User manual

NAMINO

1. Introduction

1.1. Reference standards

The **NAMINO** device is designed and manufactured in compliance with current legislation for the CE marking for devices that can be used in industrial and residential environments according to the:

EN 55032:2015 /AC:2016 /A11:2020 /A1:2020

EN 55035:2017 /A1:2020

2. Product description

Namino is the first device powered with ESP 32 that faces the **maker** and **industrial** worlds at the **same time**.

Namino is presented on the maker side with connectors for Arduino Uno compatible shields and on the industrial side with interfaces compatible with the signals that can be used in automation.

Namino is the new range of electronic boards that integrates a large number of configurable interfaces and power supply modes (even 100Vac÷250Vac) to simplify and speed up any electronic project.

Something about Namino:

Programming can be done by means of Arduino/ESP32 and VS code (PlatformIO)tools. See Software section.

Each board has a connector compatible with the Arduino Uno shields

Extremely compact form-factor: it can be inserted in a common 503 flush-mounting box 3 modules for home automation applications.

The widest range of possible power supplies: from 5Vdc÷30Vdc and from 100Vac÷250Vac. See Namino Boards section.

In summary:

Namino can be used in the most challenging application.

Namino simplifies wiring by 83% and eliminates the annoying "tangles" of wires typical of early applications or prototypes.

Sensors and devices belonging to both the industrial and amateur world can be easily used for the development of new ideas.

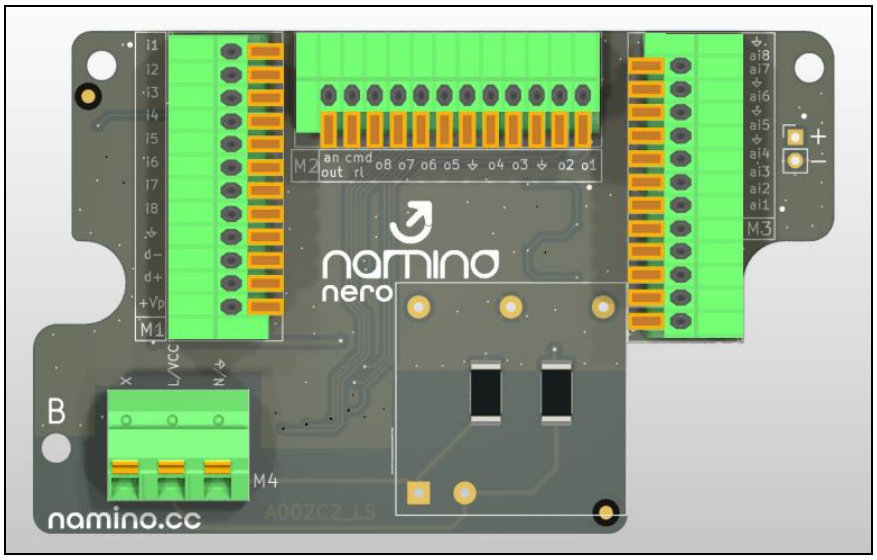
Namino boards has been designed and built directly by Mect Srl: an Italian manufacturer of PLC and products for industrial automation with over 40 years of history.

Namino boards:

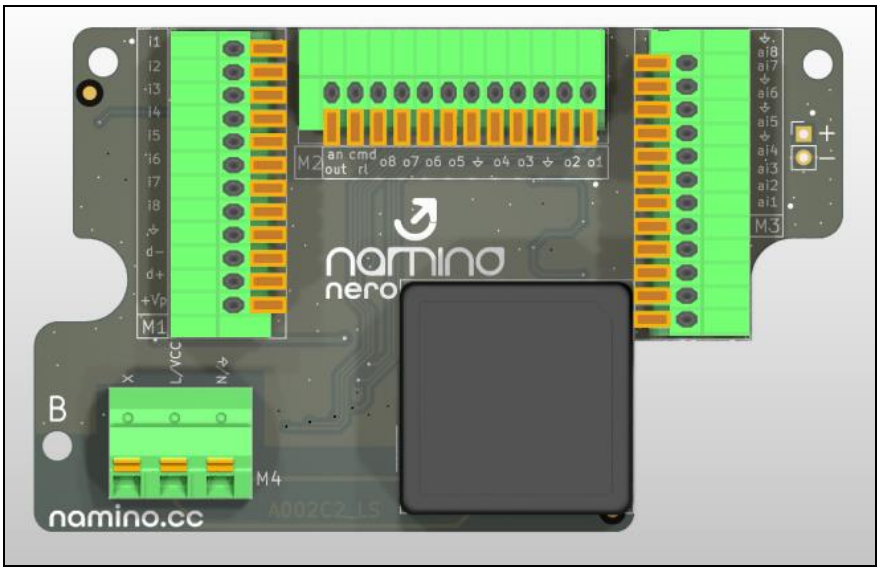
- NERO VDC
- NERO VAC
- ROSSO
- ARANCIO

The following images show the boards:

Namino NERO VDC



Namino NERO VAC



Namino ROSSO

Namino ARANCIO

2.1. System composition

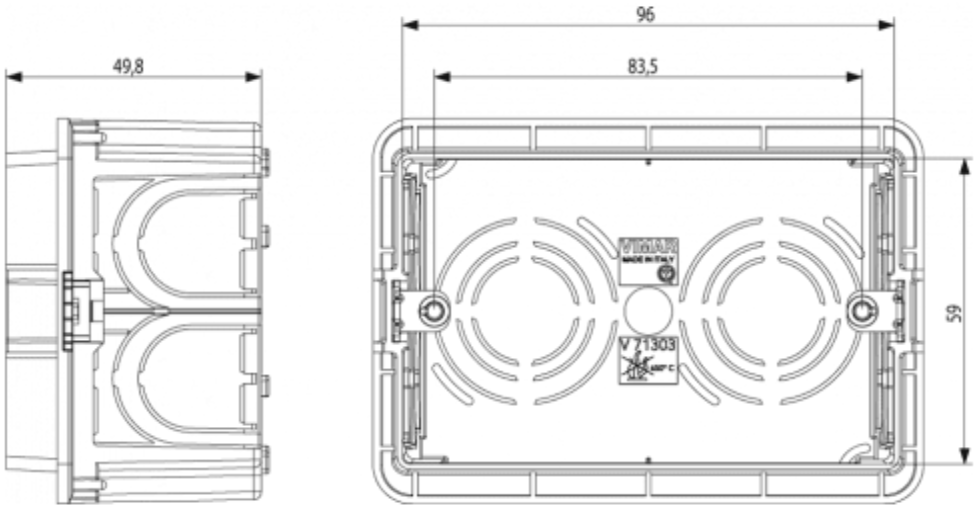
The NAMINO system boards can be assembled as follows:

1. ARANCIO
2. ROSSO
3. ARANCIO + NERO (VDC or VAC)
4. ROSSO + NERO (VDC or VAC)

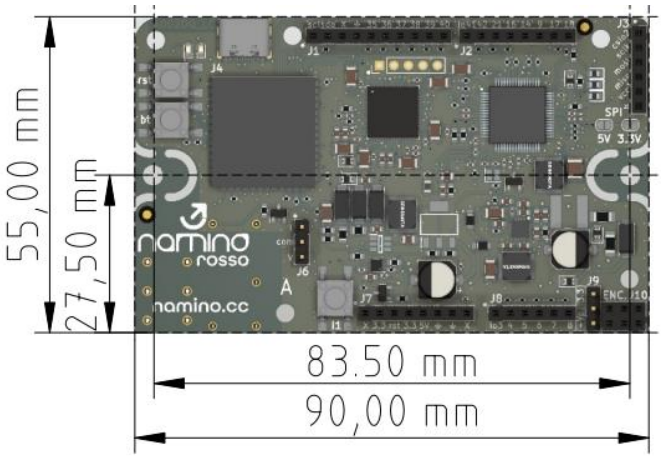
As can be seen, NERO can only be used in combination with the other boards and cannot live separately.

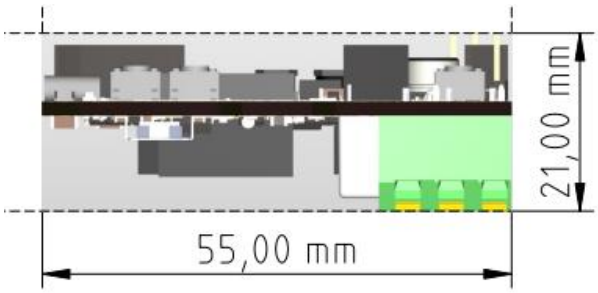
2.2. Dimensions

Namino can be inserted in a common 503 flush-mounting box 3 modules for home automation.

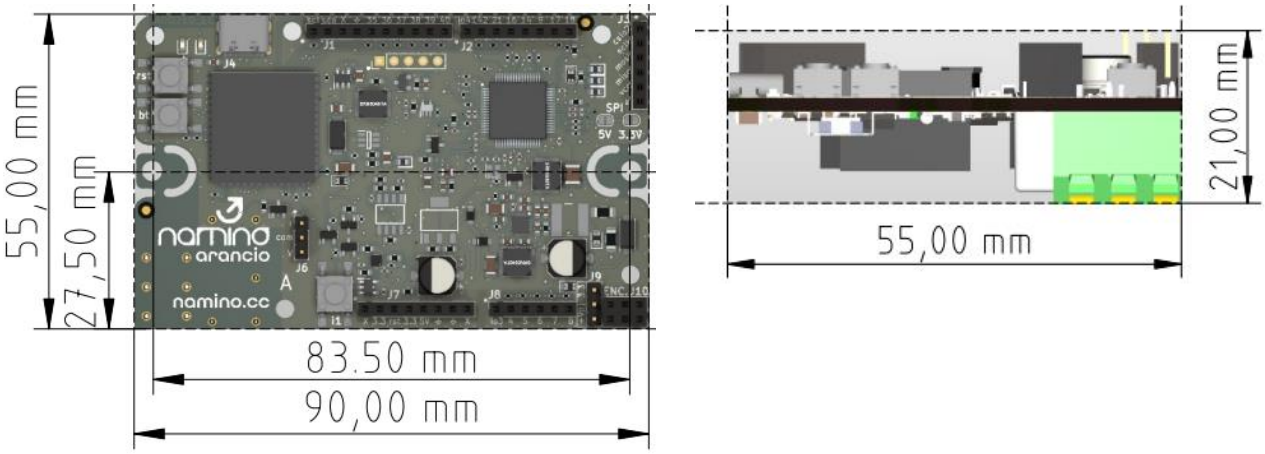


Namino ROSSO

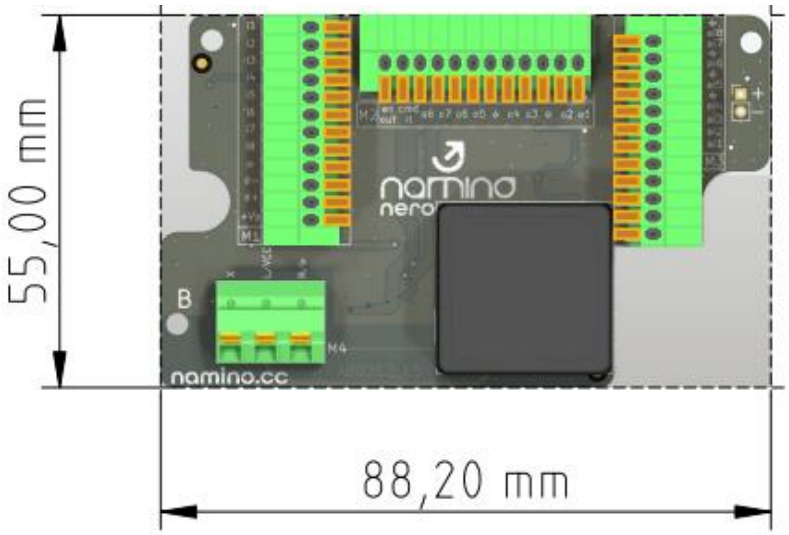




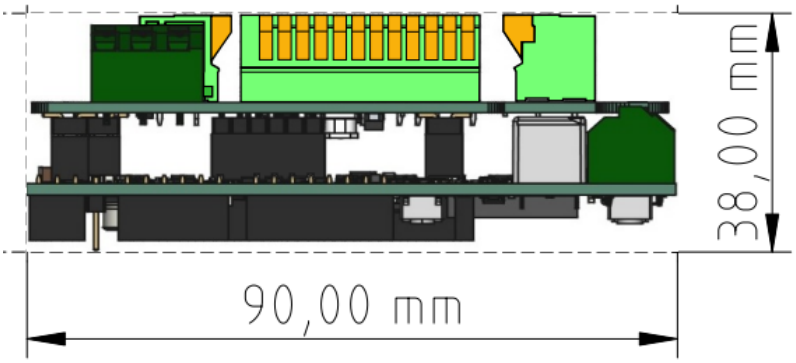
Namino ARANCIO



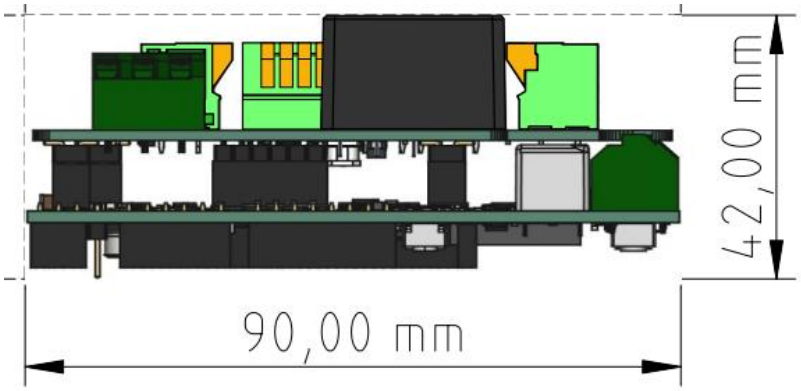
Namino NERO VDC and VAC



Namino ROSSO/ARANCIO and NERO VDC



Namino ROSSO/ARANCIO and NERO VAC



2.3. Generic power supplies

NAMINO can be powered by various sources:

- 5Vdc via USB
- 7Vdc ÷ 30Vdc
- Adding Namino NERO board:
    - Namino NERO VAC: 100Vac ÷ 250Vac
    - Namino NERO VDC: 7Vdc ÷ 30Vdc
    - Rechargeable 18650 lithium battery for power backup (holder and battery not included)

3. Hardware

3.1. CPUs and memories

- ESP32-S3-WROOM-1U-N4R8 (ESP32-S3 series of embedded SoCs, Xtensa®dual-core 32-bit LX7 microprocessor, up to 240MHz)
- 4MB Flash
- 512KB SRAM
- 8MB PSRAM
- 16KB SRAM in RTC

3.2. Input / output ROSSO

NAMINO ROSSO provides the following input and output lines:

- 8 PNP digital inputs (up to 24Vdc)
- 1 knob input and button
- 8 PNP digital outputs (up to 24Vdc) of which 1 for stepper motor command with total current 200mA
- 1 SPDT relay output of 5A
- 8 single ended analog inputs (0-10V or 0-20mA) or 4 thermometric (NTC, PTC, PT1000) or 4 thermocouples (J, K, T) or 2 load cells, software configurable
- 1 analog output jumper configurable: 0-10V or 0-20mA
- ARDUINO UNO compatible shield interface

In addition to the lines present on the ARDUINO UNO interface, there are the following communication interfaces:

- RS485
- Wi-Fi (802.11 b/g/n, Bit rate: 802.11n up to 150 Mbps) - need antenna (not included)
- Bluetooth LE: Bluetooth 5, Bluetooth mesh - need antenna (not included)
- Wi-fi / bluetooth antenna connector
- USB C device for programming
- SPI interface (SD card module not included)
- RTC (real time clock / calendar) equipped with a rechargeable buffer battery
- I^2C
- UART

NOTE: WiFi and BT shares the same antenna.

3.3. Input / output ARANCIO

NAMINO ARANCIO provides the following input and output lines:

- 8 PNP digital inputs (up to 24Vdc)
- 1 knob input and button
- 8 PNP digital outputs (up to 24Vdc) of which 1 for stepper motor command with total current 200mA
- 1 SPDT relay output of 5A
- 4 analog inputs (0-10V or 0-20mA) software configurable
- 1 input for NTC (10kΩ)
- 1 analog output jumper configurable: 0-10V or 0-20mA
- ARDUINO UNO compatible shield interface

In addition to the lines present on the ARDUINO UNO interface, there are the following communication interfaces:

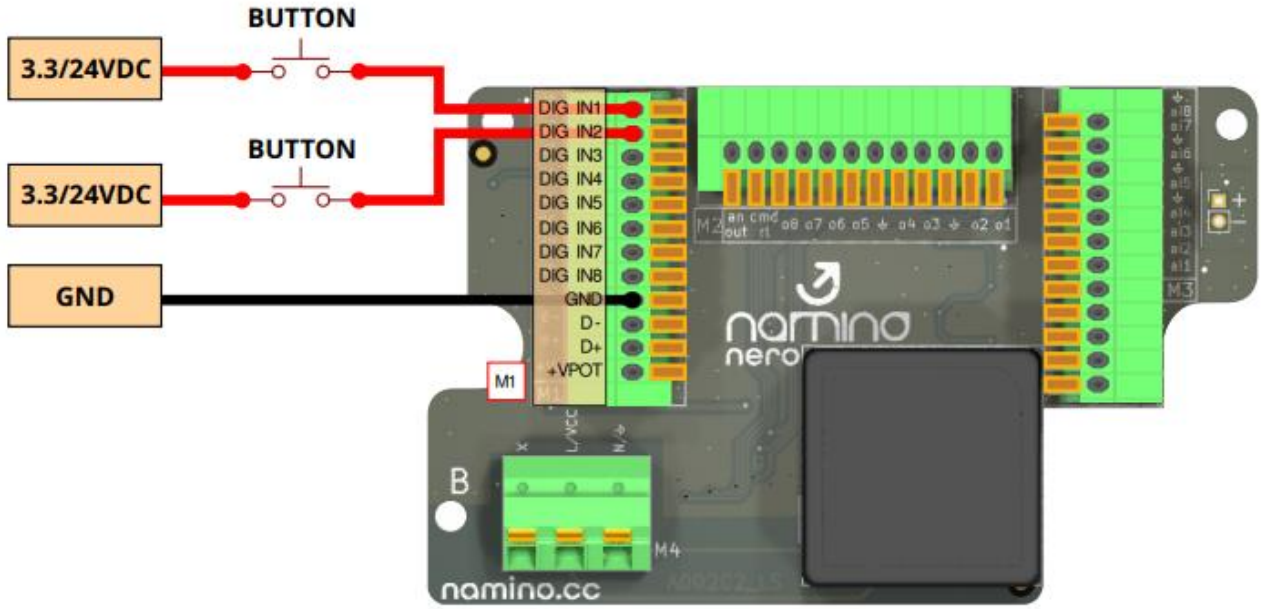
- RS485
- Wi-Fi (802.11 b/g/n, Bit rate: 802.11n up to 150 Mbps) - need antenna (not included)
- Bluetooth LE: Bluetooth 5, Bluetooth mesh - need antenna (not included)
- Wi-fi / bluetooth antenna connector
- USB C device for programming
- SPI interface (SD card module not included)
- RTC (real time clock / calendar) equipped with a rechargeable buffer battery
- I^2C
- UART

NOTE: WiFi and BT shares the same antenna

3.4. Wiring Inputs

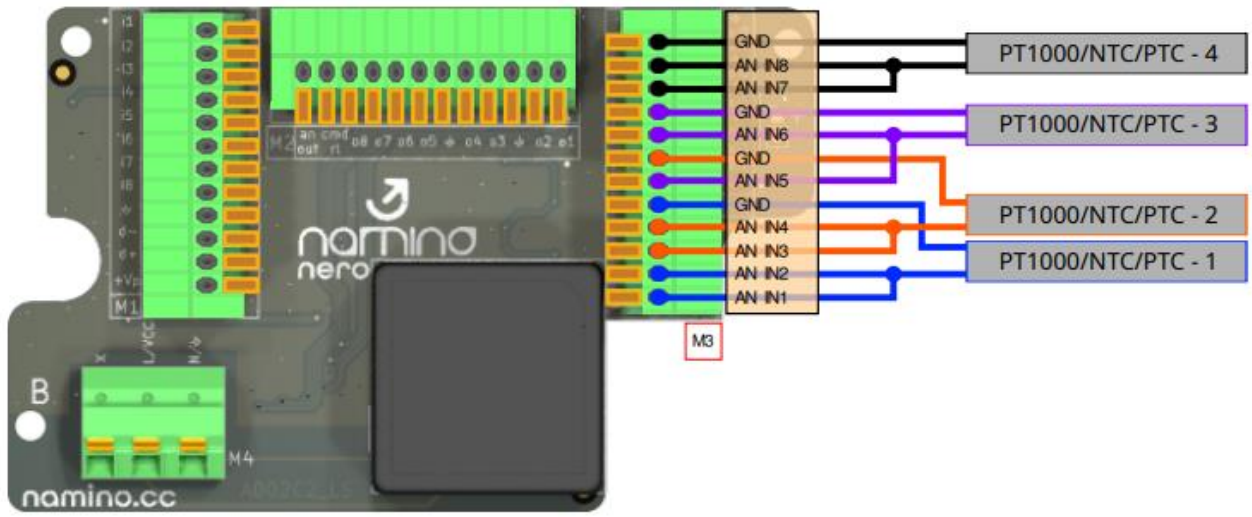
Connection examples are presented using the Namino NERO board which simplifies the connection of the industrial side of Namino. If you do not want to use the Namino NERO board, the same signals are present on side B (ROSSO - ARANCIO) of the relevant boards.

3.4.1. Digital input – PNP



3.4.2. PT1000 – NTC – PTC (ONLY for Namino ROSSO)

Use only 2-wire probes.



Namino ARANCIO: for the NTC connection see the "Input / output signals" chapter.

Software information:

- Probe 1 must be read in logical channel 1
- Probe 2 must be read in logical channel 2

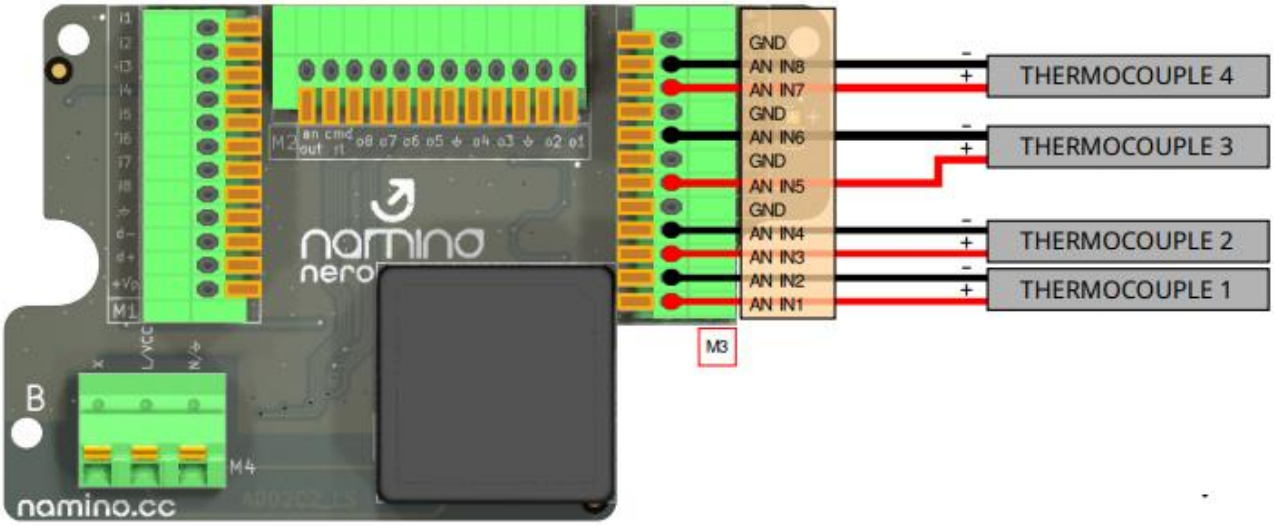
- Probe 3 must be read in logical channel 3
- Probe 4 must be read in logical channel 4

3.4.3. Thermocouples J – K – T

Thermocouples type: J, K and T.

Thermocouples must be of the insulated type.

Use only suitable thermocouple cables.

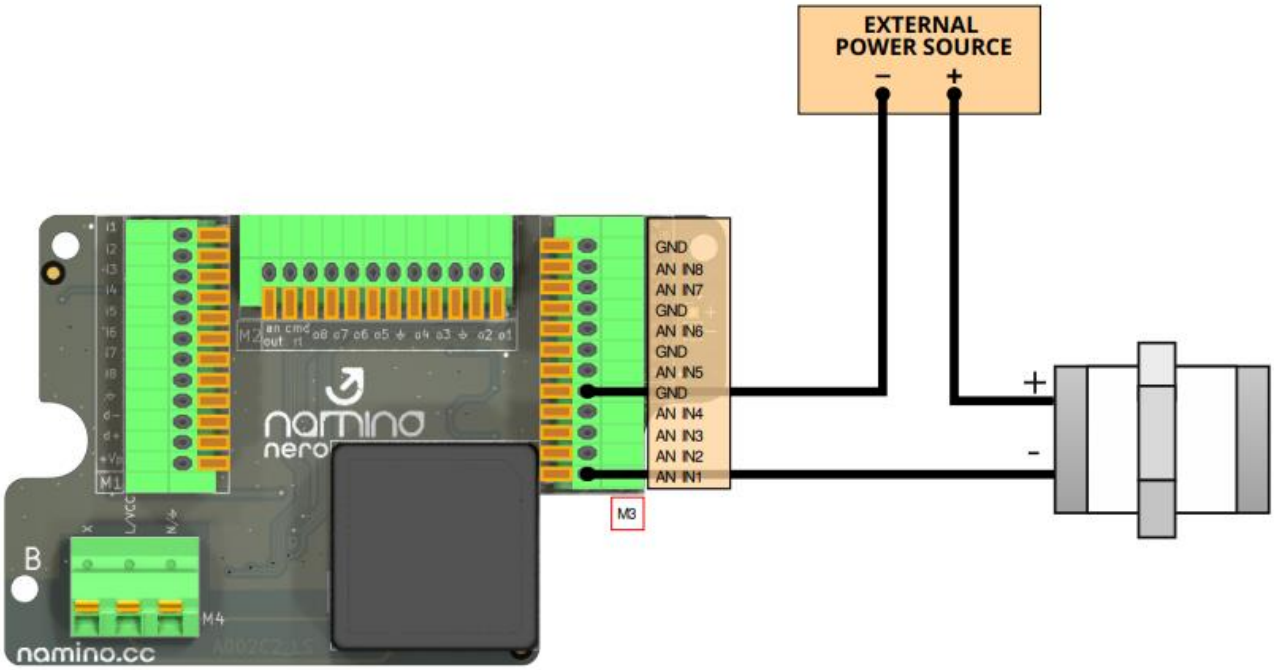


Software information:

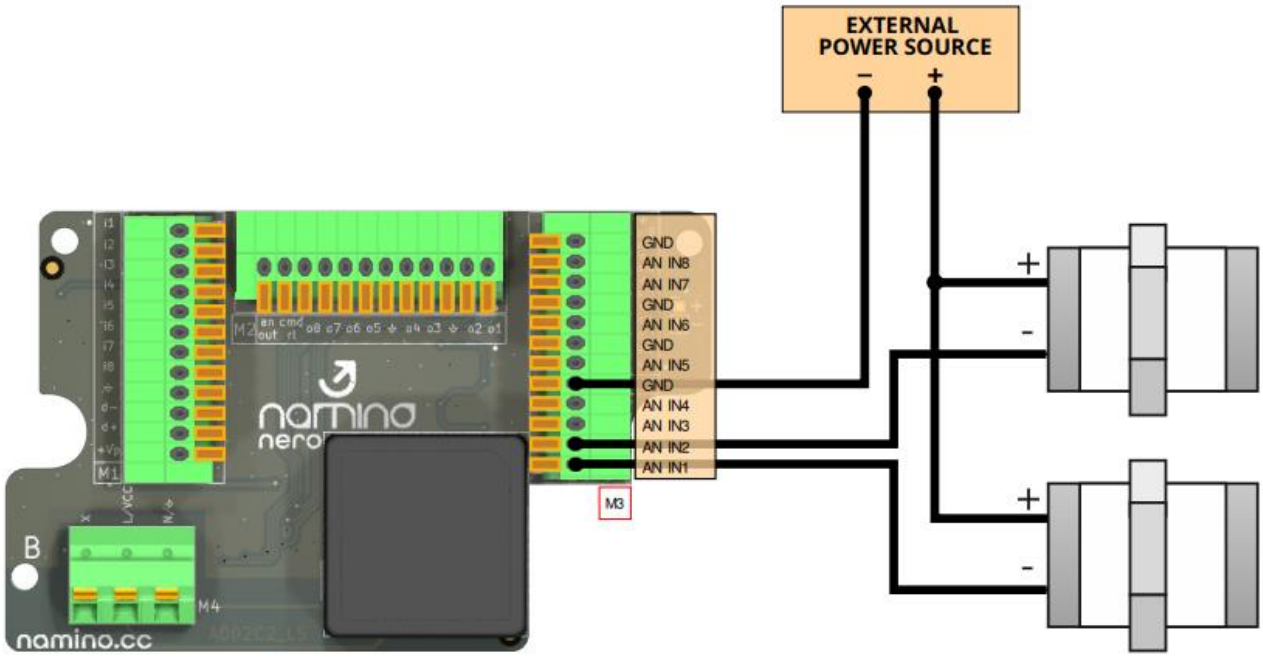
- Probe 1 must be read in logical channel 1
- Probe 2 must be read in logical channel 2
- Probe 3 must be read in logical channel 3
- Probe 4 must be read in logical channel 4

3.4.4. Current transducers

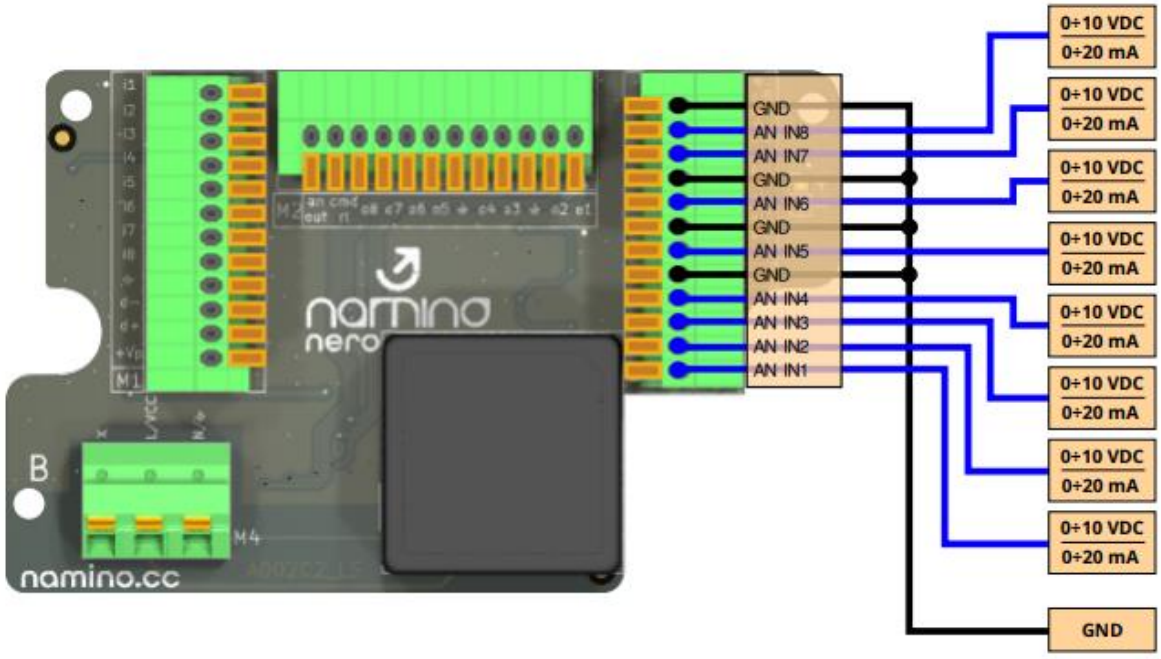
Using a 2-wire current transducer with external power supply:



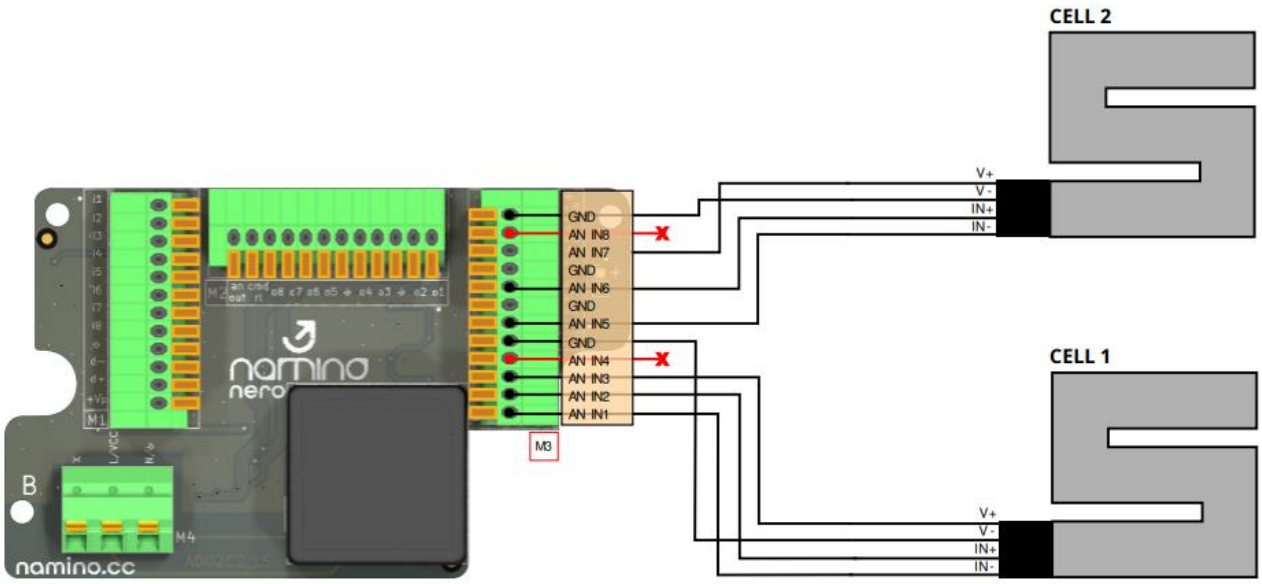
Usage of multiple 2-wire current transducers with external power supply:



3.4.5. Current – Voltage input



3.4.6. Load cells

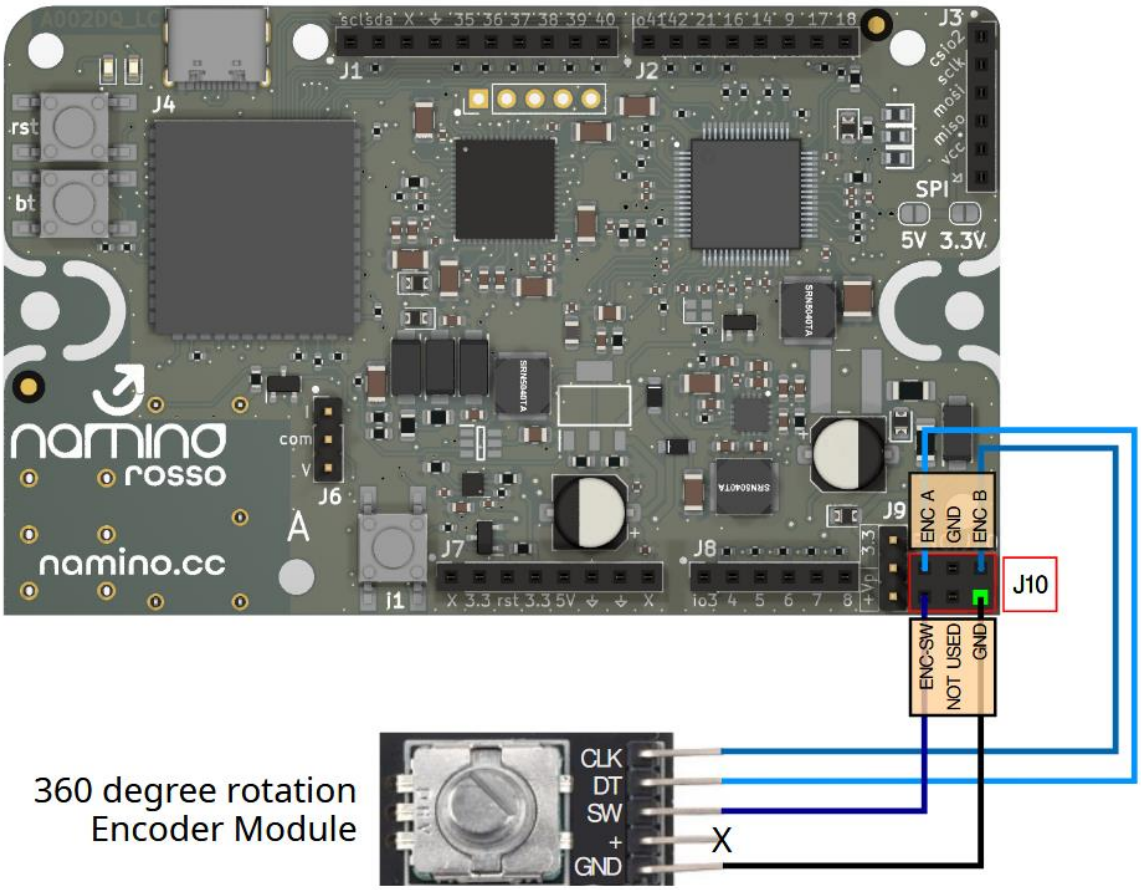


Note: “AN IN4” and “AN IN8” are not usable despite not having the wired pin.

Software information:

- Load cell 1 must be read in logical channel 1
- Load cell 2 must be read in logical channel 2

3.4.7. Encoder NPN

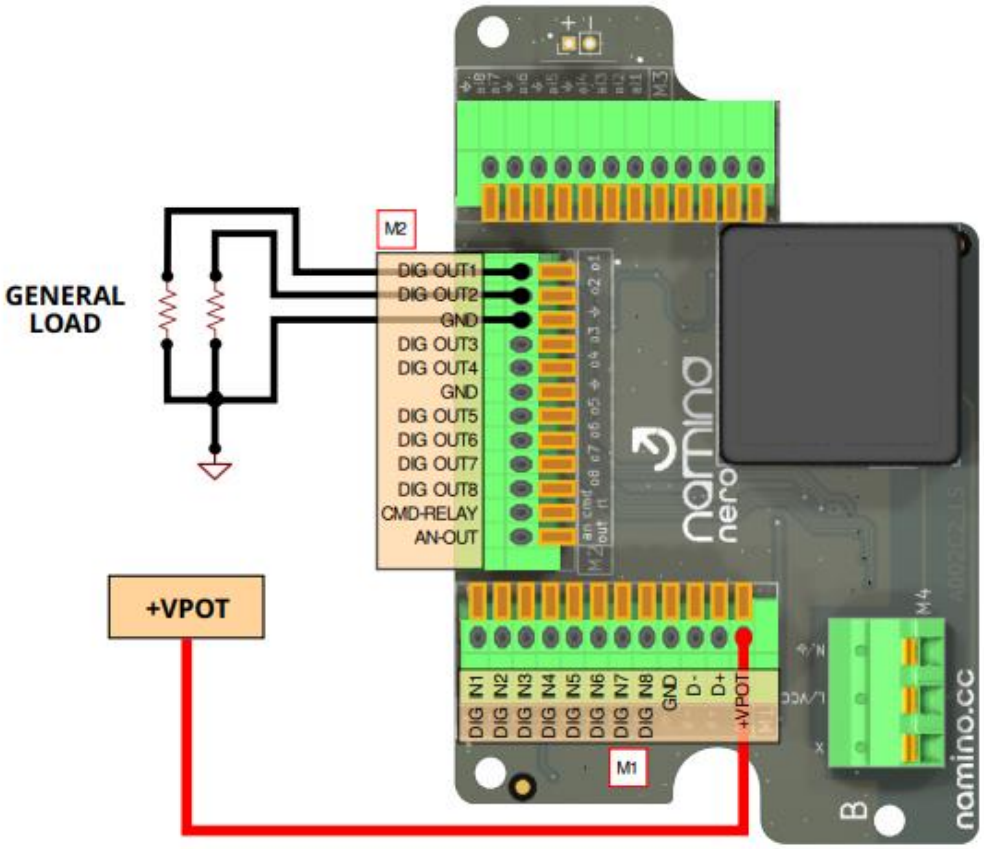


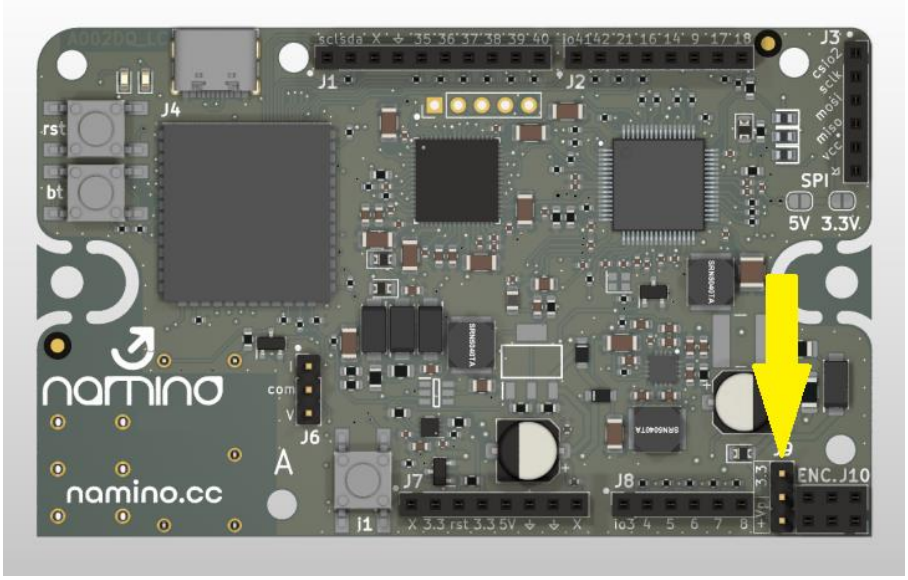
3.5. Wiring Outputs

Connection examples are presented using the Namino NERO board which simplifies the connection of the industrial side of Namino. If you do not want to use the Namino NERO board, the same signals are present on side B (ROSSO - ARANCIO) of the relevant boards.

3.5.1. Digital output – PNP

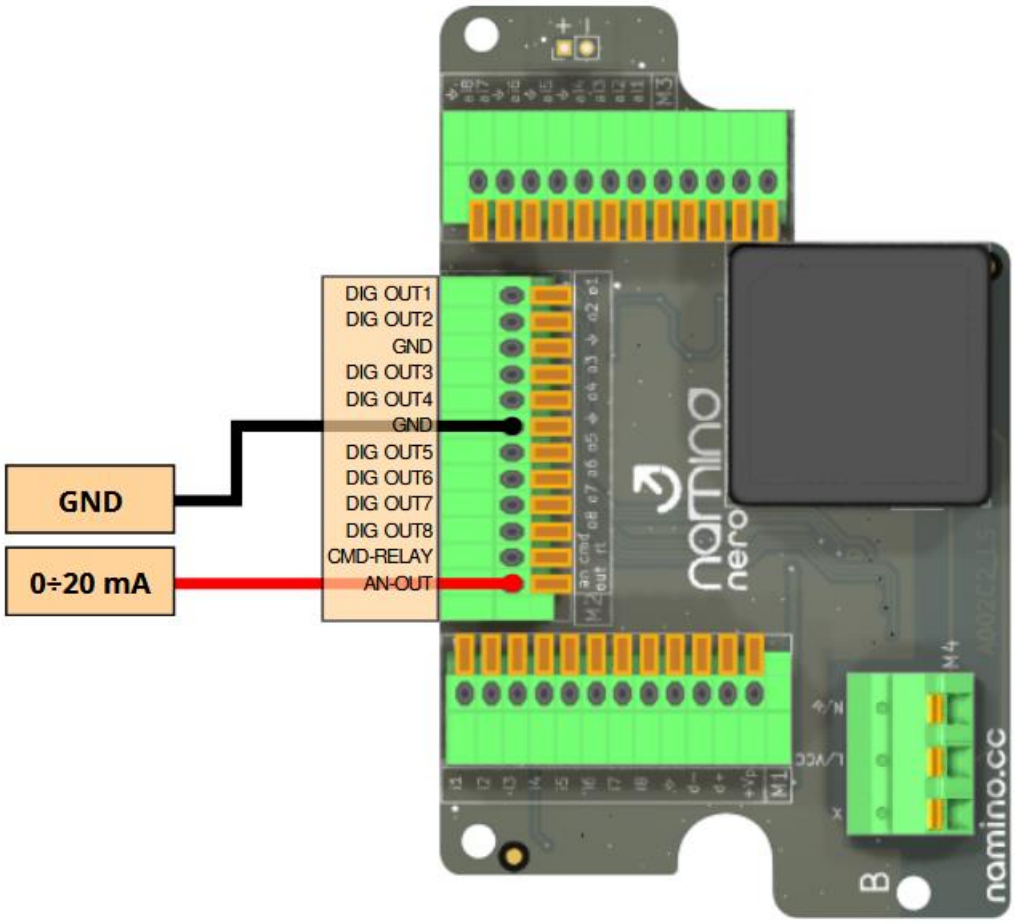
Power up the +VPOT terminal and by the jumper on ROSSO / ARANCIO (side A – J9) select the type of power supply for the outputs, in this case Vpot.

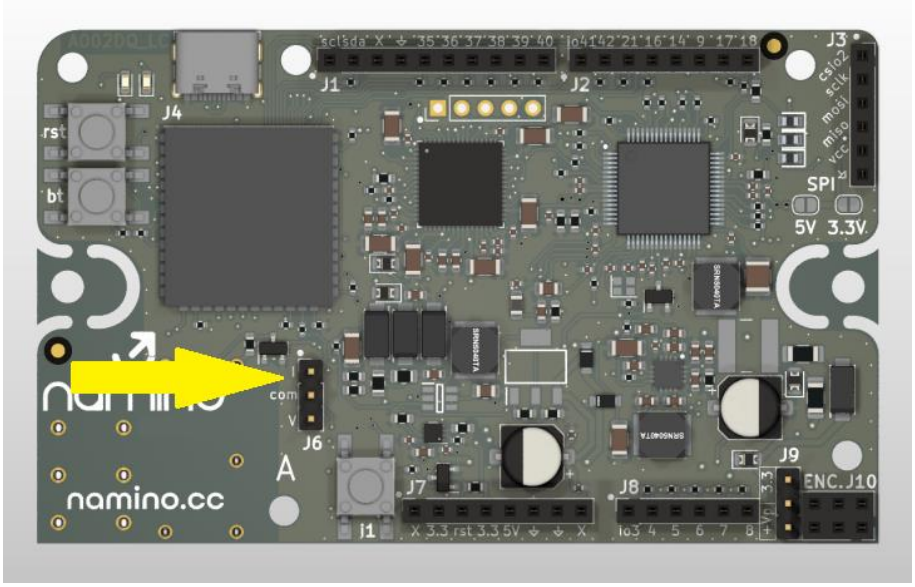




3.5.2. Current output

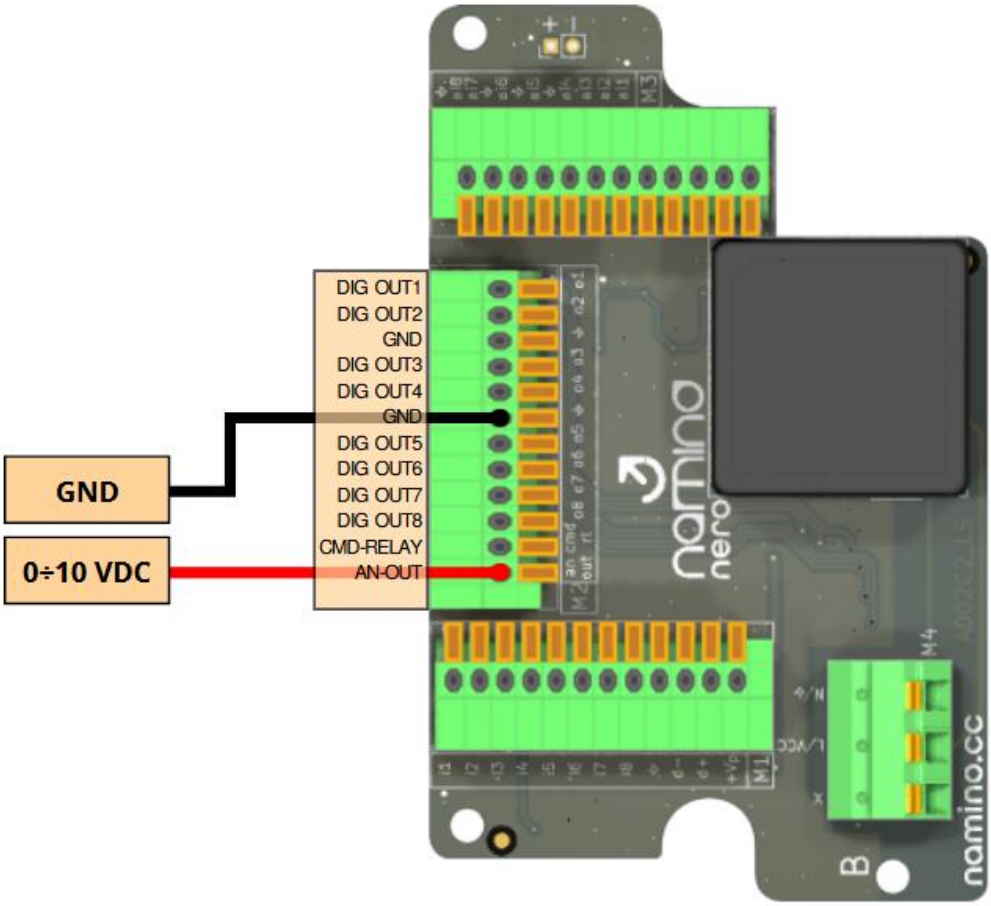
Requires hardware (on ROSSO / ARANCIO (side A – J6)) and software configuration.

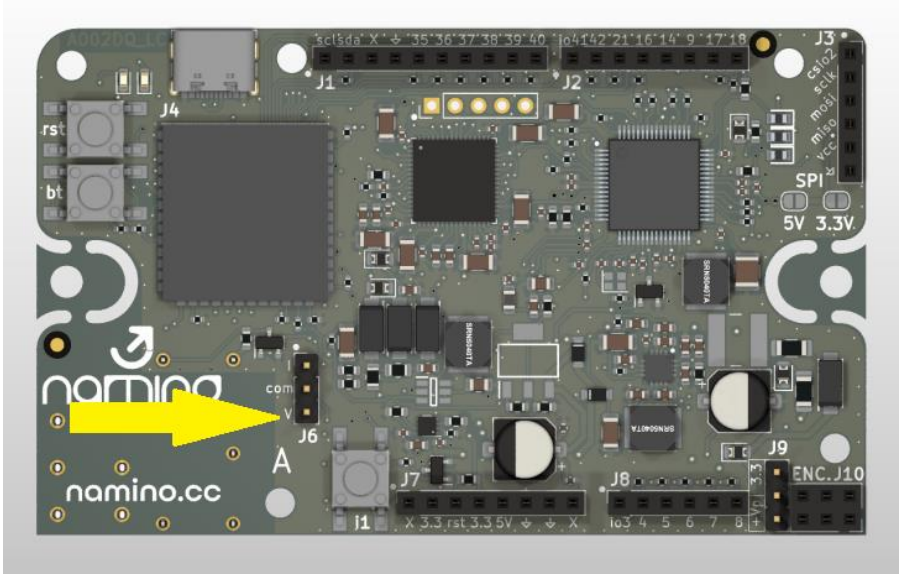




3.5.3. Voltage output

Requires hardware (on ROSSO / ARANCIO (side A – J6)) and software configuration.





4. Boards features

4.1. NERO

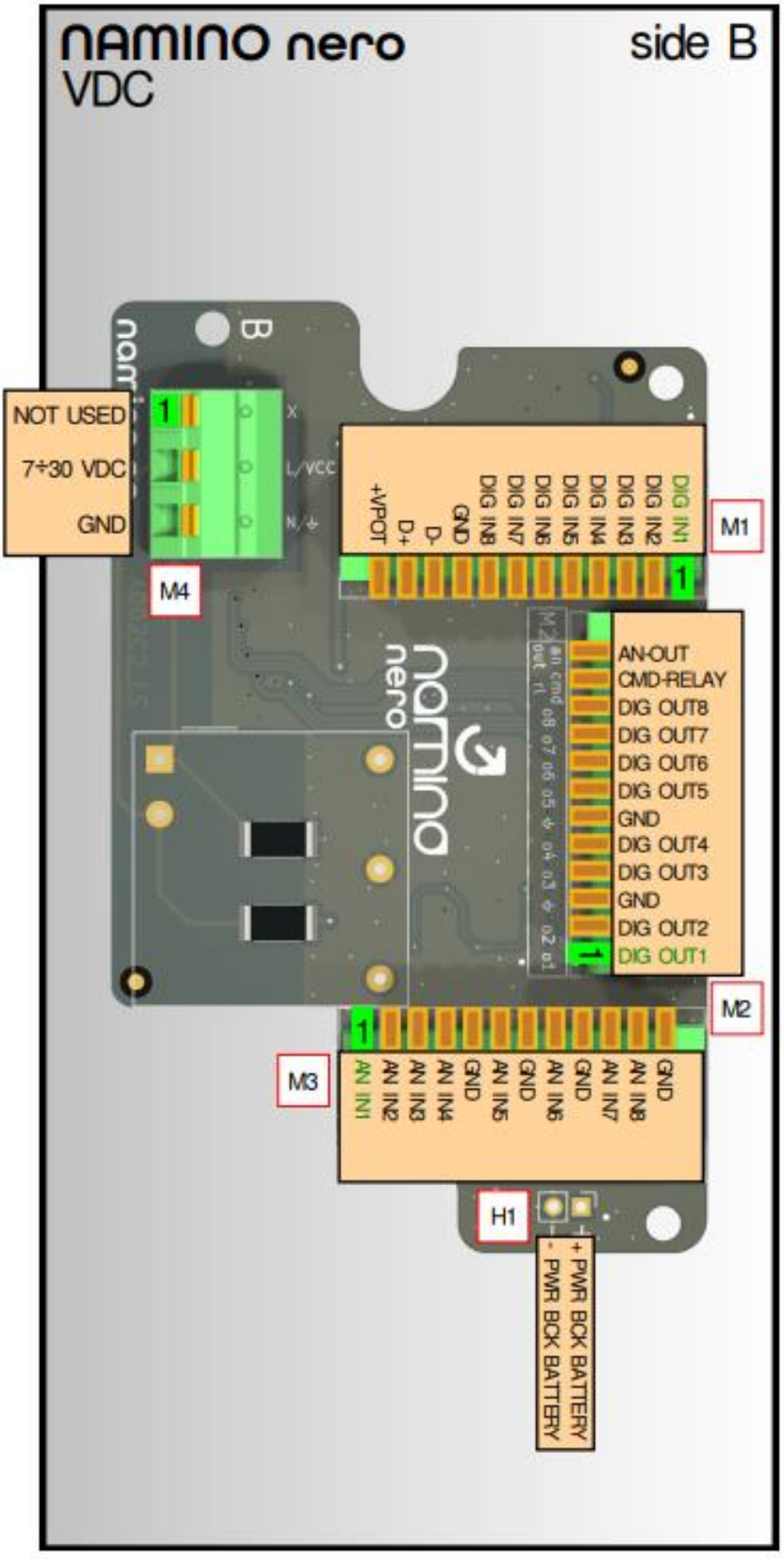
Namino NERO needs to be assembled on the Namino ROSSO/ARANCIO boards to make the signal lines available on the 2.5mm pitch terminal block.

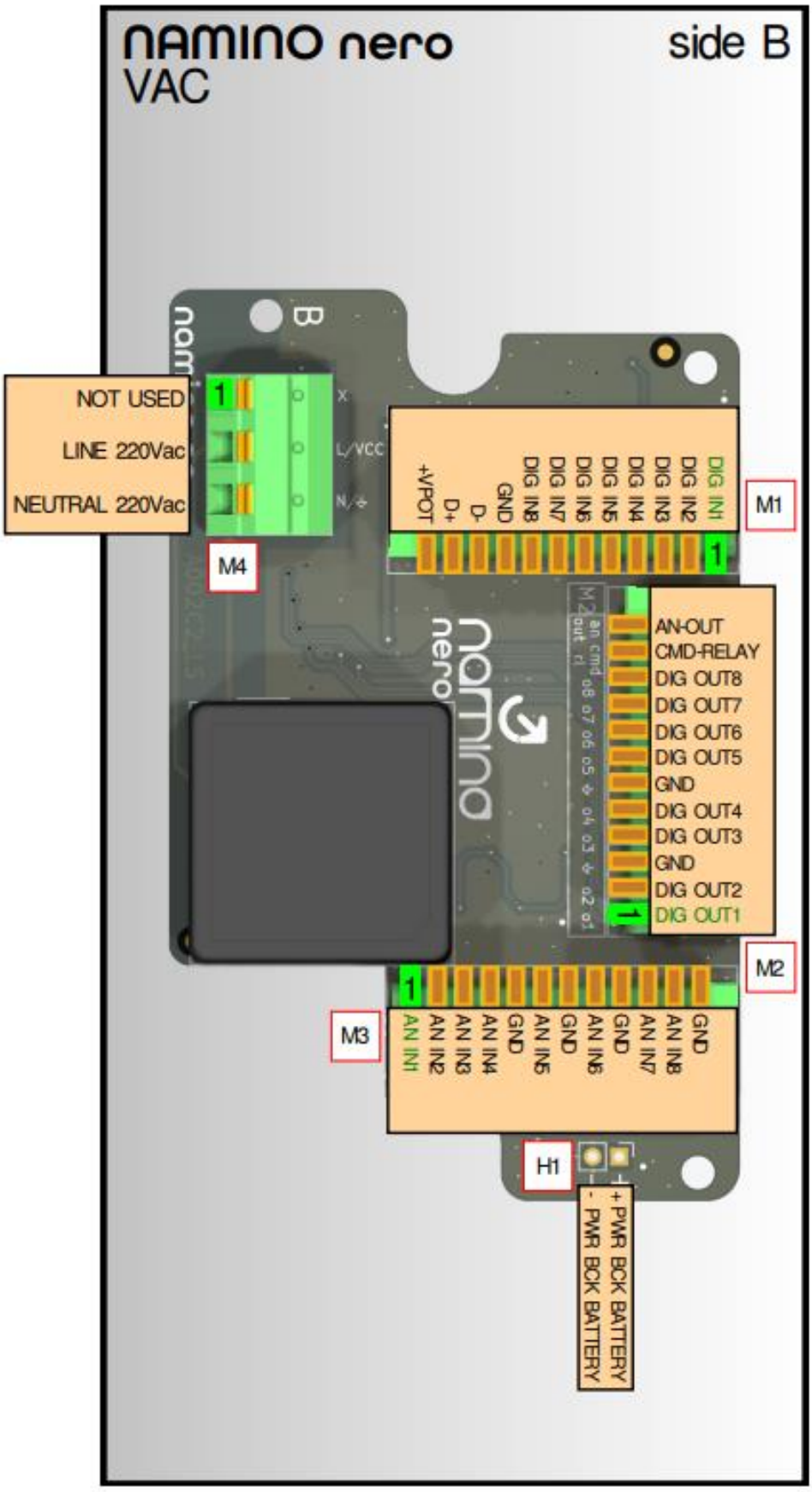
The Namino NERO board provides power supply terminals and industrial signal terminals blocks (field I/O) to facilitate wiring.

It is possible to weld a rechargeable 18650 lithium battery as a power backup, see H1 (holder and battery not included).

There are two versions of Namino NERO which differ in the type of power supply:

- Namino NERO VDC: Power supply in VDC (7Vdc ÷ 30Vdc)
- Namino NERO VAC: Power supply in VAC (100Vac ÷ 250Vac)


namino nero
VDC
side B
NOT USED
7÷30 VDC
GND
M4
M1
+VPOT
D-
D+
GND
DIG IN8
DIG IN7
DIG IN6
DIG IN5
DIG IN4
DIG IN3
DIG IN2
DIG IN1
AN-OUT
CMD-RELAY
DIG OUT8
DIG OUT7
DIG OUT6
DIG OUT5
GND
DIG OUT4
DIG OUT3
GND
DIG OUT2
DIG OUT1
M2
M3
AN IN1
AN IN2
AN IN3
AN IN4
GND
AN IN5
GND
AN IN6
GND
AN IN7
AN IN8
GND
H1
+ PWR BCK BATTERY
- PWR BCK BATTERY


namino nero
VAC
side B
NOT USED
LINE 220Vac
NEUTRAL 220Vac
M4
M1
M2
M3
H1
AN-OUT
CMD-RELAY
DIG OUT8
DIG OUT7
DIG OUT6
DIG OUT5
GND
DIG OUT4
DIG OUT3
GND
DIG OUT2
DIG OUT1
+ PWR BCK BATTERY
- PWR BCK BATTERY

4.1.1. Input / output signals

The following signals can be accessed from the Namino NERO board:

Line Name	Position	Function	Note
DIG IN1	M1 – 1	Digital input 1 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN2	M1 – 2	Digital input 2 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN3	M1 – 3	Digital input 3 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN4	M1 – 4	Digital input 4 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN5	M1 – 5	Digital input 5 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN6	M1 – 6	Digital input 6 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN7	M1 – 7	Digital input 7 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
DIG IN8	M1 – 8	Digital input 8 - PNP	Logic 1 = min. 3÷24VDC Logic 0 = max 0÷1VDC
GND	M1 – 9	Ground	
D-	M1 – 10	Line D- RS485	
D+	M1 – 11	Line D+ RS485	
+Vpot	M1 – 12	Digital output power supply voltage	V max = 24VDC JUMPER configuration needs on ROSSO / ARANCIO (side A – J9)
DIG OUT 1	M2 – 1	Static digital output 1 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
DIG OUT 2	M2 – 2	Static digital output 2 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
GND	M2 – 3	Ground	
DIG OUT 3	M2 – 4	Static digital output 3 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
DIG OUT 4	M2 – 5	Static digital output 4 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
GND	M2 – 6	Ground	
DIG OUT 5	M2 – 7	Static digital output 5 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
DIG OUT 6	M2 – 8	Static digital output 6 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
DIG OUT 7	M2 – 9	Static digital output 7 - PNP	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (max 24VDC)
DIG OUT 8	M2 – 10	Static digital output 8 - PNP or Step output for step-by-step command	Output voltage 3.3VDC (internally generated) or voltage corresponding to +Vpot supplied externally (M1 -12 pin, max 24VDC) Alternatively it is possible to supply a PTO (pulse train output) according to the frequency value set in register 39 (STEP

			FREQ) If STEP FREQ = 0 => static output If STEP FREQ > 0 => PTO
CMD-RELAY	M2 – 11	Command for external relay	Command to the coil of an external relay (5V)
AN-OUT	M2 – 12	Analog output	Current (0-20mA) or voltage (0 -10V) output. Configuration must be done both: - by JUMPER on ROSSO / ARANCIO (side A – J6) - by software
AN IN 1	M3 – 1	Analog input 1 - single ended	*** See description below**
AN IN 2	M3 – 2	Analog input 2 - single ended	*** See description below**
AN IN 3	M3 – 3	Analog input 3 - single ended	*** See description below**
AN IN 4	M3 – 4	Analog input 4 - single ended	*** See description below**
GND	M3 – 5	Ground	
AN IN 5	M3 – 6	Analog input 5 - single ended	*** See description below**
GND	M3 – 7	Ground	
AN IN 6	M3 – 8	Analog input 6 - single ended	*** See description below**
GND	M3 – 9	Ground	
AN IN 7	M3 – 10	Analog input 7 - single ended	*** See description below**
AN IN 8	M3 – 11	Analog input 8 - single ended	*** See description below**
GND	M3 – 12	Ground	

***** The use of the analog inputs depends on the board being used: ROSSO or ARANCIO.

In Namino **ROSSO** all the inputs from "**AN IN 1**" to "**AN IN 8**" are available .

The number of usable inputs also depends on the type used, for example in the case of Namino ROSSO you can use 8 current/voltage inputs or 2 load cells or 4 resistance thermometers or thermocouples BUT you can mix the various types of inputs.

Specifically, it must be considered that each input (AN IN 1..8) corresponds to a terminal.

Depending on the type of input, a specific number of terminals must be used for the purpose, as shown in the table:

Type Input	Number of terminals required
0-10V **0-20mA**	1
NTC, PTC, PT1000 **Thermocouples**	2
Load cells	4

This implies that if, for example, you have to connect:

• N. 3 0-10V sensors

• N. 3 Thermocouples

There would not be enough terminals available as the total number is 8.

This because 3 terminals are used by 0-10V sensors and 6 for thermocouples, for a total of 9 terminals. Either one 0-10V sensor or a thermocouple must be omitted.

Another example:

It is possible to connect one load cell (4 terminals) + one 0-10V sensor (1 terminal) + one 0-20mA sensor (1 terminal) + one NTC (1 terminal) for a total of 8 terminals used.

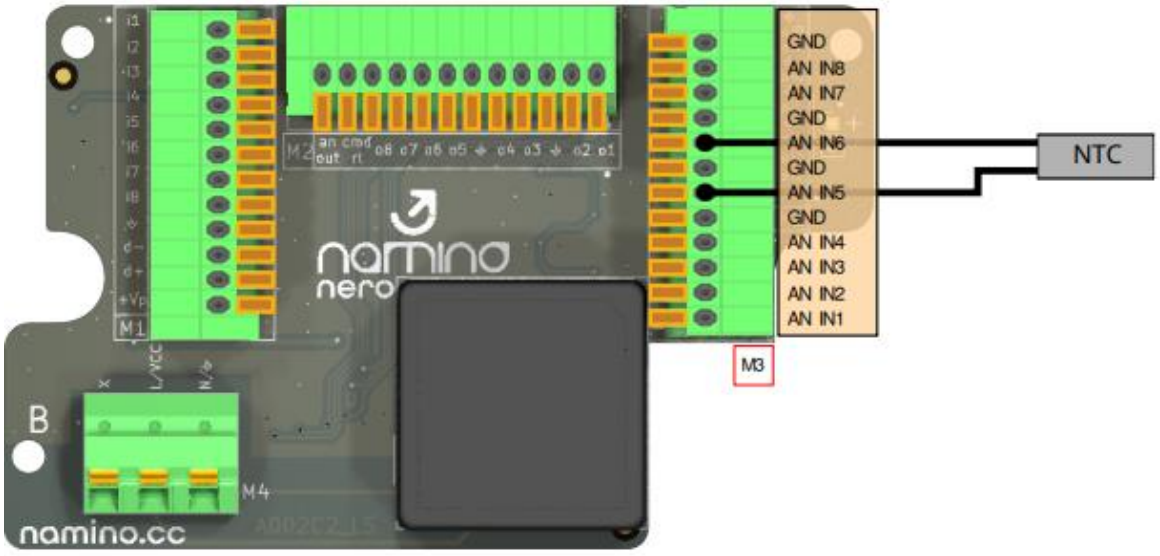
For the connections, see the "Wiring Inputs" chapter.

In Namino **ARANCIO** are available only inputs from "**AN IN 1**" to "**AN IN 6**".

The first four analog inputs from "**AN IN 1**" to "**AN IN 4**" can only be used in 0-10V and 0-20mA.

For connections, see "Current trasducers" and "Current – Voltage input" chapters.

The "**AN IN 5**" to "**AN IN 6**" inputs can be used only to connect an NTC probe, as shown in the following figure:

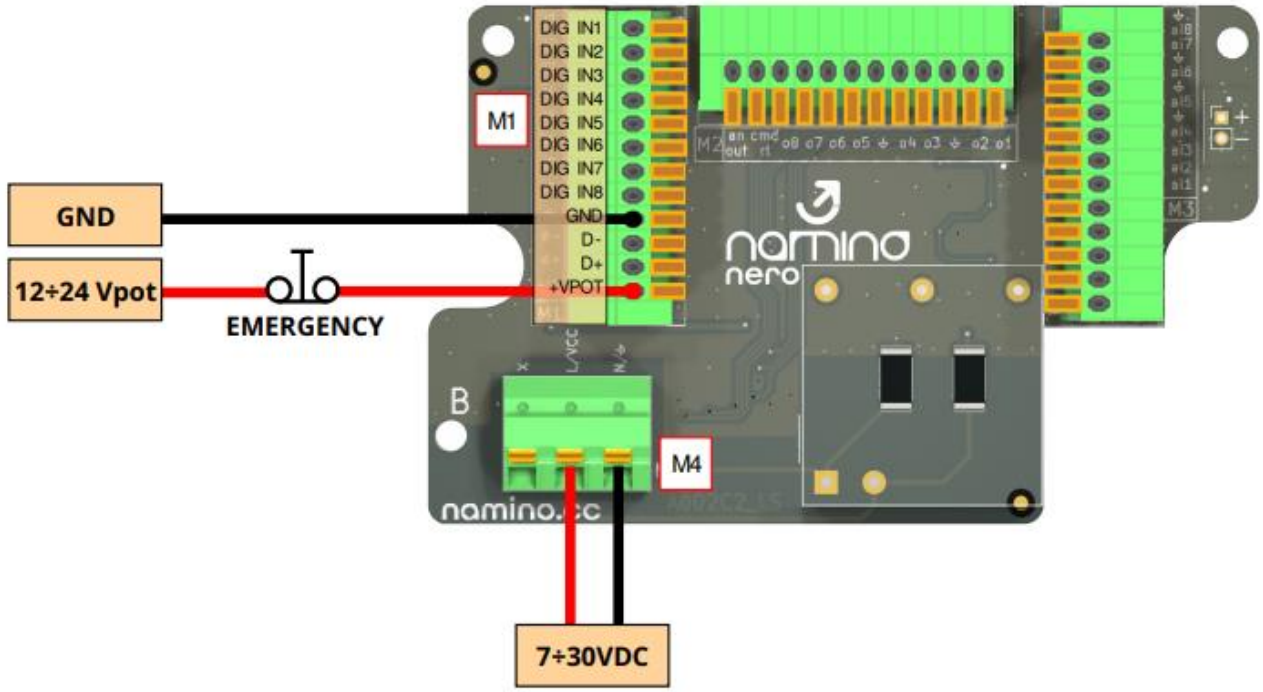


4.1.2. Power supplies

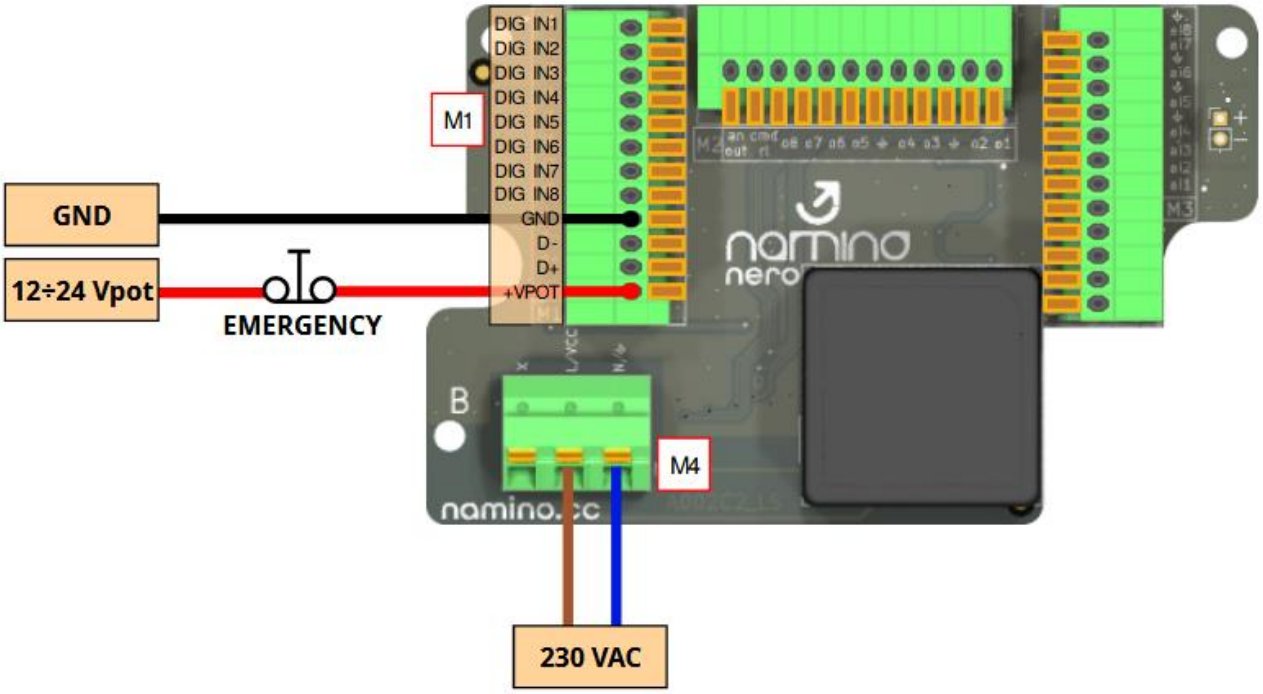
Line Name	Position	Function	Note
X	M4 – 1	NOT USED	
L/VCC	M4 – 2	Line 230VAC or +VDC	NERO VAC = 100VAC ÷ 250VAC power line NERO VDC = 7VDC ÷ 30VDC power line (+)
N/GND	M4 – 3	Neutral or GND	NERO VAC = Neutral NERO VDC = GND
	H1 +	Battery power +	Connection for charging a rechargeable 18650 lithium battery as power backup
	H1 -	Battery power -	Connection for charging a rechargeable 18650 lithium battery as power backup

4.1.2.1. Digital output power supply (+Vpot)

Namino NERO VDC:



Namino NERO VAC:



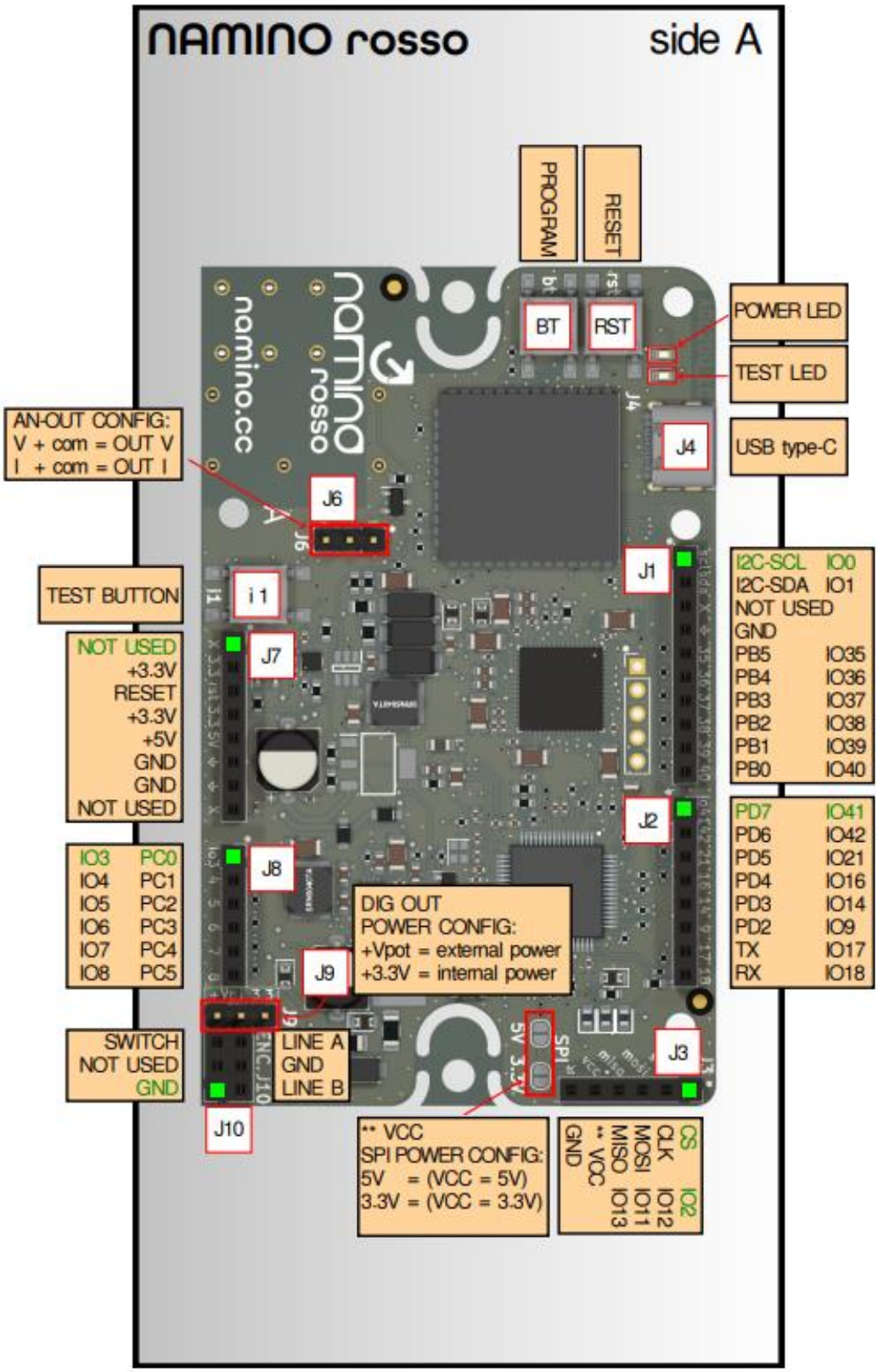
4.2. ROSSO

There are two kind of connections on the board:

- Side A: Connections to compatible ARDUINO UNO shields
- Side B: Connections with field inputs / outputs or connection with Namino NERO VDC or VAC

4.2.1. A-side connections

On side A there is an interface compatible with the ARDUINO UNO shields. The available pins are software configurable. There are also additional interfaces: knob encoder with switch, SPI, UART, I²C and USB C type.



- **J1, J2, J7, J8:** Interface compatible with ARDUINO UNO shields.
  The PBx, PCx and PDx registers/signals inherent to the Arduino Uno interface must be programmed according to the relative **IOx name which corresponds to the ESP32 interface**.
  J1: I²C (I2C-SCL/ I2C-SDA – IO0/IO1). On the I²C bus there is an RTC at the address 0x68.
  J2: UART id 1, serial hardware (TX/RX – IO17/IO18).

- **J3:** SPI interface (requires a soldered jumper to select the power supply). For example, it is possible to connect an external card such as SD-CARD (available for purchase in the accessories section of shop.namino). The IOx signals corresponds to the ESP32 interface.

- **J4**: USB C interface is of type Device and looks like VCOM (Virtual serial COMmunication port). From the USB port it is possible to power up the entire board.

- **J5:** Internal use.

- **J6:** Configuration required via jumper to configure the analog output in voltage [V] or Current [mA]. IMPORTANT: it needs a software configuration too.

- **J9:** Jumper configuration to choose the power voltage of the digital outputs: internal (3.3VDC) or equal to the +Vpot power supply (max 24VDC).

- **J10:** Knob encoder interface with switch or encoder NPN.

- **i1 BUTTON:** Connected in parallel to DIG IN1 (digital input 1) - Side B – J11 (for TEST).

- **RST BUTTON:** ESP32 RESET command (as if the device were turned on again).

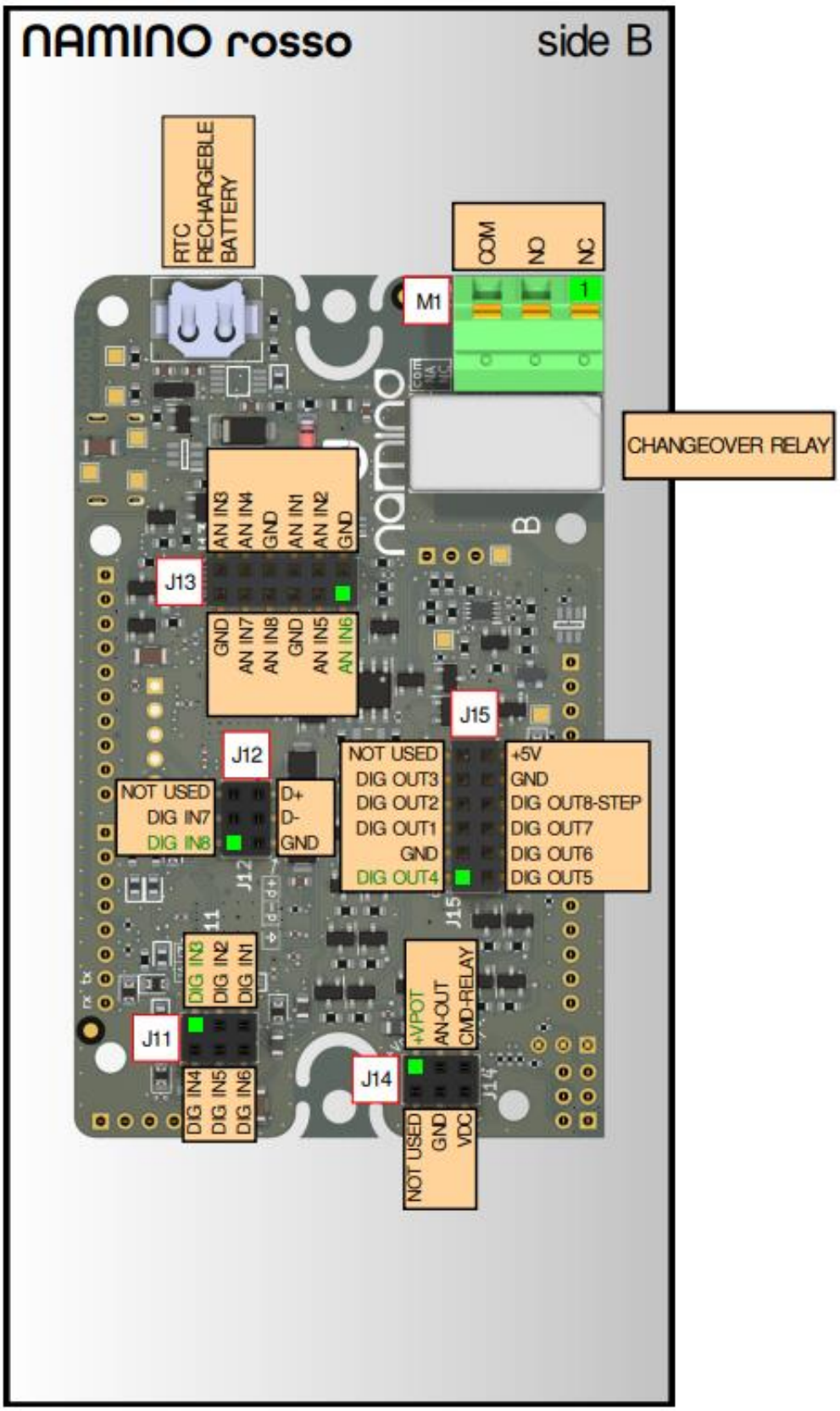
- **BT BUTTON:** Set up the programming mode: press this button to enter in programming mode, then press the **RST** button to exit programming mode.

- **POWER LED:** Indicates that the device is powered.

- **TEST LED:** Connected in parallel to DIG OUT1 (digital output 1) - Side B – J15 (for TEST).

4.2.2. B-side connections

Side B provides the signals of digital and analog inputs / outputs for the field. Optionally, these strip pins goes to the Namino NERO board for greater wiring convenience.



- **J11:** Digital inputs. For more details, see the Input / output signals chapter.
- **J12:** Digital inputs and RS485 signals. For more details, see the Input / output signals chapter.
- **J13:** Analog inputs. For more details, see the Input / output signals chapter.
- **J14:** Board power supply (7-30VDC), digital output power supply (+Vpot max 24VDC), command for an external 5V relay (in parallel with 5A SPDT relay - M1) and the analog output. For more details, see the Input / output signals chapter.
- **J15:** Digital outputs. For more details, see the Input / output signals chapter.
- **M1:** 5A SPDT relay interface. **COM** = COMMON, **NO** = NORMALLY OPEN, **NC** = NORMALLY CLOSED.
- **RTC:** (real time clock / calendar) equipped with a rechargeable buffer battery.

4.2.2.1. I/O features

I/O features					
Analog inputs	N° 8 (AN IN1 – AN IN8)	**Input type**	**Resolution**	**Bit**	**Note**
		0÷20 mA	0.001mA	16	Input impedance 13Ω
		0÷10V	0.001V	16	Input impedance >1MΩ
		Thermocouples J(0°C ÷ 600°C) T(0°C ÷ 400°C) K(0°C ÷ 1200°C)	1°C	16	Cold junction compensation
		PT1000 -40.0°C ÷ 130.0°C	0.1°C	16	
		NTC -40°C ÷ 130°C	0.5°C	16	Beta Value 25/50 K 3950 10kΩ
		PTC -40°C ÷ 130°C	0.5°C	16	1KΩ
		Load cells	TBD	TBD	
Analog output	N° 1 (AN-OUT)	**Output type**	**Resolution**	**Bit**	**Note**
		0÷20 mA	0.02mA	10	Max impedance : 400Ω
		0÷10V	0.01V	10	Min impedance : 1KΩ
Digital inputs	N°8 (DIG IN1 – DIG IN 8)	**Input type**	**Resolution**		**Note**
		PNP	cycle time		
Digital outputs	N°8 (DIG OUT1 - DIG OUT8)	**Output type**	**Resolution**		**Note**
		PNP	cycle time		200mA total

4.3. ARANCIO

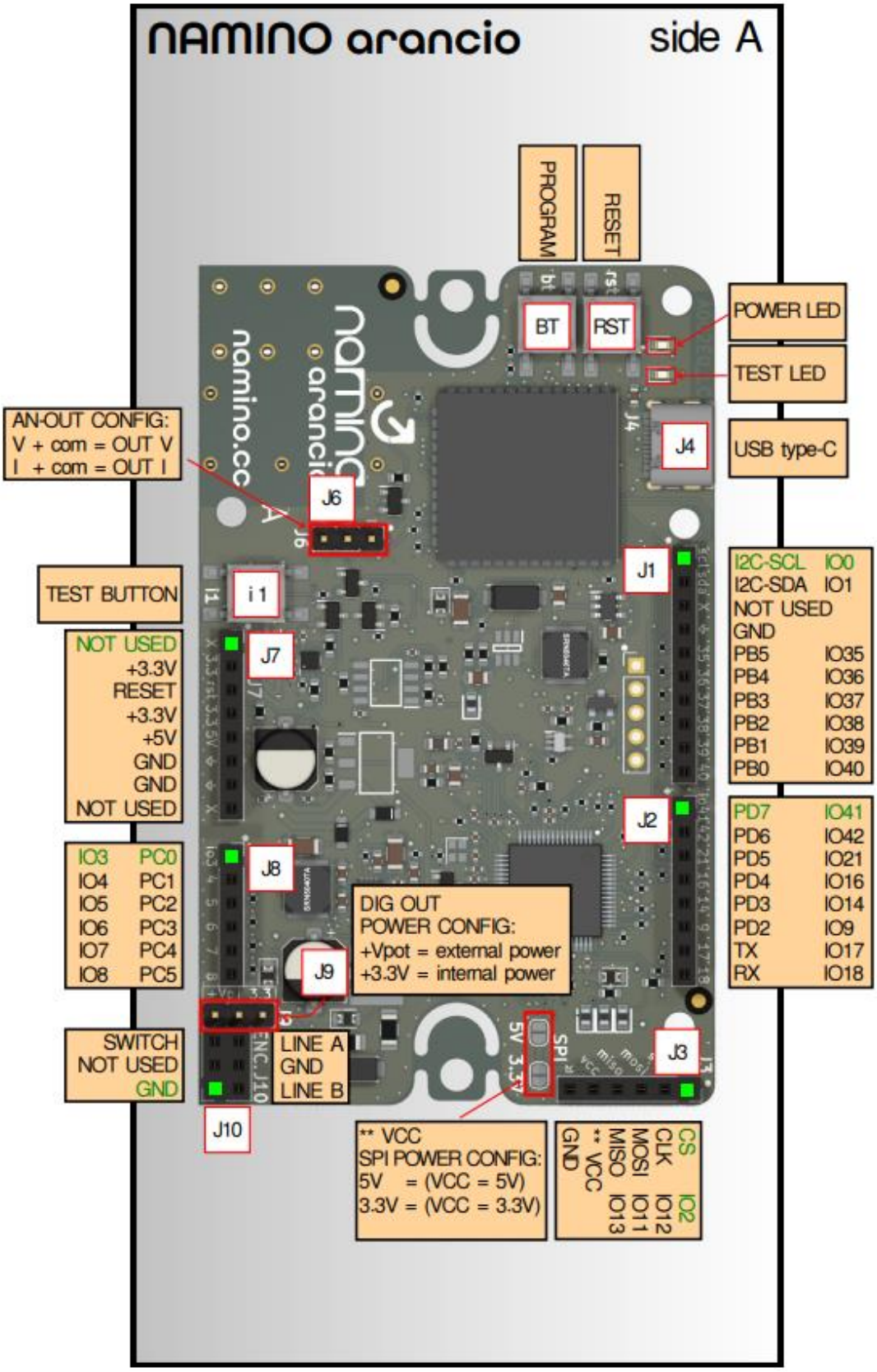
Namino ARANCIO contains the electronics for signal processing and it differs from Namino ROSSO only for the analog input section.

There are two kind of connections on the board:

- Side A: Connections to compatible ARDUINO UNO shields
- Side B: Connections with field inputs / outputs or connection with Namino NERO VDC or VAC

4.3.1. A-side connections

On side A there is an interface compatible with the ARDUINO UNO shields. The available pins are software configurable. There are also additional interfaces: knob encoder with switch, SPI, UART, I²C and USB C type.



- **J1, J2, J7, J8:** Interface compatible with ARDUINO UNO shields.
  The PBx, PCx and PDx registers/signals inherent to the Arduino Uno interface must be programmed according to the relative **IOx name which corresponds to the ESP32 interface**.
  J1: I²C (I2C-SCL/ I2C-SDA – IO0/IO1). On the I²C bus there is an RTC at the address 0x68.
  J2: UART id 1, serial hardware (TX/RX – IO17/IO18).

- **J3:** SPI interface (requires a soldered jumper to select the power supply). For example, it is possible to connect an external card such as SD-CARD (available for purchase in the accessories section of shop.namino). The IOx signals corresponds to the ESP32 interface.

- **J4**: USB C interface is of type Device and looks like VCOM (Virtual serial COMmunication port). From the USB port it is possible to power up the entire board.

- **J5:** Internal use.

- **J6:** Configuration required via jumper to configure the analog output in voltage [V] or Current [mA]. IMPORTANT: it needs a software configuration too.

- **J9:** Jumper configuration to choose the power voltage of the digital outputs: internal (3.3VDC) or equal to the +Vpot power supply (max 24VDC).

- **J10:** Knob encoder interface with switch or encoder NPN.

- **i1 BUTTON:** Connected in parallel to DIG IN1 (digital input 1) - Side B – J11 (for TEST).

- **RST BUTTON:** ESP32 RESET command (as if the device were turned on again).

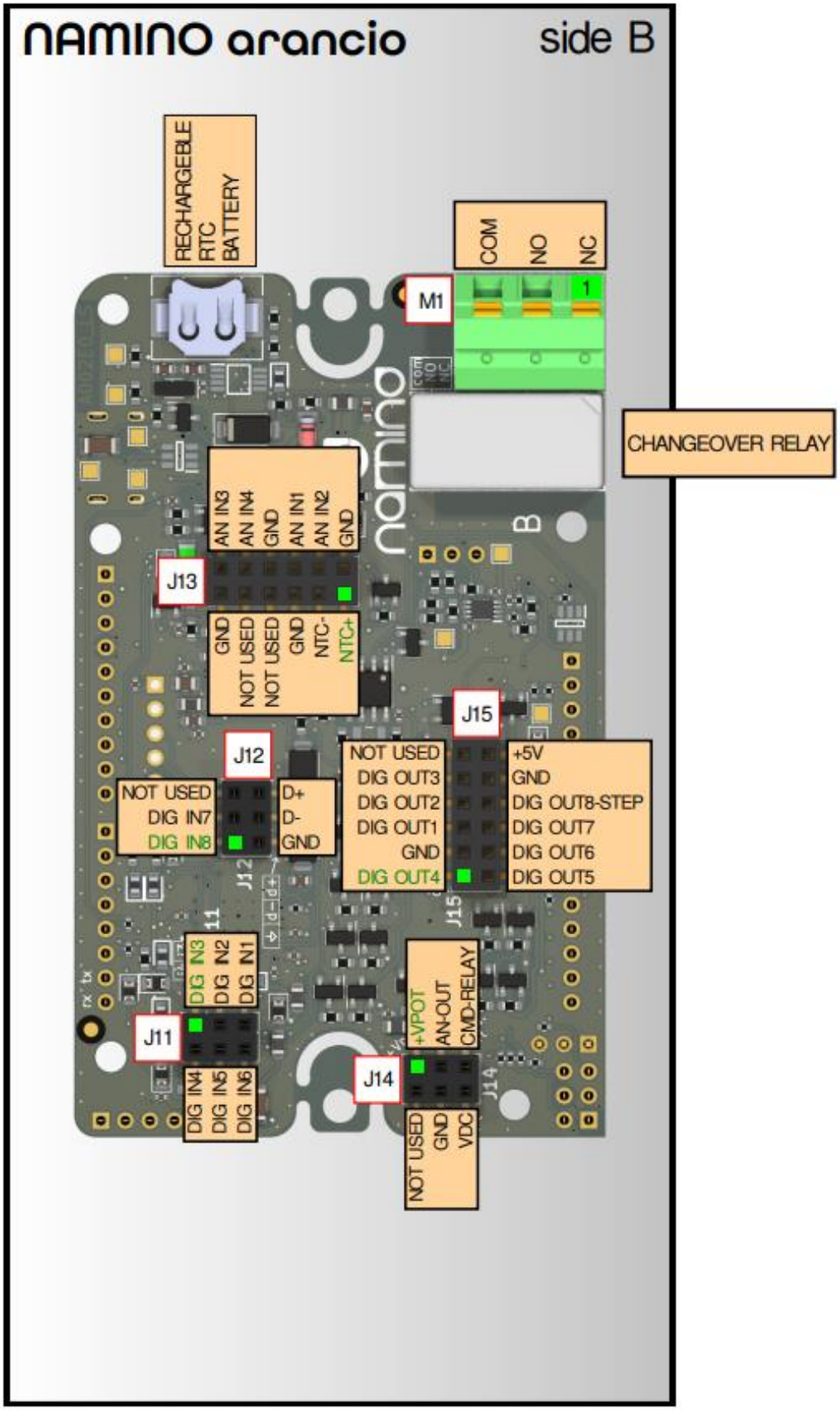
- **BT BUTTON:** Set up the programming mode: press this button to enter in programming mode, then press the **RST** button to exit programming mode.

- **POWER LED:** Indicates that the device is powered.

- **TEST LED:** Connected in parallel to DIG OUT1 (digital output 1) - Side B – J15 (for TEST).

4.3.2. B-side connections

Side B provides the signals of digital and analog inputs / outputs for the field. Optionally, these strip pins goes to the Namino NERO board for greater wiring convenience.



- **J11:** Digital inputs. For more details, see the Input / output signals chapter.
- **J12:** Digital inputs and RS485 signals. For more details, see the Input / output signals chapter.
- **J13:** Analog inputs. For more details, see the Input / output signals chapter.
- **J14:** Board power supply (7-30VDC), digital output power supply (+Vpot max 24VDC), command for an external 5V relay (in parallel with 5A SPDT relay - M1) and the analog output. For more details, see the Input / output signals chapter.
- **J15:** Digital outputs. For more details, see the Input / output signals chapter.
- **M1:** 5A SPDT relay interface. **COM** = COMMON, **NO** = NORMALLY OPEN, **NC** = NORMALLY CLOSED.
- **RTC:** (real time clock / calendar) equipped with a rechargeable buffer battery.

4.3.2.1. I/O features

I/O features					
Analog inputs	N° 4 (AN IN1 – AN IN4)	**Input type**	**Resolution**	**Bit**	**Note**
		0÷20 mA	0.001mA	16	Input impedance 13Ω
		0÷10V	0.001V	16	Input impedance >1MΩ
Analog input	N° 1 (AN IN5)	**Input type**	**Resolution**	**Bit**	**Note**
		NTC -40°C ÷ 130°C	0.5°C	16	Beta Value 25/50 K 3950 10kΩ
Analog output	N° 1 (AN-OUT)	**Output type**	**Resolution**	**Bit**	**Note**
		0÷20 mA	0.02mA	10	Max impedance : 400Ω
		0÷10V	0.01V	10	Min impedance : 1KΩ
Digital inputs	N°8 (DIG IN1 – DIG IN 8)	**Input type**	**Resolution**		**Note**
		PNP	cycle time		
Digital outputs	N°8 (DIG OUT1 - DIG OUT8)	**Output type**	**Resolution**		**Note**
		PNP	cycle time		200mA total

5. Programming

5.1. Introduction

NAMINO can be programmed in C/C++ using::

- Arduino IDE
- Platformio.org

Note: press the **BT button** to enter in programming mode, see chapter "A-side connections" (ROSSO) or "A-side connections" (ARANCIO).

Boards can be programmed only using a USB type A -> USB type C cable.

5.2. Functions for C++ programming interface

The industrial/analog functions are exposed as a series of registers read/written from the SPI bus. For convenience, the functions for handling the industrial side of NAMINO are made available to the programmer as methods of a class.

ROSSO

- namino_rosso.cpp
- namino_rosso.h
- pins_arduino.h

Downloadable from the examples.

ARANCIO

- namino_arancio.cpp
- namino_arancio.h
- pins_arduino.h

Downloadable from the examples.

The header of the file contains the version/date:

```
Copyright (c) 2023 Namino Team, version: x.y.z @ yyyy-mm-dd
```

5.2.1. File platformio.ini

Use the following code to initialize NAMINO profile with platformio, some code is useful in future for console/debug/etc

[env:adafruit_feather_esp32s3_reversetft]
platform = espressif32
board = adafruit_feather_esp32s3_reversetft
board_build.f_cpu = 240000000L
board_flags = -DARDUINO_USB_CDC_ON_BOOT=1
build_flags = -DCORE_DEBUG_LEVEL=0
framework = arduino
monitor_speed = 115200
monitor_echo = yes
monitor_eol = LF
monitor_filters =
colorize
time
send_on_enter
lib_deps =
stevemarple/RTCx@^1.1.10
emelianov/modbus-esp8266@^4.1.0

5.2.2. Initial setup

Use the following code to initialize the NAMINO program:

#include <Arduino.h>
#include <SPI.h>
#include <Wire.h>
#include <RTCx.h>
#include "WiFi.h"

#include "./namino_rosso.h"
…

```
void setup() {
  Serial.begin(115200);
  delay(500);

  // reset namino microcontroller
  nr.resetSignalMicroprocessor();

  nr.begin(800000U, MISO, MOSI, SCK, SS);

  // optional: only if using RTC
  Wire.begin(NAMINO_ROSSO_I2C_SDA, NAMINO_ROSSO_I2C_SCL);
...
```

5.2.3. Loop cycle

In the loop() routine must be insert the following functions, which allow the exchange of values with the industrial side of the board:

```
nr.readAllRegister();
… reading functions from the industrial side
… write functions to the industrial side
nr.writeAllRegister();
```

5.2.4. Base

Opening/closing of communication / initialization of the industrial side interface:

void begin(uint32_t clock=800000U);

void begin(uint32_t clock=800000U, uint8_t miso=MISO, uint8_t mosi=MOSI, uint8_t sck=SCK, uint8_t ss=SS);

void end();

5.2.5. Industrial side register map

typedef enum { ... } NAMINO_ROSSO_REGISTERS;

low-level interface, handled by the following functions.

5.2.5.1. Digital I/O functions

Reading of the single digital input:

Parameters:

channel from 1 to 8

```
bool readDigIn(int channel);
```

Reading of the single digital output. After a write to the digital output, the latched value can be read:

```
bool readDigOut(int channel);
```

Writing of the single bit of the digital outputs from 1 to 8:

Parameters:

channel: number of outputs to update

value: value to be assumed by the output

```
void writeDigOut(int channel, bool value);
```

Relay status setting:

```
void writeRele(bool value);
```

5.2.5.2. Step direction

Setting the parameters of the stepper PTO (Pulse Train Output), on digital output 8 (DIG OUT8):

Parameters:

frequency: pulse train frequency, in Hz from 1 to 65535, if zero, output 8 is configured as a standard digital output.

stepsNumber: number of steps that the PTO must carry out. If set to 0, the PTO does not finish counting until the stop command is received.

```
void setupStep(uint16_t frequency, uint32_t stepsNumber);
void writeStepNumber(uint32_t value);
void writeStepFreq(uint16_t value);
```

Departure of the PTO:

```
void startStepper();
```

PTO stop:

```
void stopStepper();
```

5.2.5.3. Encoder knob

Reading of the state of the switch associated with the encoder knob:

```
bool readEncoderSwitch();
```

Reading of the current value of the encoder:

```
int32_t readEncoder();
```

Reset of the encoder current value:

```
resetEncoderCounter();
```

Writing the current value of the encoder:

```
setEncoderCounter(int32_t value);
```

5.2.5.4. Analog output

Analog output programming:

Note: also set the jumper (side A - J6) in the V/mA position.

Parameters (**conf**):

```
  OUT_CH01_DISABLED    = 0,
  OUT_CH01_CURRENT     = 1,
  OUT_CH01_VOLTAGE     = 2,
writeAnalogOutConf(ANALOG_OUT_CH01_CONF_VALUES conf);
```

Analog output setting in units of measure: 0-10V / 0-20mA.

```
void writeAnalogOut(float_t value);
```

value = 1000 = 10V
value = 2000 = 20mA

5.2.5.5. Analog input configuration ROSSO

This is followed by the analog input configuration setting.

Note: each input has its specific modes in V/A/PTC/NTC/thermocouple/PT1000/load cell, and can engage 1/2/4 analog inputs in the terminal board, example, channel 5 has these functions:

```
typedef enum {
  CH05_DISABLED    = 0,
  CH05_CURRENT     = 1,
  CH05_VOLTAGE     = 2,
  CH05_TCJ         = 10,
  CH05_TCK         = 11,
  CH05_TCT         = 12,
  CH05_PT1000      = 20,
  CH05_NTC         = 21,
  CH05_PTC         = 22,
  CH05_LOAD_CELL   = 30,
} ANALOG_IN_CH05_CONF_VALUES;
```

To set the configuration, in the loop() inside a **conditional**, there must be all the initial analog configurations, after having waited for the industrial side to start,

```
if (configAN && nr.isReady()) {
    configAN = false;
... analog configuration …
      nr.writeRegister(WR_ANALOG_IN_CH01_CONF, ANALOG_IN_CH01_CONF_VALUES::CH01_VOLTAGE);
      nr.writeRegister(WR_ANALOG_IN_CH02_CONF, ANALOG_IN_CH02_CONF_VALUES::CH02_VOLTAGE);
...
}
```

where **nr** is a global class:

```
namino_rosso nr = namino_rosso();
```

and initially we have:

```
bool configAN = true;
```

5.2.5.6. Analog input configuration ARANCIO

This is followed by the analog input configuration setting

Note: each input has its own specific configuration modes in V/A and engages 1 terminal in the terminal block (excluding channel 5), for example, channel 2 has these functions:

```
typedef enum {
  CH02_DISABLED    = 0,
  CH02_CURRENT     = 1,
  CH02_VOLTAGE     = 2,
} ANALOG_IN_CH02_CONF_VALUES;
```

To set the configuration, in the loop() inside a **conditional**, there must be all the initial analog configurations, after having waited for the industrial side to start,

```
if (configAN && nr.isReady()) {
    configAN = false;
... analog configuration …
      nr.writeRegister(WR_ANALOG_IN_CH01_CONF, ANALOG_IN_CH01_CONF_VALUES::CH01_VOLTAGE);
      nr.writeRegister(WR_ANALOG_IN_CH02_CONF, ANALOG_IN_CH02_CONF_VALUES::CH02_VOLTAGE);
...
}
```

where **nr** is a global class:

```
namino_arancio nr = namino_arancio();
```

and initially we have:

```
bool configAN = true;
```

5.2.5.7. Reading analog inputs ROSSO

Allows reading converted into reference unit from analog channel 1..8:

```
float_t readAnalogIn(int channel);
```

Reading converted to reference units by the channel with a connected thermocouple and/or thermometric probe:

```
float_t readPt1000(int channel);
float_t readNTC(int channel);
float_t readTCJ(int channel);
float_t readTCK(int channel);
float_t readTCT(int channel);
float_t readPTC(int channel);
```

Note: the values of the PT1000/PTC/NTC/TC J, TC K, TC T probes can be read in logical channels from 1 ÷ 4.

Reading of the load cell value referred to the analog channel:

```
float_t readLoadCell(int channel);
```

Note: Load cell values can only be read in logical channels 1 and 2.

5.2.5.8. Reading analog inputs ARANCIO

Allows reading converted into reference unit from analog channel 1..5:

```
float_t readAnalogIn(int channel);
```

Reading converted into reference units by the channel with an NTC probe connected:

```
float_t readNTC(int channel);
```

Note: the NTC probe value can only be read in channel 5.

5.2.5.9. System functions

Allows reading of the life time counter, i.e. a counter that increases 1 time per second from power on:

```
uint32_t readLifeTime();
```

Return signal board firmware version:

```
uint16_t fwVersion();
```

Return board code identifier values 0x1338=rosso, 0x2904=arancio, ...:

```
uint16_t boardType();
```

Internal temperature of the board (only for Namino ROSSO):

```
float_t readTBoard();
```

Writing in a register of the type NAMINO_ROSSO_REGISTERS:

```
void writeRegister(uint8_t address, uint16_t value);
```

Industrial side board reset:

```
resetSignalMicroprocessor();
```

The industrial side NAMINO board performs some initializations, if the board has finished the initialization and is ready this function returns true:

```
bool isReady();
```

5.3. Firmware updates

ROSSO / ARANCIO namino board firmware update:

Follow the video tutorial to update the firmware version of the namino board. Download the update files.

There are 2 folders:

* pyblhost

* pyblhost_bridge_esp32

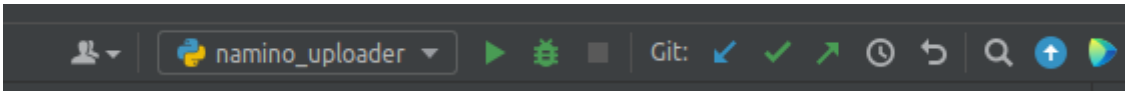
Note: pyblhost is not a fancy name, blhost is a NXP sw and pyblhost is the open source python 3 client version of the bootloader.

5.3.1. pyblhost

Prerequisites:

A python environment, eg: pycharm community, installed with this tool:
https://www.jetbrains.com/toolbox-app/

pyblhost contains a pycharm project already configured with the firmware .bin and launch arrow key.




```
[INFO]
-----------------------------------
[INFO] Namino Rosso NM firmware uploader
[INFO] -----------------------------------
[INFO] Namino Rosso firmware: Namino_ROSSO_Project.bin start addr: 0 zero bytes count: 63000
```

Note: the software has been prepared for the serial ports of a linux / ubuntu environment therefore the ports are called:

- /dev/ttyACM0
- /dev/ttyUSB0

For windows permanently change the vs. port assigned to file line *pyblhost_uploader.py*

```
def main():
    # BlhostSerial specific arguments
    port, baudrate, title = '/dev/ttyACM0', 115200, 'Namino Rosso'
```

5.3.2. pyblhost_bridge_esp32

The “pyblhost_bridge_esp32” directory contains a program to load into the namino board's ESP32, and after starting it, wait a few seconds for it to start, and run pyblhost to load the firmware.

Note: It's a vscode / codium project.

5.4. Examples

Examples can be downloaded from the Github section.