

enable_rs485 9600
while true; do
    echo "Send \"${DATA}\" using RS-485 full duplex"
    send_rs485_full "${DATA}"
    sleep "${PERIOD}"
done
```

rs485-receiver.sh

```
#!/bin/bash
```

```
. common.sh
enable_rs485 9600
echo "Receiving using RS-485 ..."
receive_rs485
```

7.1.6 fan Test

12Vdc fan can be attached inside the Touchberry 7" using the fan connectors.

This sketch tests the fan operation by activating and deactivating it:

7.1.6.1 12.4.1 Start Fan

The **start-fan** function will activate the fan of the Raspberry PLC on the moment that the command is called. Example:

```
./start-fan
```

7.1.6.2 12.4.2 Stop Fan

The **stop-fan** function will deactivate the fan of the Raspberry PLC on the moment that the command is called. Example:

```
./stop-fan
```

7.2 Tinkertouch 7" hardware test scripts

In order to test all the scripts in your 7" Panel PC, you must activate before the SSH.

To activate it, you must go to:

Preferences -> Tinkerboard Configuration -> Activate SSH

Also, take into account that in order to do the tests, it's necessary download the Python-smbus program in your Tinkertouch 7".

To download Python-smbus it is necessary to have internet connection.

Once you have an internet connection, you must open the command line and enter the following command:

```
$> sudo apt-get install python-smbus
```

7.2.1 Common library TEST

There is a common library for all the test called *common.sh*. It defines common functions and pinout mappings to access them from the test scripts.

common.sh

```
#!/bin/bash

IOS="171 239 223 187 188 224 185 238 168 165"
SPI_CS0="255"
SPI_CS1="251"
INPUTS="${IOS}"
OUTPUTS="${IOS}"
FAN="184"

SERIAL_PORT0="/dev/ttyS2"
SERIAL_PORT1="/dev/ttyS1"
RS485_PORT="${SERIAL_PORT1}"
RS485_DE="166"
RS485_RE="164"
RS232_PORT="${SERIAL_PORT0}"
TTL_PORT="${SERIAL_PORT0}"

enable_io() {
    if [ ! -d "/sys/class/gpio/gpio${1}" ]; then
        echo "${1}" > "/sys/class/gpio/export"
    fi

    echo "${2}" > "/sys/class/gpio/gpio${1}/direction"
}

disable_io() {
    if [ -d "/sys/class/gpio/gpio${1}" ]; then
        echo "in" > "/sys/class/gpio/gpio${1}/direction"
        echo "${1}" > "/sys/class/gpio/unexport"
    fi
}

enable_input() {
    enable_io "${1}" "in"
}

disable_input() {
    disable_io "${1}"
}

read_input() {
    echo -n "ReadInput: gpio${1} = "
    cat "/sys/class/gpio/gpio${1}/value"
}

enable_output() {
    enable_io "${1}" "out"
}

disable_output() {
    disable_io "${1}"
}
```

```

set_output() {
    echo "1" > "/sys/class/gpio/gpio${1}/value"
}

clear_output() {
    echo "0" > "/sys/class/gpio/gpio${1}/value"
}

enable_serial() {
    DEV="${1}"
    RATE="${2}"
    SERIAL_RATE="${RATE}"
    SERIAL_DATABITS="8"
    if [ -n "${3}" ]; then
        DATABITS="cs${3}"
        SERIAL_DATABITS="${3}"
    fi
    if [ "${4}" = "even" ]; then
        PARITY="parenb -parodd"
        SERIAL_DATABITS=$((SERIAL_DATABITS + 1))
    elif [ "${4}" = "odd" ]; then
        PARITY="parenb parodd"
        SERIAL_DATABITS=$((SERIAL_DATABITS + 1))
    fi
    if [ "${5}" = "2" ]; then
        STOPBITS="cstopb"
        SERIAL_DATABITS=$((SERIAL_DATABITS + 2))
    else
        SERIAL_DATABITS=$((SERIAL_DATABITS + 1))
    fi
    SERIAL_DATABITS=$((SERIAL_DATABITS + 1)) # Start bit
    stty -F "${DEV}" ${RATE} ${DATABITS} ${PARITY} ${STOPBITS} ignpar ignbrk -brkint -
    icrnl -imaxbel -opost -onlcr -isig -icanon -iexten -echo -echoe -echok -echoctl -
    echoke noflsh -ixon -crtcts
}

send_serial() {
    echo -ne ${2} > ${1}
    SERIAL_DATALEN=$(echo ${2} | wc -c)
}

flush_serial() {
    T=$(echo "scale=3;${SERIAL_DATABITS} * ${SERIAL_DATALEN} / ${SERIAL_RATE}" | bc)
    sleep ${T}
}

receive_serial_byte() {
    dd if="${1}" bs=1 count=1 2> /dev/null
}

receive_serial() {
    dd if="${1}" bs=1 2> /dev/null
}

enable_rs485() {
    enable_serial "${RS485_PORT}" ${@}
    enable_output "${RS485_RE}"
    enable_output "${RS485_DE}"
    clear_output "${RS485_RE}"
    clear_output "${RS485_DE}"
}

disable_rs485() {
    disable_output "${RS485_RE}"
    disable_output "${RS485_DE}"
}

```

```
}

send_rs485_full() {
    set_output "${RS485_DE}"
    send_serial "${RS485_PORT}" "${1}"
    flush_serial
    clear_output "${RS485_DE}"
}

send_rs485_half() {
    set_output "${RS485_RE}"
    set_output "${RS485_DE}"
    send_serial "${RS485_PORT}" "${1}"
    flush_serial
    clear_output "${RS485_DE}"
    clear_output "${RS485_RE}"
}

receive_rs485_byte() {
    receive_serial_byte "${RS485_PORT}"
}

receive_rs485() {
    receive_serial "${RS485_PORT}"
}

enable_rs232() {
    disable_rs485
    enable_serial "${RS232_PORT}" ${@}
}

send_rs232() {
    send_serial "${RS232_PORT}" "${1}"
}

receive_rs232_byte() {
    receive_serial_byte "${RS232_PORT}"
}

receive_rs232() {
    receive_serial "${RS232_PORT}"
}

enable_ttl() {
    disable_rs485
    enable_serial "${TTL_PORT}" ${@}
}

send_ttl() {
    send_serial "${TTL_PORT}" "${1}"
}

receive_ttl_byte() {
    receive_serial_byte "${TTL_PORT}"
}

receive_ttl() {
    receive_serial "${TTL_PORT}"
}
```

7.2.2 IOs TEST

There are two scripts in order to control the inputs and outputs of the TouchScreen 7", `get-digital-value` and `set-digital value`. They are located on the directory **/home/pi/test**.

7.2.2.1 Outputs

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. A logical 1 will turn on the pin while a 0 will stop it.

By default, if not value option is provided it will be initialized as a 1 for the Digital and f. 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>
```

Example:

```
./set-digital-output IO5 1
```

7.2.2.2 Inputs

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. Input pins go from Ix.0 to Ix.9. In order to call the function, we will do the following:

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

Example:

```
./get-digital-output IO5
```

7.2.3 RS-232 TEST

The *rs232-periodic-sender.sh* script periodically sends a message through the **RS-232** port periodically.

The *rs232-receiver.sh* script prints the received messages from the RS-232 port.

It is possible to connect the RS-232 RX pin to the RS-232 TX pin and run both scripts at the same time.

rs232-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello RS-232!\n"
PERIOD=1

enable_rs232 9600
while true; do
    echo "Send \"${DATA}\" using RS-232"
    send_rs232 "${DATA}"
    sleep "${PERIOD}"
done
```

rs232-receiver.sh

```
#!/bin/bash

.common.sh

enable_rs232 9600
echo "Receiving using RS-232 ..."
receive_rs232
```

7.2.4 TTL TEST

The *ttn-periodic-sender.sh* script periodically sends a message through the **TTL serial** port.

The *ttn-receiver.sh* script prints the received messages from the TTL serial port.

It is possible to connect the TTL RX pin to the TTL TX pin and run both scripts at the same time.

ttn-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello TTL!\n"
PERIOD=1

enable_ttn 9600
while true; do
    echo "Send \"${DATA}\" using TTL"
    send_ttn "${DATA}"
    sleep "${PERIOD}"
done
```

ttl-receiver.sh

```
#!/bin/bash

. common.sh

enable_ttl 9600

echo "Receiving using TTL ..."
receive_ttl
```

7.2.5 RS-485 TEST

The *rs485-hd-periodic-sender.sh* script periodically sends a message through the **RS-485 port**, using the RS-485 half duplex mode. It sends the data through the A and B pins.

The *rs485-fd-periodic-sender.sh* script sends a message through the **RS-485 port** periodically, using the RS-485 full duplex mode. It sends the data through the Y and Z pins. In this mode, the A and B pins are used to receive data.

The *rs485-receiver.sh* script prints the received messages from the RS-485 port, in both half duplex and full duplex modes.

In the half duplex mode it is required another device to receive data from and send data to the panel PC.

In the full duplex mode it is not required an additional device and it is possible to connect the A/B pins to the Y/Z ones. In this case it is possible to run the send and receive scripts at the same time.

rs485-hd-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello RS-485!\n"
PERIOD=1

enable_rs485 9600
while true; do
    echo "Send \"${DATA}\" using RS-485 half duplex"
    send_rs485_half "${DATA}"
    sleep "${PERIOD}"
done
```

rs485-fd-periodic-sender.sh

```
#!/bin/bash

. common.sh

DATA="Hello RS-485!\n"
PERIOD=1

enable_rs485 9600
while true; do
    echo "Send \"${DATA}\" using RS-485 full duplex"
    send_rs485_full "${DATA}"
    sleep "${PERIOD}"
done
```

rs485-receiver.sh

```
#!/bin/bash

. common.sh
enable_rs485 9600
echo "Receiving using RS-485 ..."
receive_rs485
```

7.2.6 fan Test

12Vdc fan can be attached inside the Touchberry 7" using the fan connectors.

This sketch tests the fan operation by activating and desactivating it:

7.2.6.1 12.4.1 Start Fan

The **start-fan** function will activate the fan of the Raspberry PLC on the moment that the command is called. Example:

```
./start-fan
```

7.2.6.2 12.4.2 Stop Fan

The **stop-fan** function will deactivate the fan of the Raspberry PLC on the moment that the command is called. Example:

```
./stop-fan
```

8 Installation and Maintenance

Notes for installation:

- The installation position should be free from the following: dust or oil smoke, conductive dust, corrosive or flammable gas, high temperature, condensation, and rain.
- Besides, vibration and impact also affect the Panel PC normal operation and shorten its lifespan; electric shock, fire or misact also damages the product. During drilling or wiring, prevent the metal particles or wire segments from falling into the PLC casing, which may cause fire, fault or misact.
- After the PLC installation, clean the ventilation duct to prevent blocking, which may cause bad ventilation, or even fire, faults or misact.
- Do not online connect, plug or unplug cables, which is apt to cause electric shock or damage the circuit. Installation and wire connection must be firm and reliable. Poor connection could cause misact.
- Use shielded twisted pair for the I/O of high frequency signal and analog signal to improve system IMS.

The installation environment should be free from dust, oil smoke, conductive particle, corrosive or flammable gases, high temperature, condensation, and rain.

Besides, vibration and impact also affect the Touchberry 7" & Tinkertouch 7" normal operation and shorten its lifespan. It is recommended to install the Panel PC, together with the matching switches and contactors, in a dedicated electric cabinet and keep the cabinet ventilated. If the location has high ambient temperature or heat generating equipment nearby, install forced convection devices on top or sides of the cabinet to avoid over-temperature. During drilling or wiring, prevent the metal particles or wire segments from falling into the PLC casing, which may cause fire, fault or misact. After the Panel PC installation, clean the ventilation duct to prevent blocking, which may cause bad ventilation, or even fire, faults or misact.

The only way to disconnect the equipment from the electrical network is by removing the connectors that feed the equipment. Once installed in the electrical cabinet it is very important to ensure the power connectors for proper operation.

Separate the Touchberry 7" & Tinkertouch 7" from heat, high voltage and electrical noise:

Always separate the devices that generate high voltage and high electrical noise from the Touchberry 7" & Tinkertouch 7". When configuring the layout of the Panel PC inside your electrical box, consider the heat-generating devices and locate the electronic-type devices in the cooler areas of your cabinet. Reducing the exposure to a high-temperature environment will extend the operating life of any electronic device. Consider also the routing of the wiring for the devices in the electric cabinet. Avoid placing low-voltage signal wires and communications cables in the same tray with AC power wiring and high-energy, rapidly-switched DC wiring.

Provide adequate clearance for cooling and wiring Panel PC. Is designed for natural convection cooling. For proper cooling, you must provide a clearance of at least 25 cm above and below the devices. Also, allow at least 25 cm of depth between the front of the modules and the inside of the enclosure.

Notes for maintenance:

A well-planned and executed maintenance program is essential to the satisfactory operation of solid-state electrical equipment. The kind and frequency of the maintenance operation will vary with the kind and complexity of the equipment as well as with the nature of the operating conditions. Maintenance recommendations of the manufacturer or appropriate product standards should be followed.

The following factors should be considered when formulating a maintenance program:

- Maintenance must be performed by qualified personnel familiar 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.
- Care should be taken when servicing electrostatic sensitive components. The manufacturer's recommendations for these components should be followed.
- Ventilation passages should be kept open. If the equipment depends upon auxiliary cooling, e.g., air, water, or oil, periodic inspection (with filter replacement when necessary) should be made of these systems.
- The means employed for grounding or insulating the equipment from ground should be checked to assure its integrity.
- Accumulations of dust and dirt on all parts, including on semiconductor heat sinks, should be removed according to the manufacturer's instructions, if provided; otherwise, the manufacturer should be consulted. Care must be taken to avoid damaging any delicate components and to avoid displacing dust, dirt, or debris in a way that permits it to enter or settle into parts of the control equipment.
- Enclosures should be inspected for evidence of deterioration. Accumulated dust and dirt should be removed from the top of the enclosures before opening doors or removing covers.
- Certain hazardous materials removed as part of maintenance or repair procedure (e.g., polychlorinated biphenyls (PCBs) found in some liquidfilled capacitors) must be disposed of as described in Federal regulations.

Safety rules for maintenance personnel

Consider the following steps to follow. A false manoeuvre could be the cause of an accident or material damage.

Do not disassemble or modify the modules. This could lead to breakdowns or malfunctions and could lead to injuries or fire.

- All types of radio communication devices, including mobile phones and personal handy-phone systems (PHS), must be kept more than **25cm** away from the Panel PC in all directions. Failure to observe this precaution exposes malfunctions caused by excess of temperature.
- Disconnect the external power supply of the system (on all phases) before connecting or disconnecting a module. Failure to observe this precaution may cause faults or malfunctions of the module.

- Tighten the screws of the terminal ports and the screws of the connectors within the prescribed tightening torque. Insufficient tightening can lead to loose parts or wires and cause malfunctions. Excessive tightening can damage the screws and / or the module, with the risk of falling, short circuits and malfunctions.
- Before handling a module, dispose of the electrostatic charge accumulated by the human body by touching a suitable conductive object. Failure to observe this precaution may cause faults or malfunctions of the module.

Repair note:

If the equipment is suitable to be repaired, it must be verified that the equipment remains in a safe state after repair.

9 Revision Table

Revision Number	Date	Changes
0	15/09/2019	First implementation
1	17/12/2019	Touchberry PI 7": The internal Raspberry PI 3B+ has been changed for theRaspberry PI 4B model
2	07/08/2020	Power scheme modification
3	11/02/2021	Changes on the IOs & Fan control
4	03/06/2021	Update on chapter 2.1
5	18/06/2021	Update on chapter 1.1
6	14/12/22	Update on chapter 2.2

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