PiGear Nano:

Nano-ITX Raspberry Pi Compute Module 4 Carrier Board

User Manual (revision 1.04)

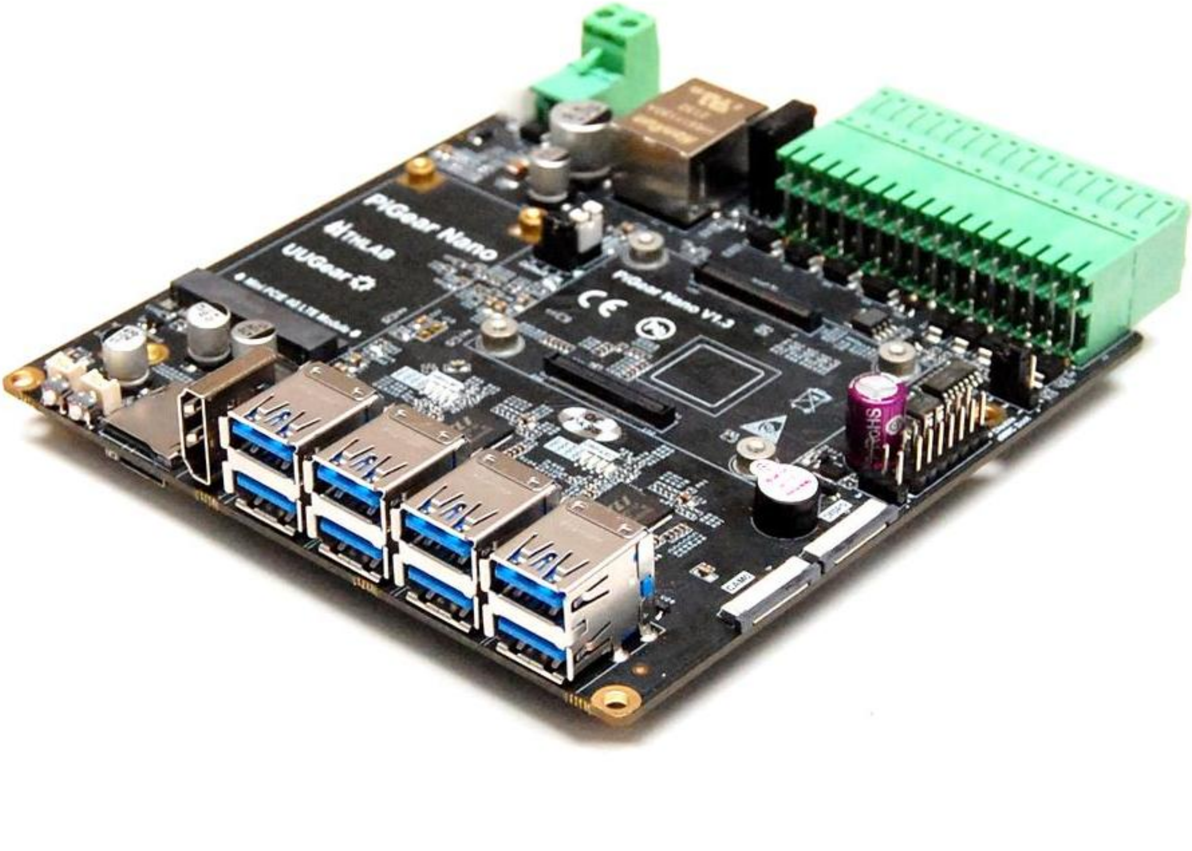
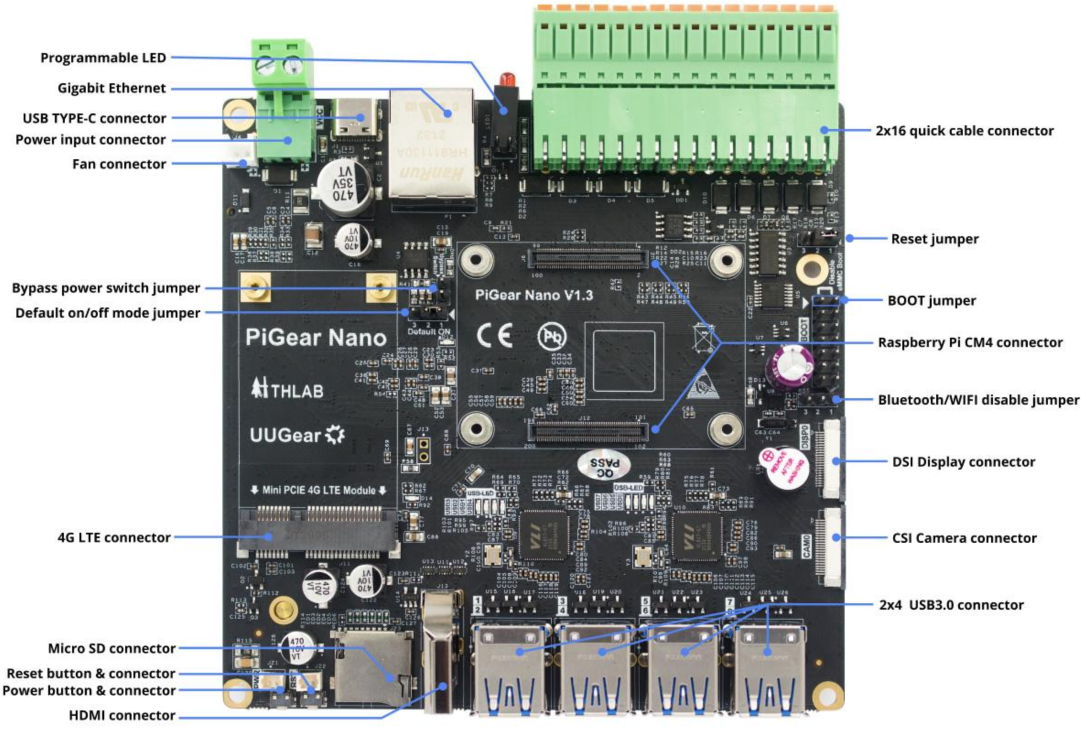


Table of Contents

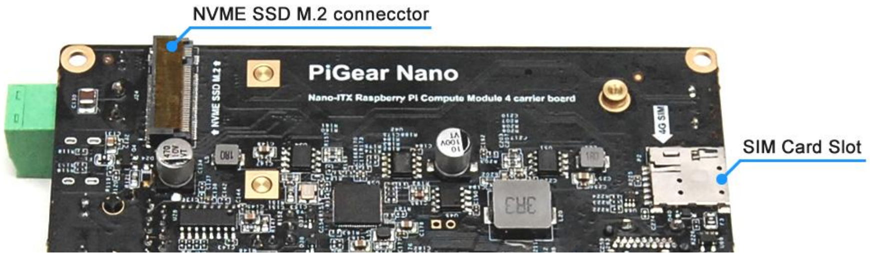
1. Product Overview

PiGear Nano is a high performance Nano-ITX (12x12cm) carrier board for Raspberry Pi Compute Module 4. PiGear Nano is mainly designed for industrial applications, but you can also use it as office computer, family media center, gaming console etc.

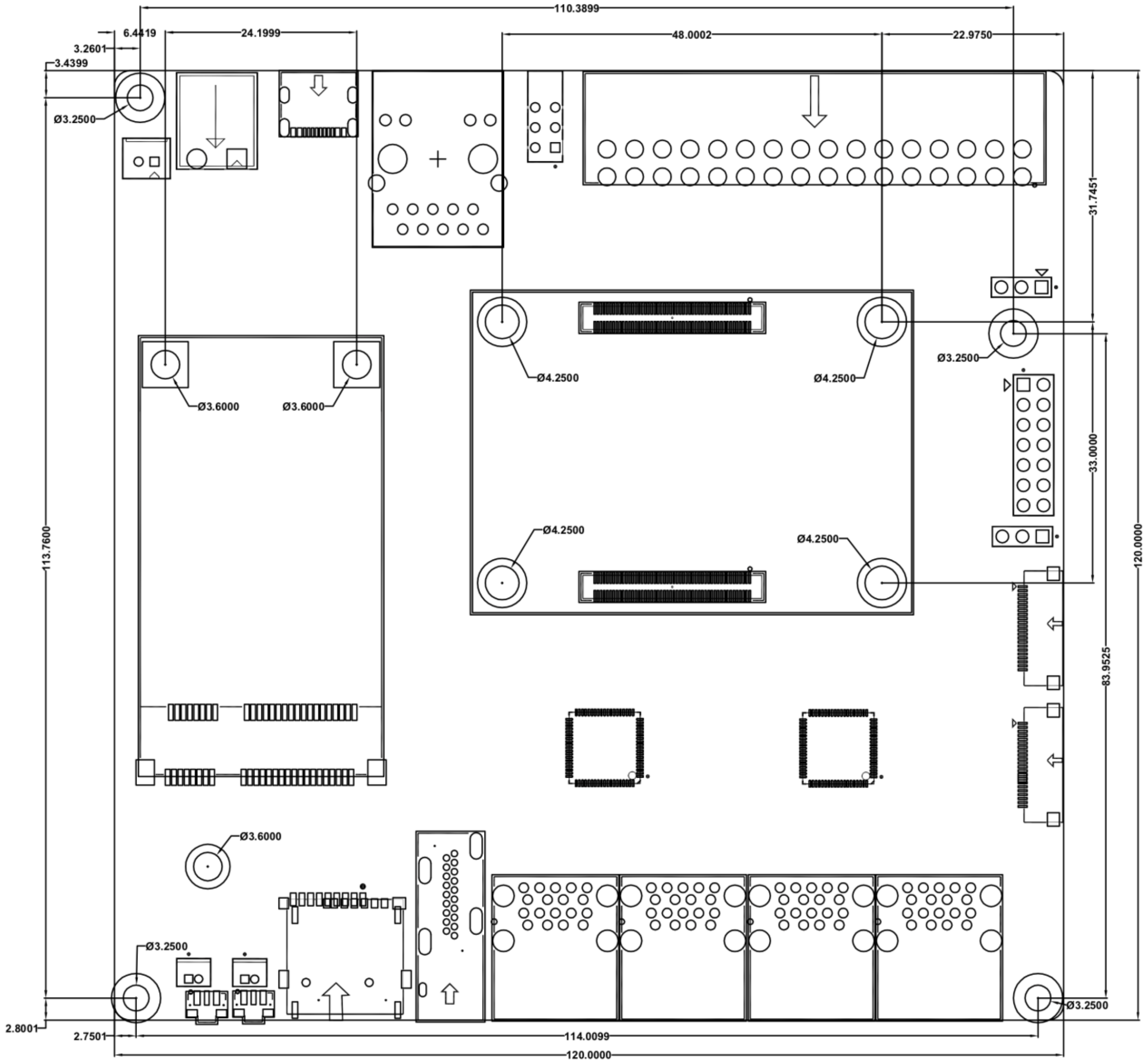
PiGear Nano offers a wide range of resources and interfaces. Please see below the diagram for details:



There are also an NVME SSD M.2 connector and a SIM card slot underneath the board.



You can place PiGear Nano into a Nano-ITX computer case. Below is a mechanical drawing of the board.


110.3899
6.4419
24.1999
48.0002
22.9750
3.2601
3.4399
Ø3.2500
Ø3.6000
Ø3.6000
Ø4.2500
Ø4.2500
Ø3.2500
31.7451
Ø4.2500
Ø4.2500
33.0000
113.7600
120.0000
83.9525
Ø3.6000
Ø3.2500
Ø3.2500
2.8001
2.7501
114.0099
120.0000

2. Specification / Technical Details

Network	Gigabit-Ethernet RJ45 socket x 1 MINI PCIE LTE/4G/GPRS interface x 1 On-board SIM card slot x 1
USB	USB 3.0 Type-A connector x 8 USB 2.0 Type-C connector x 1 (for flashing CM4, or powering the board with light load)
Analog & Digital I/O	18-bit ADC input x 4 Digital input x 4 Digital output x 4 Configurable digital I/O x 4
Signal Interface	RS232 x 2 RS485 x 2 CAN x 1 Wire interface x 1
Storage	NVME SSD M.2 interface x 1 Micro SD card slot x 1 (for Compute Module 4 Lite only)
Display	HDMI Type-A connector x 1 MIPI DSI interface x 1
Camera	MIPI CSI interface x 1
Other	Real-time clock x 1 (with supper capacitor for off-power time keeping) Buzzer x 1 Power Indicator x 1 Programmable LED indicator x 1 5V fan interface (PH2.0 connector) Power button x 1 (with extension connector) Reset button x 1 (with extension connector)
Power Supply	DC 7V~30V (with reverse polarity protection) or DC 5V (via USB Type-C connector)
Quiescent Current	~1mA

Board Size and Weight	NANO-ITX 12 x 12 cm, 150g
Operating Environment	Temperature -30°C~80°C (-22°F~176°F) Humidity 0~80%RH, no condensing No corrosive gas

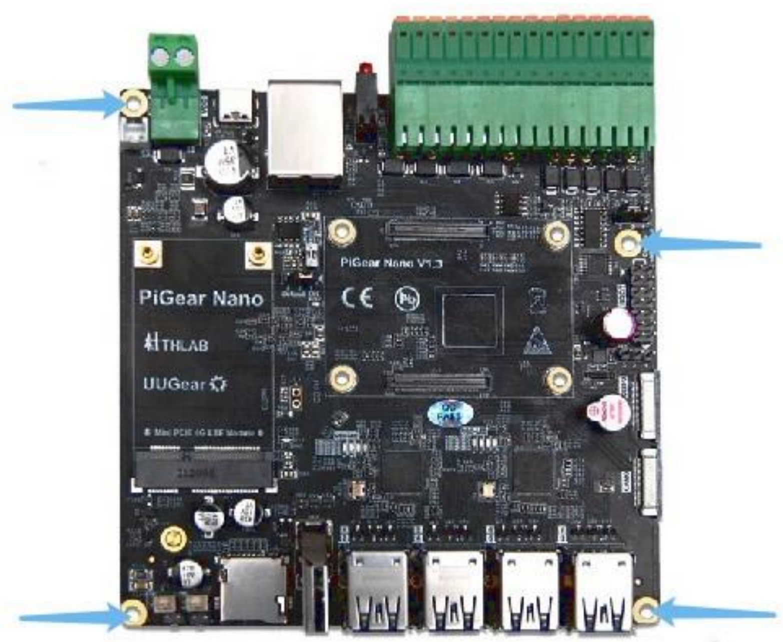
3. Package Content

Each package of PiGear Nano contains:

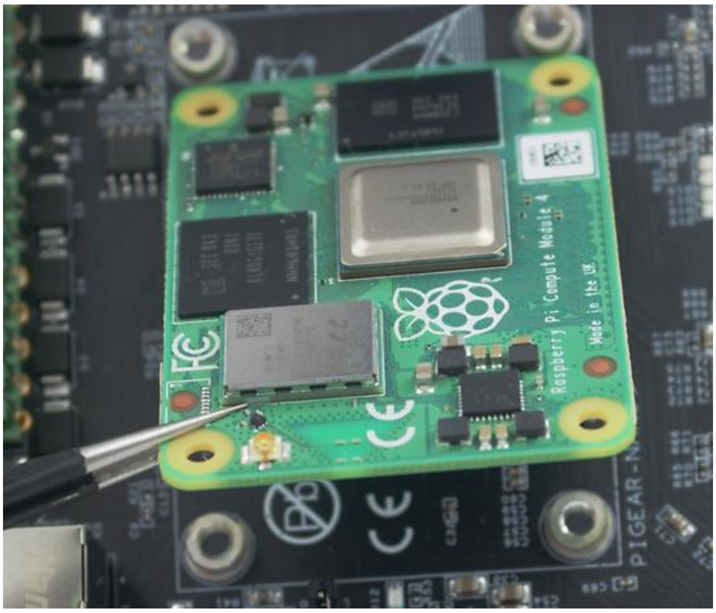
- PiGear Nano board x 1
- 6x6x5 heat sink x 1 (for VL805 chip on the back of the board)
- M3x8 metal standoff x 4
- M3 screws x 4
- M2 screws x 3
- 2.54-2P jumper cap x 5

4. Connect Raspberry Pi Compute Module 4 to the Board

Before starting, it is recommended to install four metal standoffs underneath so the board can lay flat on the desk. You can find four metal standoffs and four screws in the package. Just place the standoff under the mounting hole and use a screw to secure it.



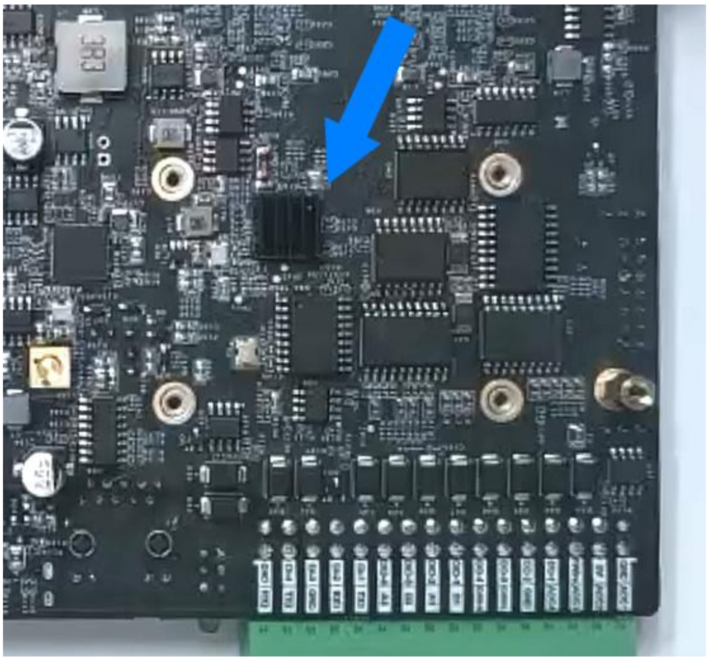
Connect Raspberry Pi Compute Module 4 to the board by simply aligning it with the two connectors on board (or alternatively you can align the CE marks on CM4 and the board), and then give it a good squeeze. It will crunch into place (and you can hear it).



If you will use the device in a vibrating environment, you may want to secure your Raspberry Pi Compute Module 4 on the board with four M2.5 screws (not included in the package).

It is also recommended to put the small heatsink (included in the package) on the VL805 chip on the back of the board. Because this chip could get rather hot if the board is processing many USB data translation.

You can peel off the paper on the heatsink and directly stick it on the chip VL805, as shown below:

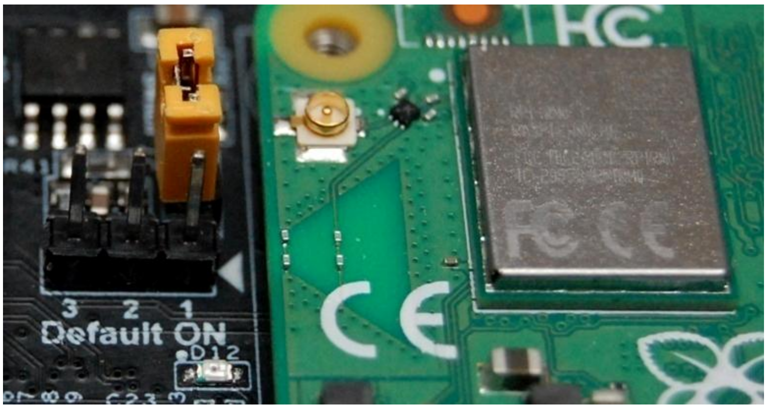


5. Installing Operating System

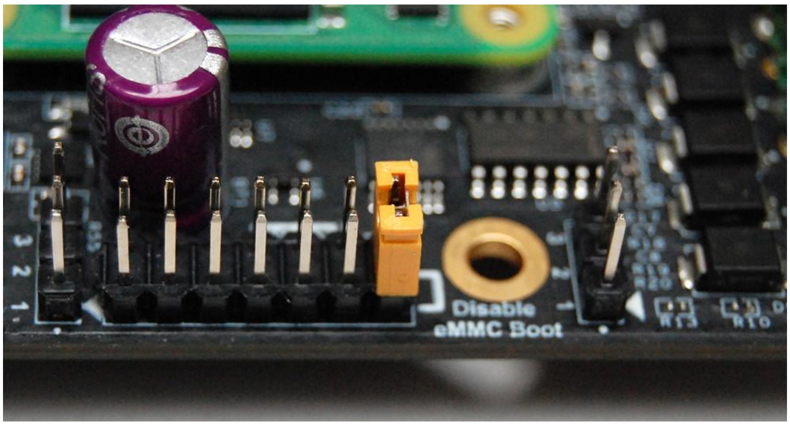
5.1 For Raspberry Pi CM4 with eMMC

If your Raspberry Pi Compute Module has eMMC, you can flash the OS into its eMMC. If you want, you can have this done with the official I/O board. However using the official I/O board is not necessary because you can get this done on PiGear Nano board directly.

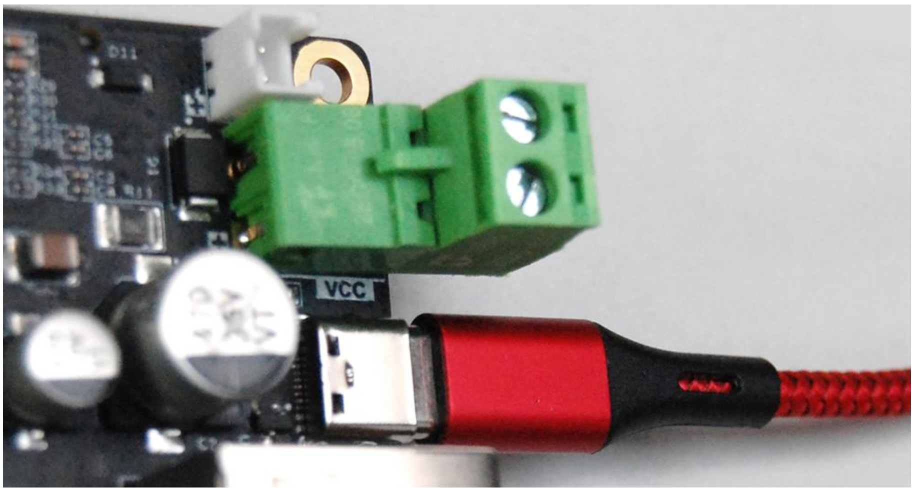
Before proceeding, please make sure to short the "bypass power switch" jumper. This is to bypass the e-latch power switch, so your PiGear Nano board will be powered even without the operating system. The "bypass switch" jumper is on the left of Raspberry Pi CM4 (above the "Default On" jumper), as shown in picture below:



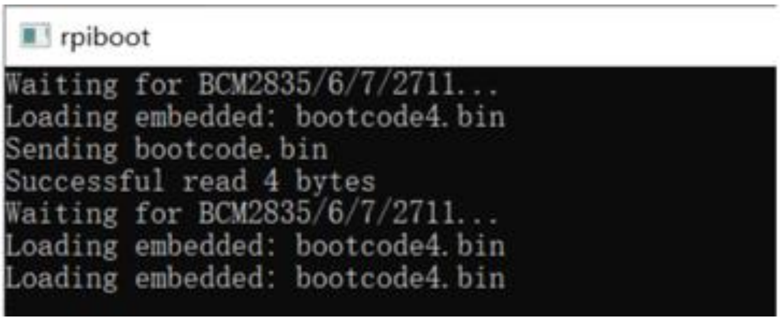
You will also need to short the "Disable eMMC Boot" jumper, so the eMMC on your Raspberry Pi CM4 will boot in USB mass storage mode, and you can flash the disk image into it.



Now connect a USB Type-C cable to PiGear Nano, and connect the other end of the cable to your computer. This USB cable will power your PiGear Nano during the installation process, and there is no need to connect any other power supply.



You will need the "usbboot" tool to turn the Raspberry Pi device into a USB Mass Storage Device. If your computer is running Windows, you can download and run the "rpiboot_setup.exe" in the "win32" directory to install the usbboot tool. If you are using Mac OS X or Linux, please follow the instructions in README.md file to build the tool and then run it.



```
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Sending bootcode.bin
Successful read 4 bytes
Waiting for BCM2835/6/7/2711...
Loading embedded: bootcode4.bin
Loading embedded: bootcode4.bin
```

You will then see a drive with label "boot" appear in your computer. The next step is to flash the OS disk image into this drive. You can achieve this with different kinds of tools. For example, you can use Win32DiskImager under Windows, or use "dd" command under Mac OS X or Linux. You can also use Raspberry Pi Imager for any operating system.

After flashing the OS disk image into the "boot" drive, please don't forget to remove the jumper cap from the "Disable eMMC Boot" jumper, so that your PiGear Nano now works as a standalone computer system.

5.2 For Raspberry Pi CM4 Lite (without eMMC)

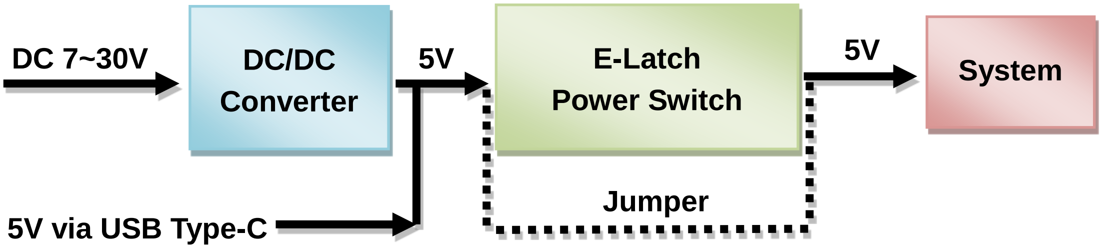
If you are using Raspberry Pi Compute Module 4 Lite, which does not come with eMMC, you can flash the OS image into an SD card. This procedure has no difference than installing the OS into SD card for other models of Raspberry Pi. If it is something new to you, you can read this detailed tutorial from the Raspberry Pi documentation.

6. Powering PiGear Nano

If you have installed the OS into your CM4's eMMC, or you have inserted an SD card with OS installed, you can power the board and boot the system.

You can power PiGear Nano with DC 7~30V via the universal 2-pin power connector, or you can power it with DC 5V via the USB Type-C connector.

When powering via the universal 2-pin power connector, the input voltage will go through a DC/DC convertor (SY8204) and output 5V for powering the system. Please make sure the input voltage is no higher than 30V. You can also bypath the DC/DC convertor and power the board directly with DC 5V via the USB Type-C connector. Either way you can optionally enable the e-latch power switch, which allows you to turn on/off the system with a single tape on the button.



Our software installation script will enable the e-latch power switch support for you, however you will still need to set the jumper correctly to make it work.

Note: when e-latch power switch is in used, after the power is fully cut, please wait a few seconds before trying to boot your PiGear Nano again, because the internal RC circuit needs some time to discharge the capacitor and restore to the standby status.

Note: if you are going to connect some high-current devices to PiGear Nano, it is recommended to power the board with higher voltage via the universal 2-pin power connector. In such case the internal DC/DC converter will output 5V up to 4A.

7. Software Installation / Update

After you boot the system up for the first time, please run our software installation script. The script will do these tasks for you:

- Create a backup of the current config.txt and cmdline.txt files.
- Modify your config.txt and cmdline.txt for correctly supporting the hardware.
- Install the daemon script to initialize the hardware and support e-latch power switch.
- Install the PiGearNano.sh application.

- Install UUGear Web Interface (UWI).

It is recommended to download and run the installation script in home directory. You can run this command to download the installation script:

```
pi@raspberrypi:~ $ wget https://www.uugear.com/repo/PiGearNano/installPiGearNano.sh
```

Then you can run the script with sudo:

```
pi@raspberrypi:~ $ sudo sh ./installPiGearNano.sh
```

After installation, you can reboot the system.

By default, the UWI server binds to “raspberrypi” host name, and you can access UWI via http://raspberrypi:8000/. However sometimes this host name cannot be resolved in your network environment, and in such case you will need to modify the UWI configuration file “uwi.conf” in “uwi” directory:

```
pi@raspberrypi:~ $ nano uwi/uwi.conf
```

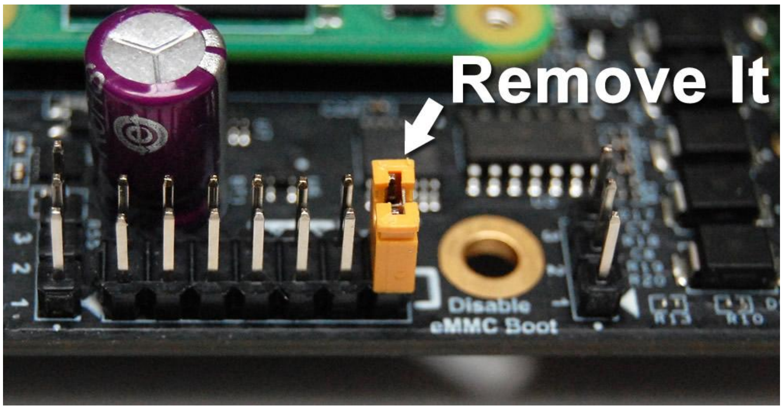
You will see **host='raspberrypi';** and you need to replace the ‘raspberrypi’ with the actual IP address or accessible host name. After rebooting the system, you can access UWI via http://[Your IP address or host name]:8000/.

In the future if there is a newer version of software available, you can rename or remove your “pgnano” directory and install the software again. UWI will be upgraded automatically and the old “uwi” directory will be renamed automatically.

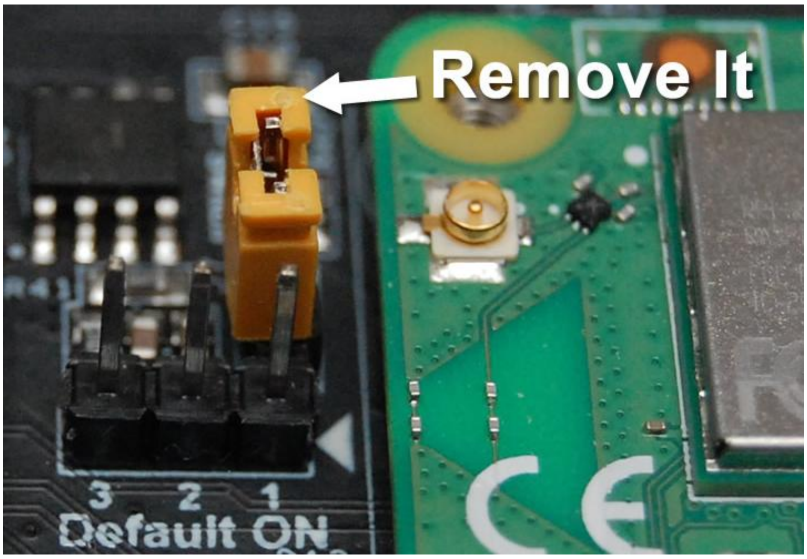
8. Jumper Settings

After installing the software, you may need to do some jumper settings to make the board fully functional.

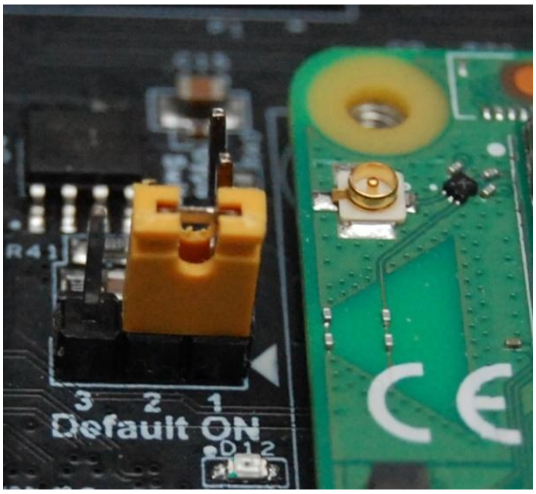
If you are using Raspberry Pi Compute Module 4 with eMMC, you need to remove the jumper cap from the "Disable eMMC Boot" jumper after installing the operating system into eMMC. This jumper is part of the multifunctional 2x7 pin jumper, which will be introduced later.



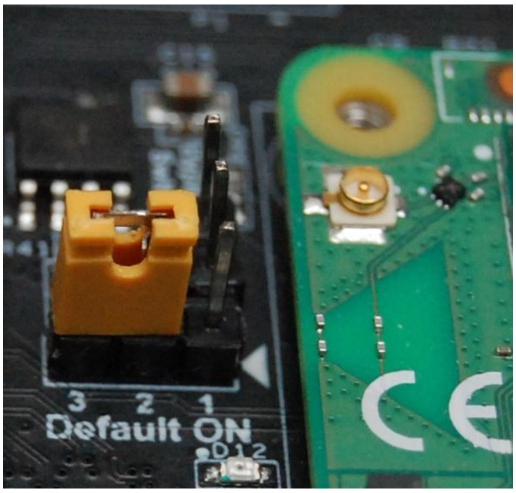
With the software installed, the e-latching power switch is supported on software level. However, you still need to remove the "bypass power switch" jumper cap, otherwise PiGear Nano cannot fully cut the power after system shutdown. If you do not want to use the e-latch power switch, you can keep this jumper cap in place.



In addition, you will need to put a jumper cap on the "Default ON" jumper. This is to let you choose whether your PiGear Nano will automatically boot up when the power supply is connected. If you put the jumper cap on pin 3-2, it will not boot up until you tap the power button.

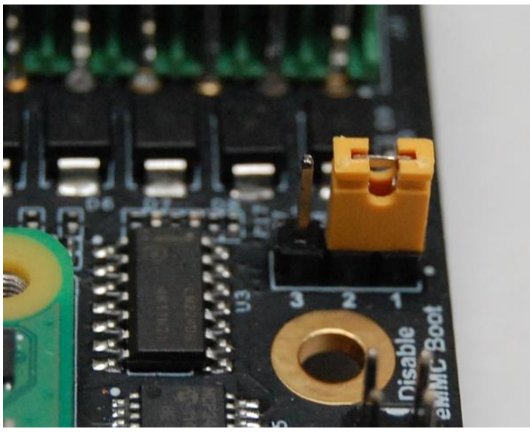


Default ON: Boot up when powered

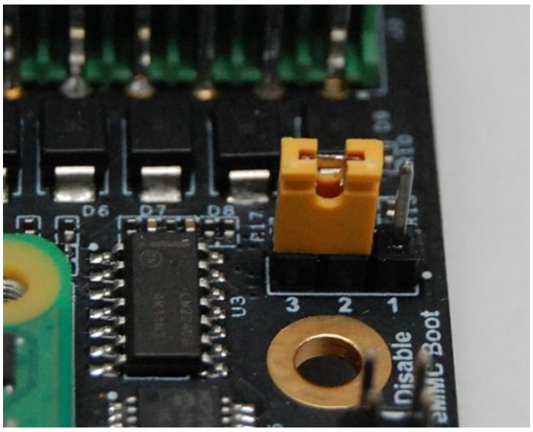


Default OFF: Tap the button to boot up

There is a "reset" jumper near the green 2x16 quick cable connector, and it is to define the behavior of the reset button. If you short pin 1-2, the reset button will pull down the GLOBAL_EN when you tap it. If you short pin 2-3, the reset button will pull down the RUN_PG when you tap it. Either way you can tap the reset button to restart your Raspberry Pi.

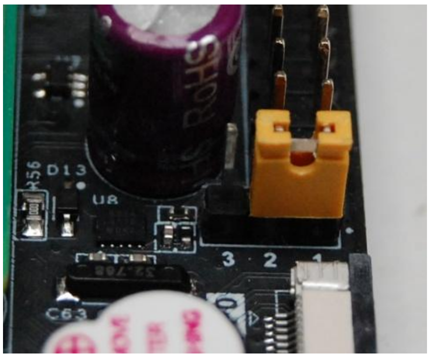


Reset button will pull down GLOBAL_EN

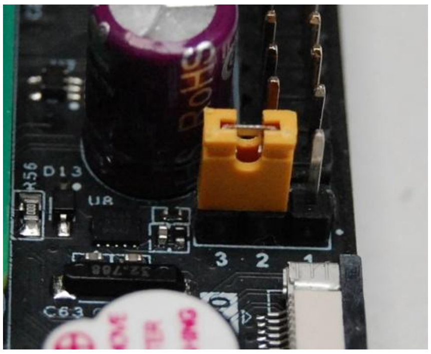


Reset button will pull down RUN_PG

There is also a 3-pin "wireless" jumper above the DISP0 connector, which is for disabling WiFi or Bluetooth.

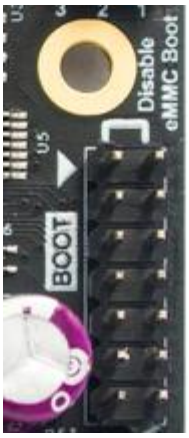


Shorting pin 1-2 to disable WiFi



Shorting pin 2-3 to disable Bluetooth

There is also a multifunctional 2x7 pin jumper between the "reset" jumper and the "wireless" jumper. It is actually the same as the J2 header on Raspberry Pi Compute Module 4 official I/O board. Please see below the descriptions.



1	2
3	4
5	6
7	8
9	10
11	12
13	14

1 and 2: force USB booting if shorted

3 and 4: write protect the EEPROM if shorted

5: MXL7704 analog input AIN0

6: MXL7704 analog input AIN1

7: GND for AIN0 and AIN1

8 and 9: SYNC_IN and SYNC_OUT for time synchronization with Precision Time Protocol

10: GND

11: TV output

12: GND

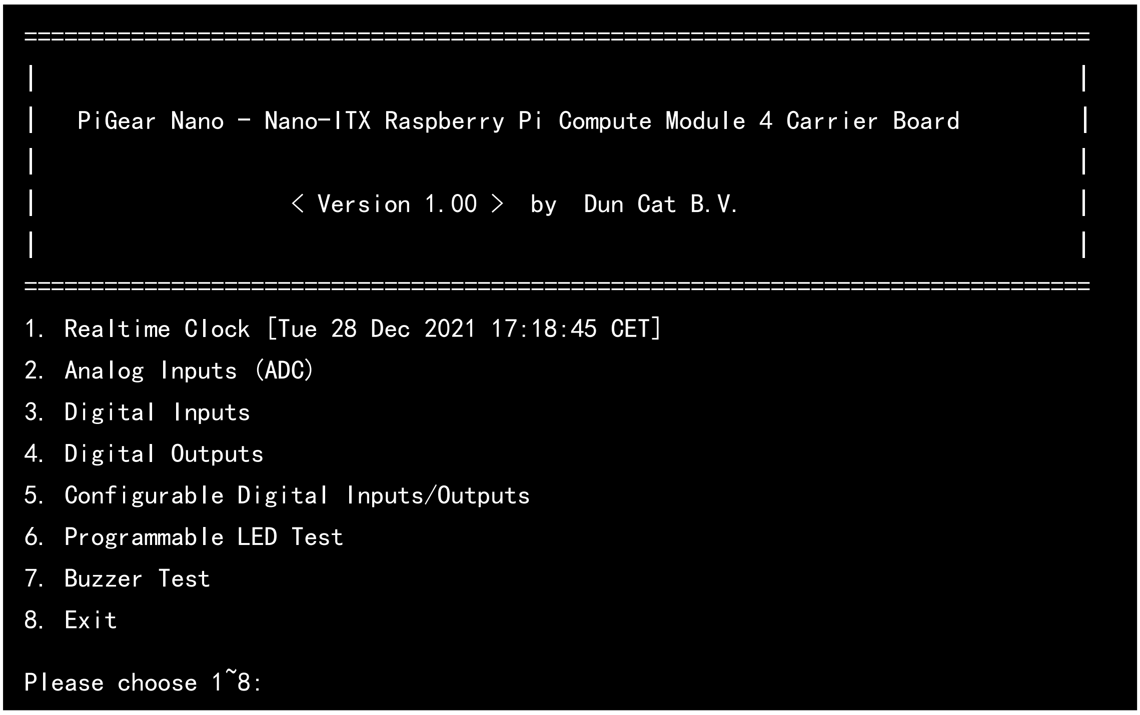
13 and 14: Connect a button between them, can be used to wake up compute module from power down. Has no effect when Raspberry Pi CM4 is on.

9. Software Usage

The software will be installed in a directory named "pgnano", while the UUGear Web Interface will be installed in a directory named “uwi”. You can run the software by running the PiGearNano.sh script:

```
pi@raspberrypi:~ $ cd pgnano

pi@raspberrypi:~/pgnano $ ./PiGearNano.sh
```

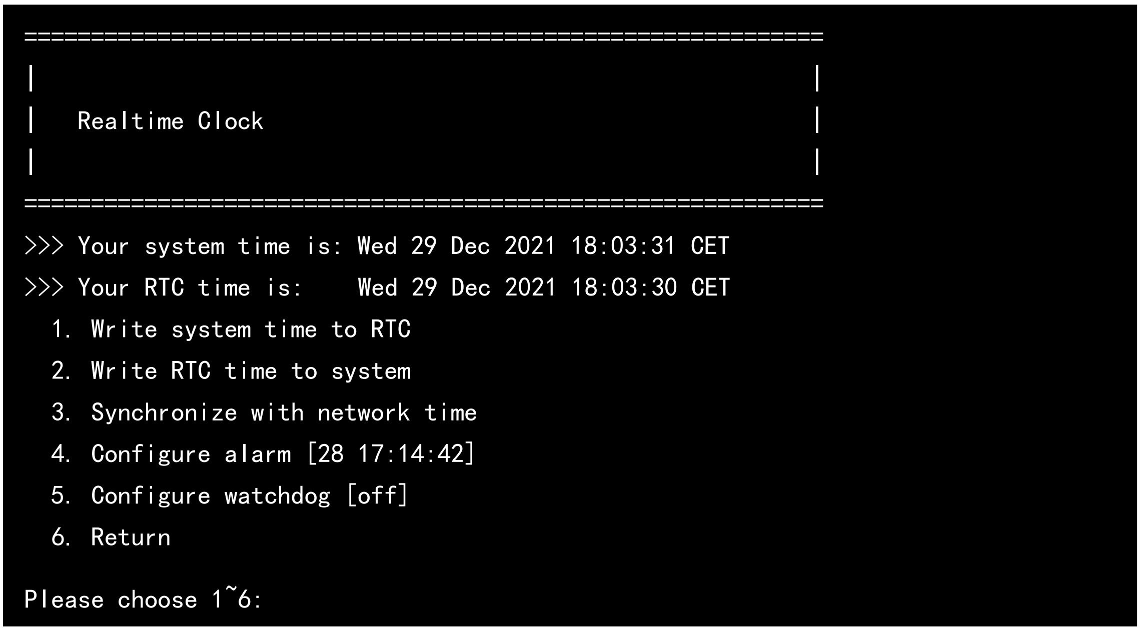
You will see the software interface like this:



```
================================================================================
|                                                                              |
|   PiGear Nano - Nano-ITX Raspberry Pi Compute Module 4 Carrier Board         |
|                                                                              |
|                   < Version 1.00 >  by  Dun Cat B.V.                         |
|                                                                              |
================================================================================
1. Realtime Clock [Tue 28 Dec 2021 17:18:45 CET]
2. Analog Inputs (ADC)
3. Digital Inputs
4. Digital Outputs
5. Configurable Digital Inputs/Outputs
6. Programmable LED Test
7. Buzzer Test
8. Exit

Please choose 1~8:
```

9.1 Realtime Clock

Here you can see the system time and the time in the real-time clock (RTC). You can write the system time into real-time clock or the other way around. You can also use Internet time to synchronize the system time and RTC time. You can press ENTER to refresh the time display.



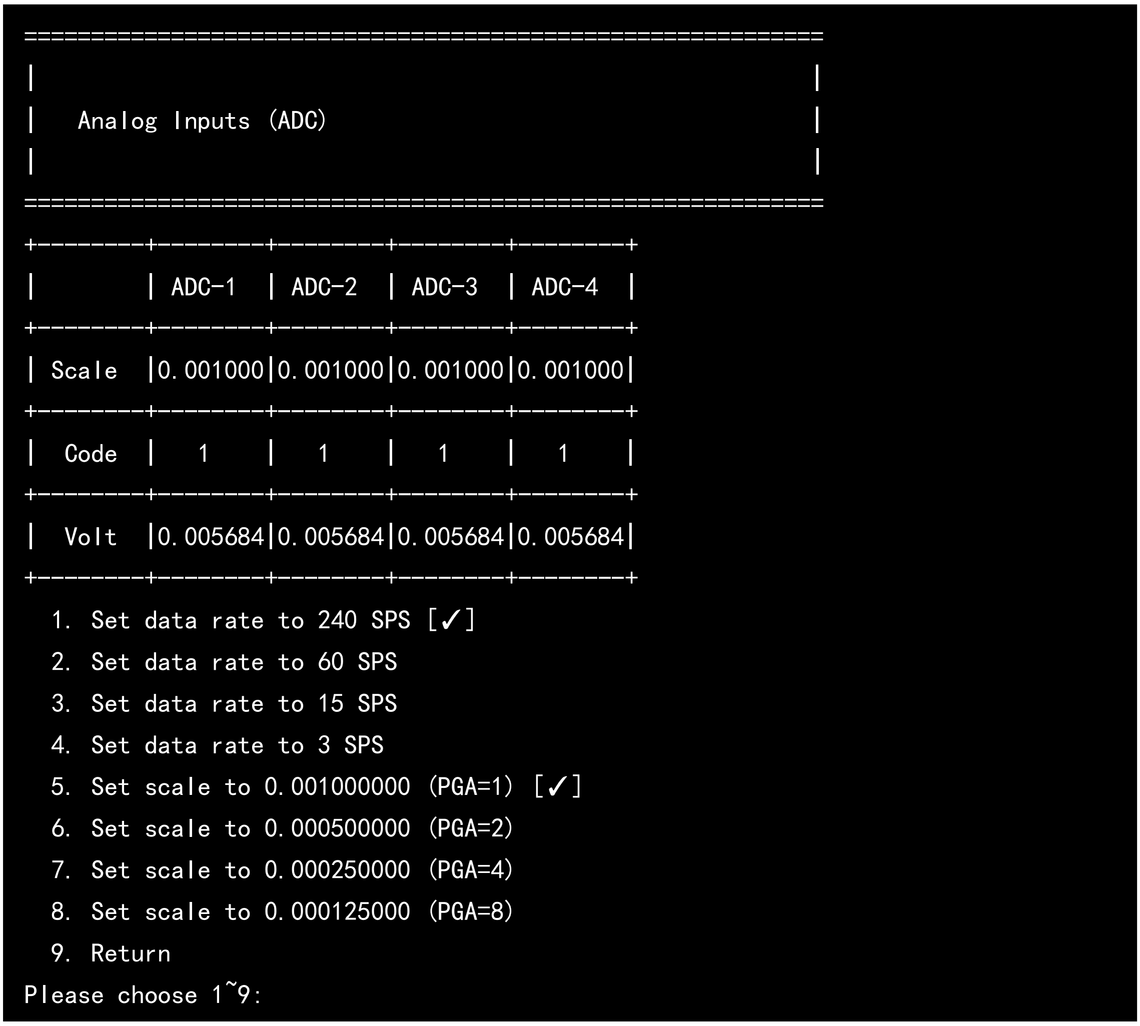
```
============================================================
|                                                          |
|   Realtime Clock                                         |
|                                                          |
============================================================
>>> Your system time is: Wed 29 Dec 2021 18:03:31 CET
>>> Your RTC time is:    Wed 29 Dec 2021 18:03:30 CET
  1. Write system time to RTC
  2. Write RTC time to system
  3. Synchronize with network time
  4. Configure alarm [28 17:14:42]
  5. Configure watchdog [off]
  6. Return

Please choose 1~6:
```

The real-time clock has one alarm and you can configure it to do different things. If you set the alarm time to a moment in the future (using option 4), the system will reboot when the alarm is triggered. If you do not use the e-latch power switch, which means your board will not be fully power off after shutdown, this alarm can also be used to wake up your Raspberry Pi.

Alternatively, you can also use this alarm as a watchdog (using option 5). When watchdog is turned on, the watchdog.sh process will run in the background and it keeps postponing the alarm. If the system is crashed and this process is halted, the alarm cannot be postponed and eventually gets triggered, which will force the system to reboot. This functionality has no confliction with the e-latch power switch.

9.2 Analog Inputs (ADC)

Here you can configure the ADC’s sampling rate (SPS) and Programmable Gain Amplifier (PGA). The current code value and actual voltage on each analog input is also listed here. You can press ENTER to refresh the view.

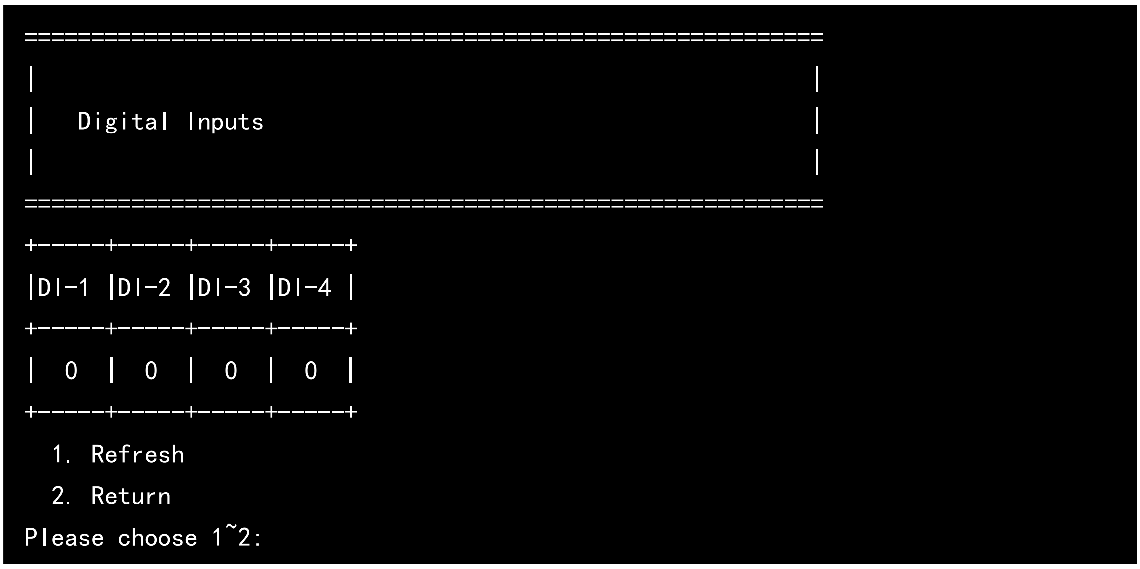


```
============================================================
|                                                          |
|   Analog Inputs (ADC)                                    |
|                                                          |
============================================================
+---------+---------+---------+---------+---------+
|         | ADC-1   | ADC-2   | ADC-3   | ADC-4   |
+---------+---------+---------+---------+---------+
| Scale   |0.001000 |0.001000 |0.001000 |0.001000 |
+---------+---------+---------+---------+---------+
|  Code   |    1    |    1    |    1    |    1    |
+---------+---------+---------+---------+---------+
|  Volt   |0.005684 |0.005684 |0.005684 |0.005684 |
+---------+---------+---------+---------+---------+
  1. Set data rate to 240 SPS [✓]
  2. Set data rate to 60 SPS
  3. Set data rate to 15 SPS
  4. Set data rate to 3 SPS
  5. Set scale to 0.001000000 (PGA=1) [✓]
  6. Set scale to 0.000500000 (PGA=2)
  7. Set scale to 0.000250000 (PGA=4)
  8. Set scale to 0.000125000 (PGA=8)
  9. Return
Please choose 1~9:
```

You can find more information about the analog inputs here.

9.3 Digital Inputs

The current values of digital inputs are listed here. You can press ENTER to refresh the view.

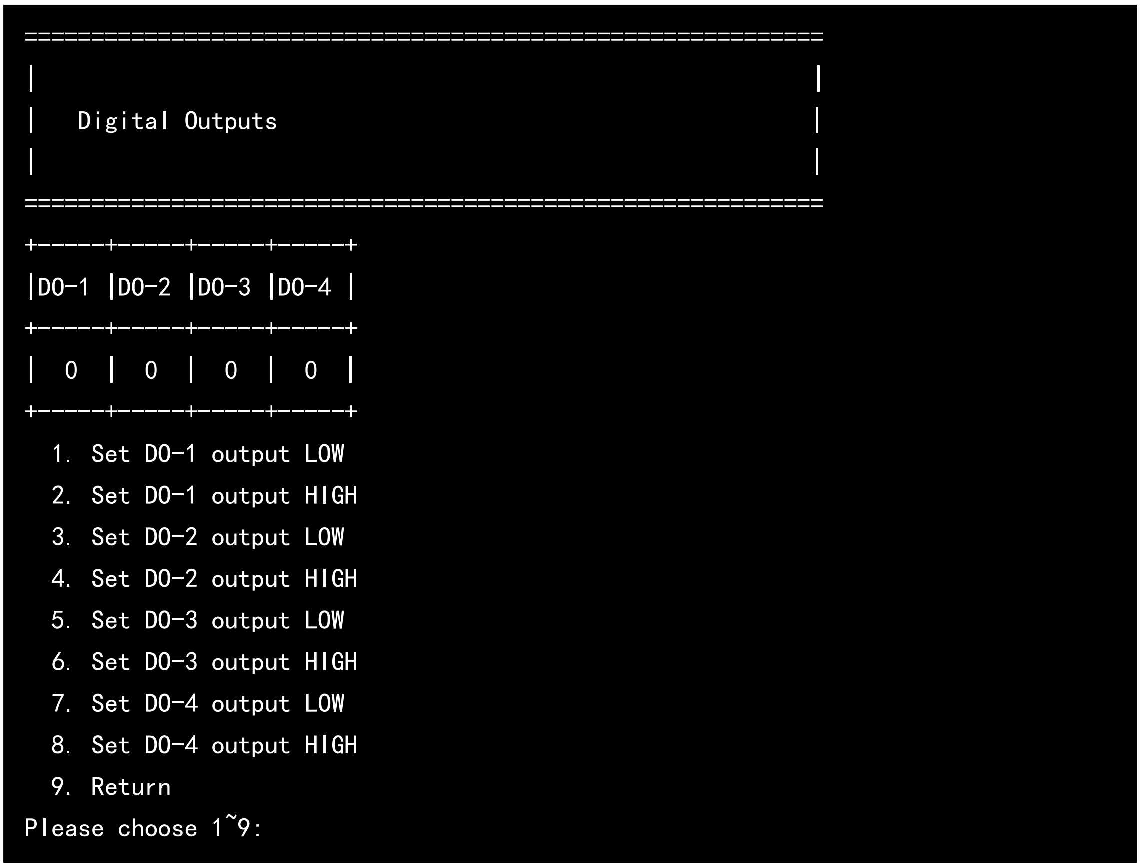


```
============================================================
|                                                          |
|   Digital Inputs                                         |
|                                                          |
============================================================
+-----+-----+-----+-----+
|DI-1 |DI-2 |DI-3 |DI-4 |
+-----+-----+-----+-----+
|  0  |  0  |  0  |  0  |
+-----+-----+-----+-----+
  1. Refresh
  2. Return
Please choose 1~2:
```

You can find more information about the digital inputs here.

9.4 Digital Outputs

Here you control each digital output, and set it to HIGH (1) or LOW (0).

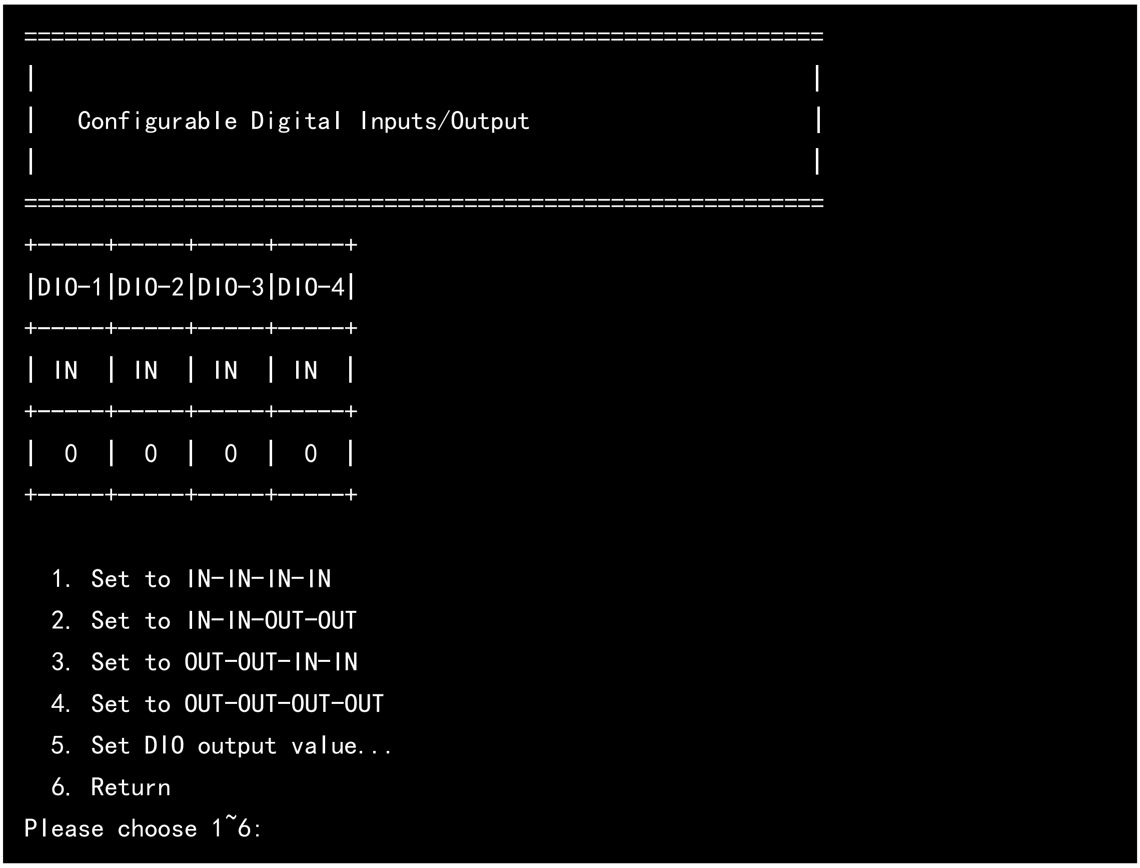


```
============================================================
|                                                          |
|   Digital Outputs                                        |
|                                                          |
============================================================
+-----+-----+-----+-----+
|DO-1 |DO-2 |DO-3 |DO-4 |
+-----+-----+-----+-----+
|  0  |  0  |  0  |  0  |
+-----+-----+-----+-----+
  1. Set DO-1 output LOW
  2. Set DO-1 output HIGH
  3. Set DO-2 output LOW
  4. Set DO-2 output HIGH
  5. Set DO-3 output LOW
  6. Set DO-3 output HIGH
  7. Set DO-4 output LOW
  8. Set DO-4 output HIGH
  9. Return
Please choose 1~9:
```

You can find more information about the digital output here.

9.5 Configurable Digital Inputs/Output

Not every digital input/output (DIO) can be configured to IN or OUT mode separately, however you can choose four different combinations here.



```
============================================================
|                                                          |
|   Configurable Digital Inputs/Output                     |
|                                                          |
============================================================
+-----+-----+-----+-----+
|DIO-1|DIO-2|DIO-3|DIO-4|
+-----+-----+-----+-----+
| IN  | IN  | IN  | IN  |
+-----+-----+-----+-----+
|  0  |  0  |  0  |  0  |
+-----+-----+-----+-----+

  1. Set to IN-IN-IN-IN
  2. Set to IN-IN-OUT-OUT
  3. Set to OUT-OUT-IN-IN
  4. Set to OUT-OUT-OUT-OUT
  5. Set DIO output value...
  6. Return
Please choose 1~6:
```

You can also choose option 5 to set the DIO's output value, given that DIO is set to output mode. More information about the configurable digital inputs/outputs can be found here.

9.6 Programmable LED Test

You can test the programmable LED here. More specifically, you can turn on the red LED by setting GPIO-26 (BCM naming) to HIGH, the yellow LED will turn off when red LED is on. This test will last for one second.

9.7 Buzzer Test

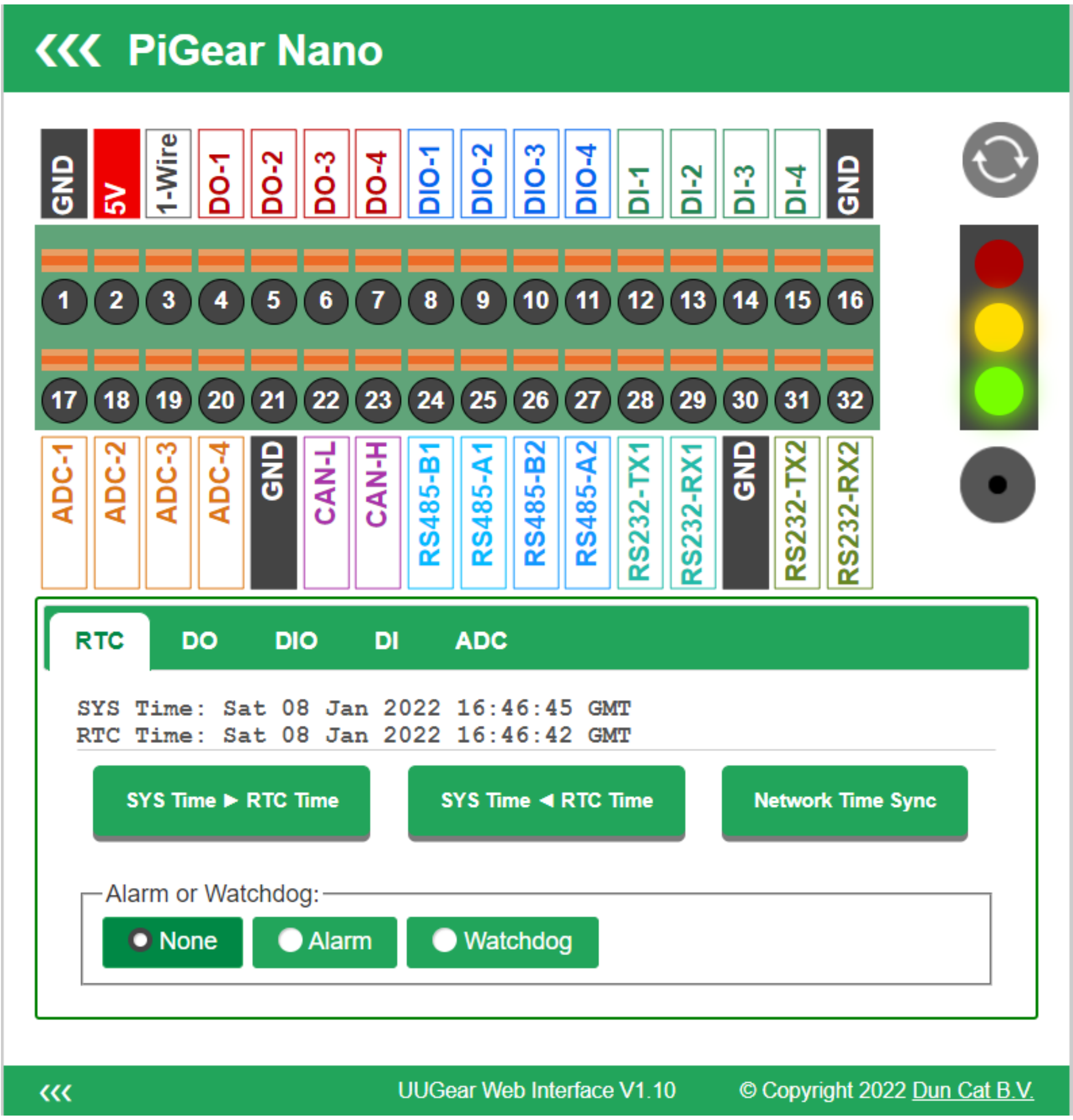
You can test the on-board buzzer here. The buzzer is on when you set GPIO-25 (BCM naming) to HIGH. This test will last for one second.

9.8 UWI Usage

UWI is a mini web server that only uses very little system resource, while it allows you to monitor and configure your board via any device that can access the network.

By default, UWI can be accessed via http://raspberrypi:8000/pgnano/. If you cannot access it, most probably is because the host name “raspberrypi” is not resolved. In such case, please configure your UWI to use actual IP address or accessible host name. You can find more details here.

UWI offers you a very intuitive interface, and here you can do anything that PiGearNano.sh can do. The best part is that it can be used on your mobile phone or tablet.

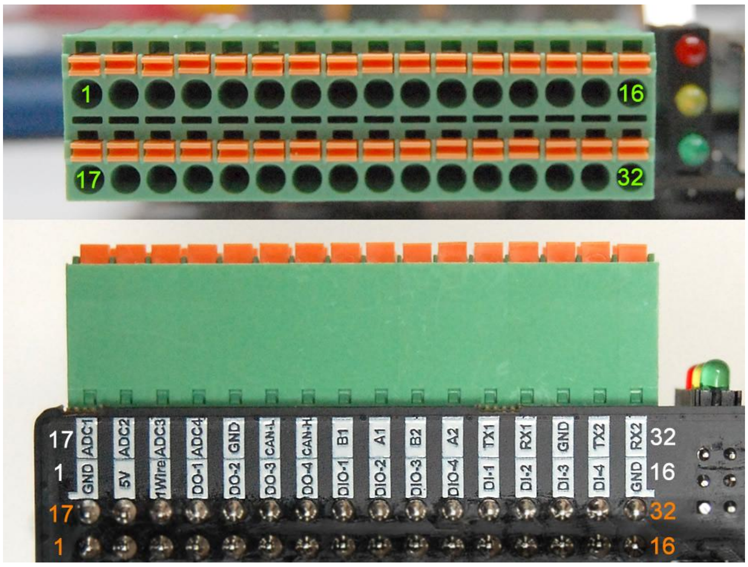


```
SYS Time: Sat 08 Jan 2022 16:46:45 GMT
RTC Time: Sat 08 Jan 2022 16:46:42 GMT
```

Alarm or Watchdog: ● None ○ Alarm ○ Watchdog

10. 2x16 Quick Cable Connector

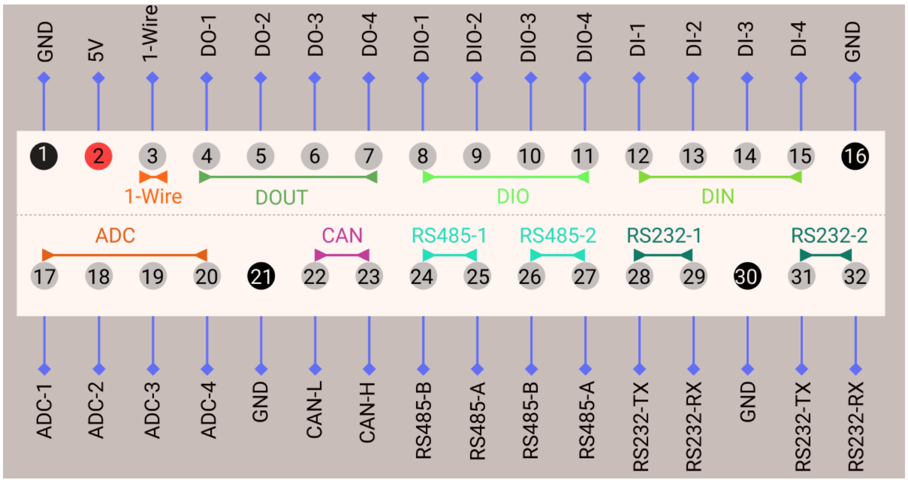
PiGear Nano offers wide range of input/output and communicating interfaces, which are routed to the green 2x16 quick cable connector.



On the back of the board, you will also see the silkscreens that mark the functionality of each pin. **Remarks: the silkscreen has the same layout as the soldering pads (instead of the holes on the connector), please pay attention to those numbers aside.**

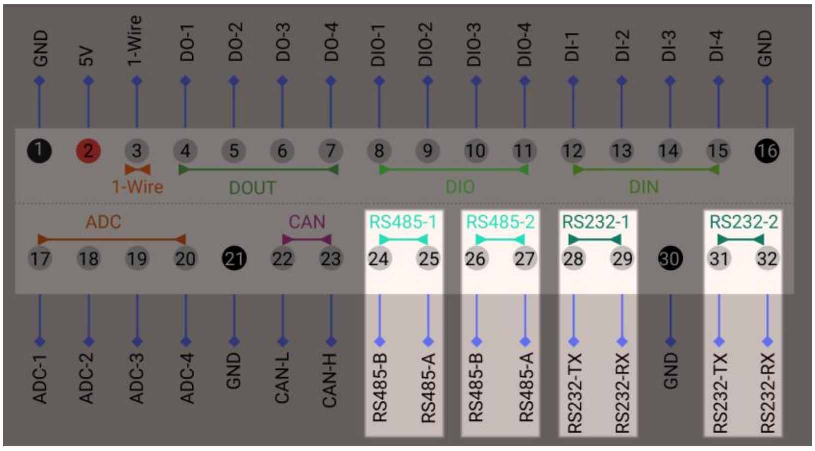
If you want to connect a wire/cable to the connector, you can just directly plug it in to the hole with some pressure and the wire/cable will be clamped automatically. If the wire/cable is too thin/soft, you may want to press the orange button beside the hole so you can insert the wire/cable into the hole easily.

The diagram bellow shows the pinout of this connector:



10.1 Serial Port

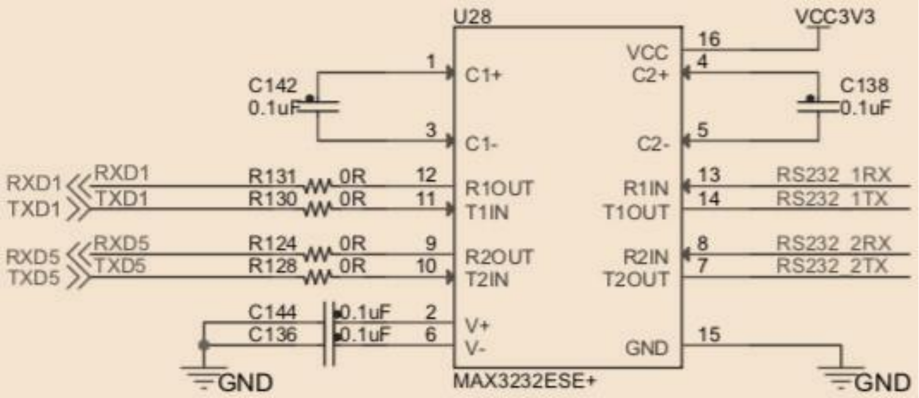
PiGear Nano implements 4 serial ports, and they are accessible via the 2x16 quick cable connector:



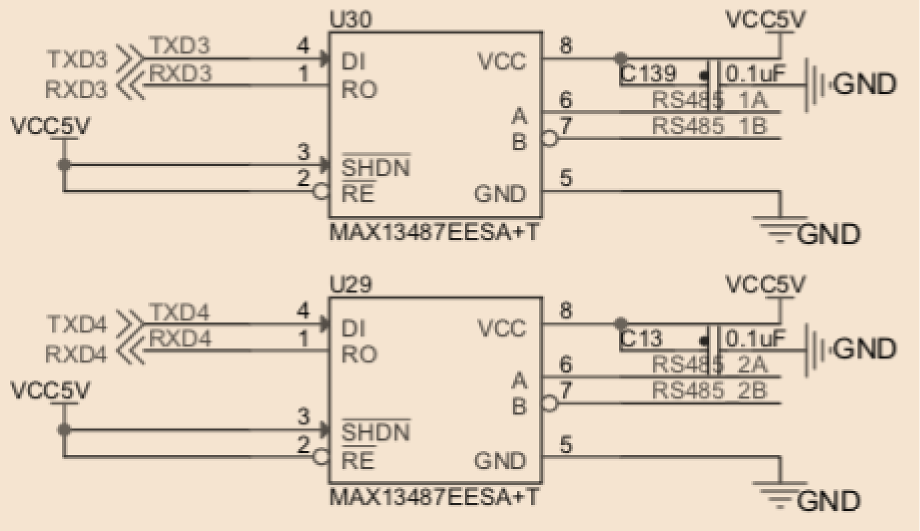
They occupy UART1, UART3, UART4 and UART5, and they aremapped to different devices in the system.

	Name	Mapped Device in System
UART1	RS232-1	/dev/ttyAMA0
UART5	RS232-2	/dev/ttyAMA3
UART3	RS485-1	/dev/ttyAMA1
UART4	RS485-2	/dev/ttyAMA2

Here is the schematic for RS232:



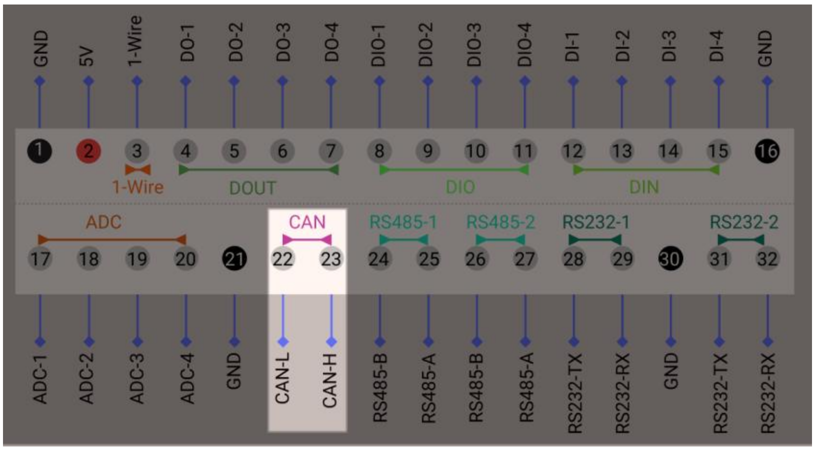
And here is the schematic for RS485:



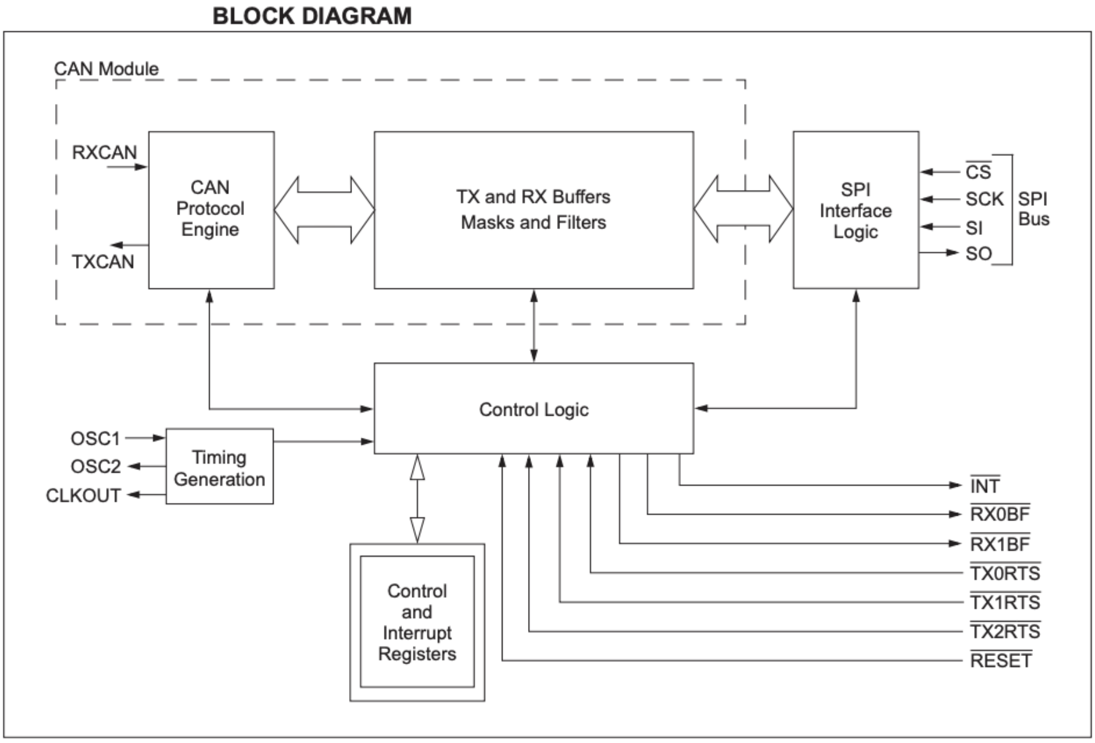
Both RS485 channels work in Half-Duplex mode, and they do not need additional I/O pin to control the transmitting direction.

10.2 Controller Area Network (CAN) Bus

PiGear Nano has one CAN bus, which can be access on pin 22 and 23 in the 2x16 quick cable connector.

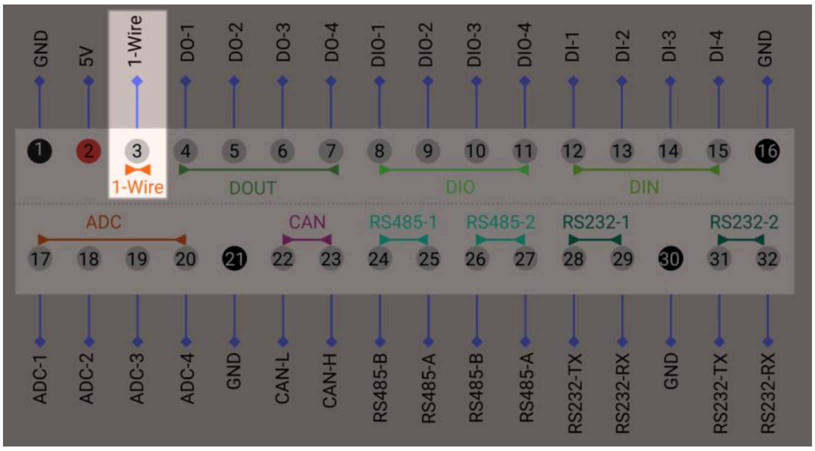


PiGear Nano uses chip MCP2515 to provide CAN controller, which talks to Raspberry Pi Compute Module 4 via SPI interface.



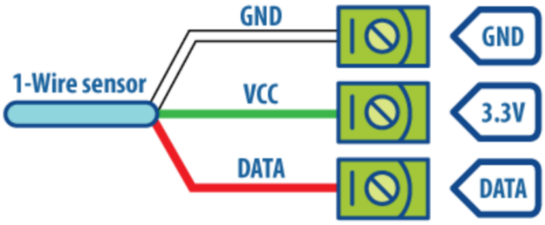
10.3 1-Wire Interface

PiGear Nano has 1-wire channel on pin 3 in the 2x16 fast cable connector.



PiGear Nano uses chip DS2482 to implement a single-channel 1-wire master device, which talks to Raspberry Pi Compute Module 4 via the I2C-1 interface.

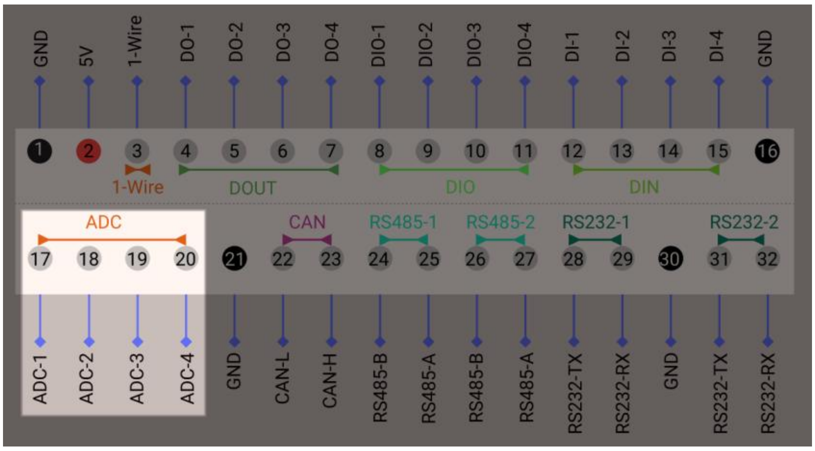
Many 1-wire devices actually have 3 wires: one for transmitting data, which should be connected to pin 3, and the other two are VCC and GND, for powering the sensor. Below is a wiring example:



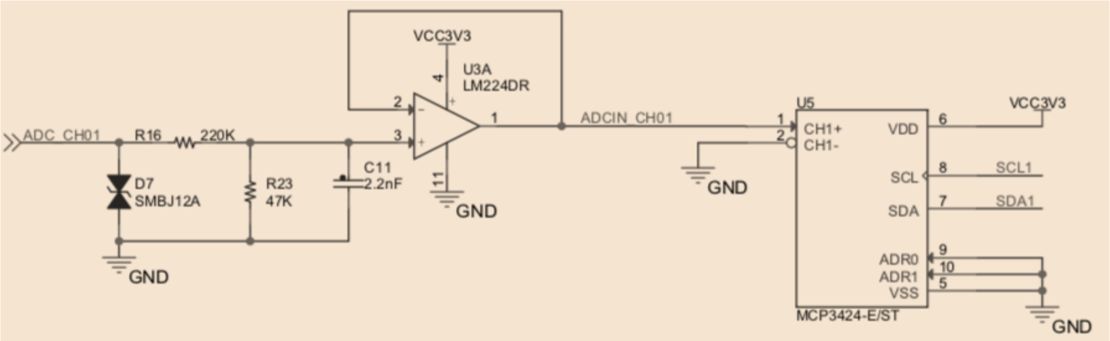
Some 1-wire device can be powered by the voltage on data line, and doesn't have the VCC wire.

10.4 Analog Inputs (A/D Converter)

PiGear Nano has four analog inputs:



PiGear Nano uses chip MCP3424 to provide four-channel, low-noise, high accuracy A/D converter with up to 18-bit resolution. The A/D converter talks to Raspberry Pi Compute Module 4 via the I2C-6 interface.



As shown in the schematic above, each channel has a voltage divider and a voltage follower before feeding to MCP3424. Because MCP3424 uses 2.048V internal voltage reference, each channel can directly measure up to 11.6V.

Remarks: analog input can directly apply up to 11.6V, and needs to have common ground with PiGear Nano.

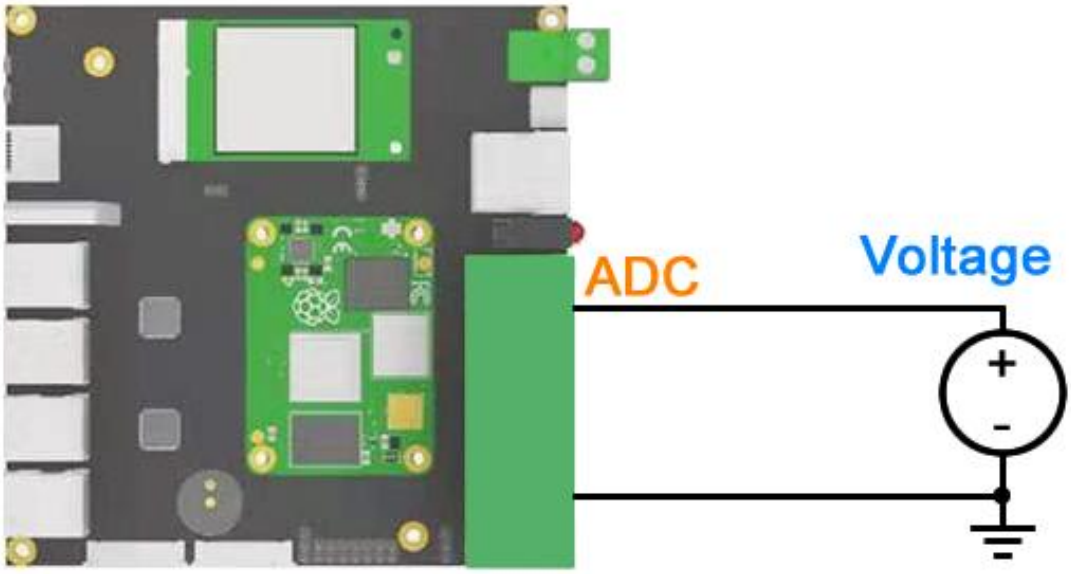
MCP3424 has on-board Programmable Gain Amplifier (PGA). You can select the PGA gain of x1, x2, x4, or x8 before the analog-to-digital conversion takes place. This allows the MCP3424 devices to convert a very weak input signal with high resolution.

The resolution of ADC is configurable, and different resolution has different LSB, data rate and range of code. Please see the table below for more details.

Resolution	LSB	Data Rate	Minimum Code	Maximum Code
12 bits	1 mV	240 SPS	0	2047
14 bits	250uV	60 SPS	0	8191
16 bits	62.5uV	15 SPS	0	32767
18 bits	15.625uV	3.75 SPS	0	131071

There is an 8-bit configuration register in MCP3424, and you can make configurations by changing specific bits. More details can be found in MCP3424's datasheet.

Here is an example for measuring voltage up to 11.6V:



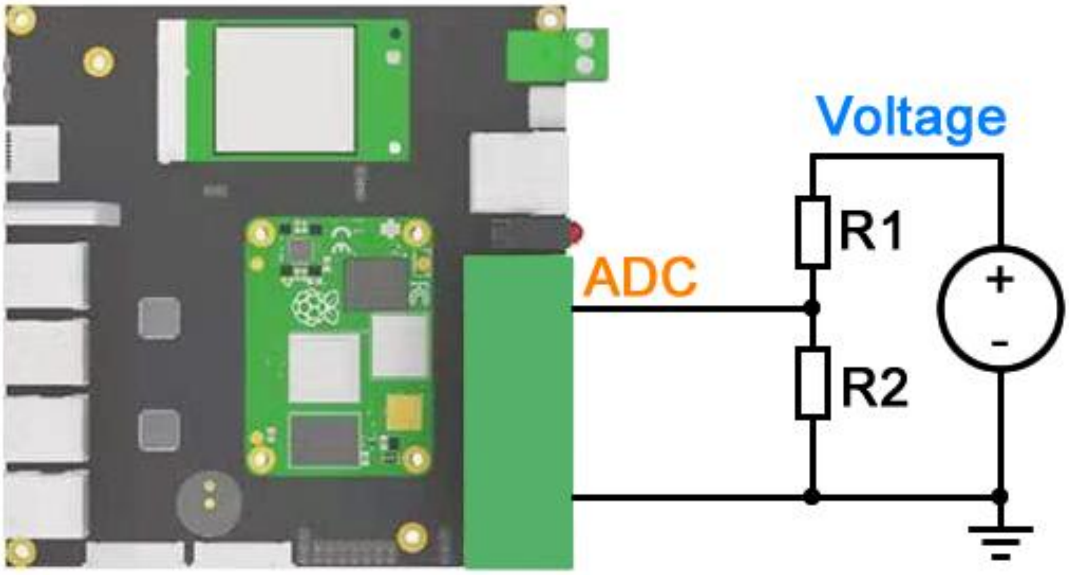
You can calculate the measured voltage with this formula:

$$V_{in} = \frac{(220 + 47) \times Code_{output} \times 2.048}{47 \times (Code_{max} + 1) \times PGA} = \frac{Code_{output} \times 11.634}{(Code_{max} + 1) \times PGA}$$

If PGA is set to x1, LSB is set to 1mV (12-bit resolution), and the ADC output code is 1712, then the measured voltage is:

$$V_{in} = \frac{1712 \times 11.634}{(2047 + 1) \times 1} \approx 9.726\,V$$

If you want to measure voltage higher than 11.6V, then you will need to implement your own voltage divider.



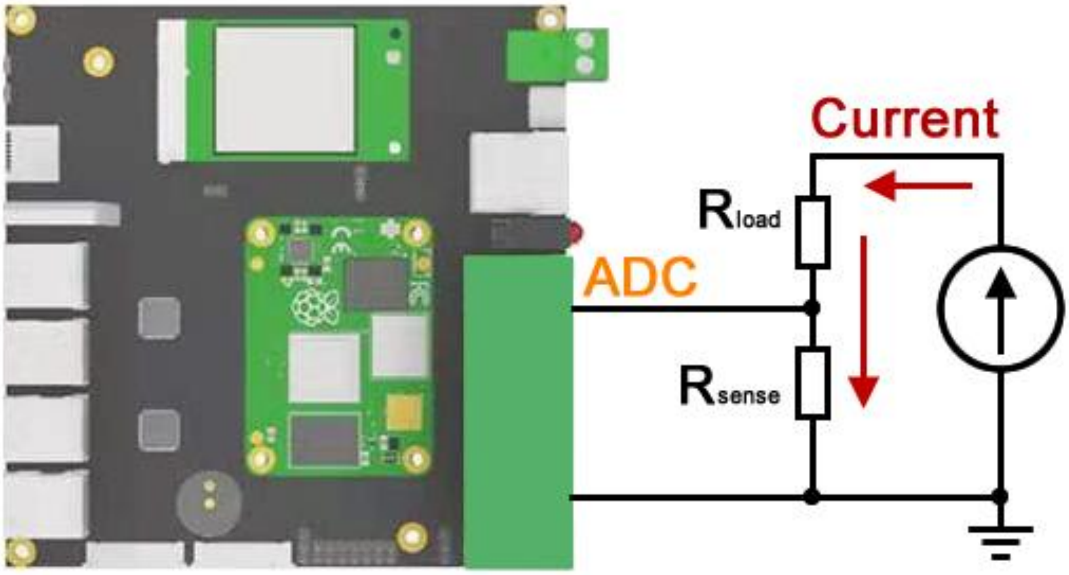
R1 and R2 should be properly chose so the actual voltage applies to ADC is no higher than 11.6V.

In this case, the formula is changed to:

$$V_{in} = \frac{(R_1 + R_2) \times (220 + 47) \times Code_{output} \times 2.048}{R_2 \times 47 \times (Code_{max} + 1) \times PGA}$$

$$= \frac{(R_1 + R_2) \times Code_{output} \times 11.634}{R_2 \times (Code_{max} + 1) \times PGA}$$

If you want to measure the current, you will need to connect a sense resistor with the load in serial, so the current will generate some voltage drop on the sense resistor. The ADC can measure that voltage drop and calculate the current, given the value of sense resistor is known already.

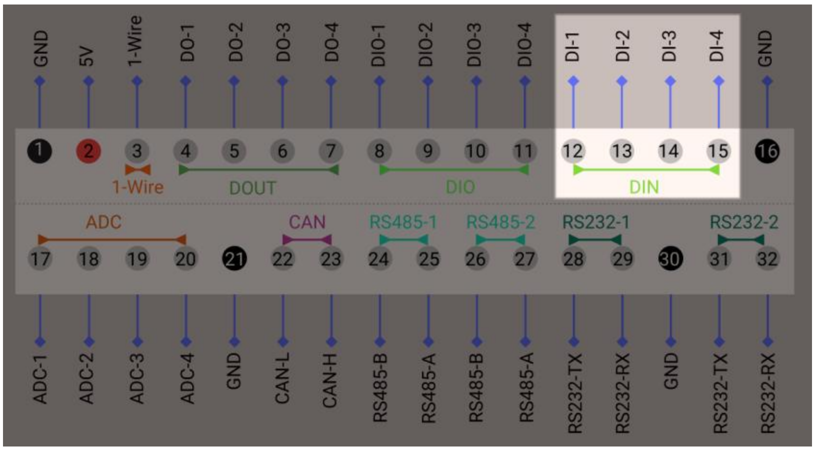


Usually R_{sense} is a rather small resistor. If R_{sense} is 0.01 Ω, and the voltage drops on R_{sense} is 0.0125V, the current can be calculated like this:

$$\mathrm{I} = \frac{V_{drop}}{R_{sense}} = \frac{0.0125}{0.01} = 1.25\ A$$

10.5 Digital Inputs

PiGear Nano has four digital inputs.



The four digital inputs are implemented by chip PCF8574, which talks to Raspberry Pi Compute Module 4 via the I2C-1 interface. Each digital input can bear up to 12V. Voltage below 0.5V is considered as LOW.

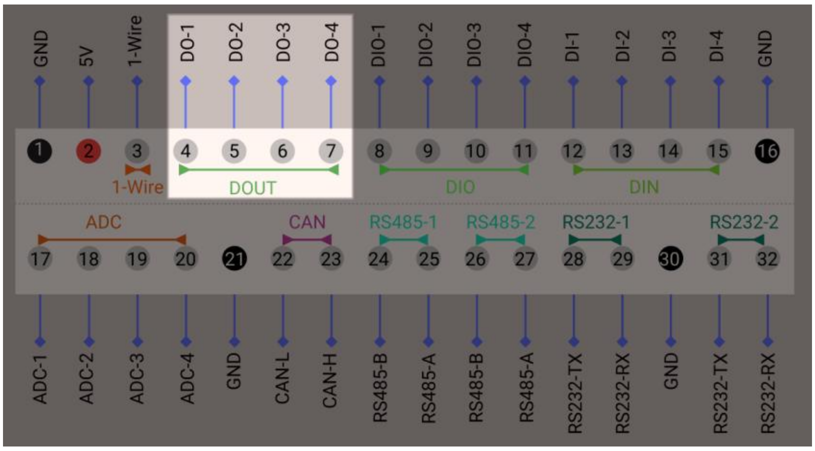
Note: all four digital inputs are pulled down by default.

You can inquire the pin status via I2C-1 (address=0x20, omitting register index), and the pin status are stored in bit 0~3 in the register.

Bit	7	6	5	4	3	2	1	0
Value	X	X	X	X	0 / 1	0 / 1	0 / 1	0 / 1
Pin	N/A	N/A	N/A	N/A	DI-4	DI-3	DI-2	DI-1

10.6 Digital Outputs

PiGear Nano has 4 digital outputs.

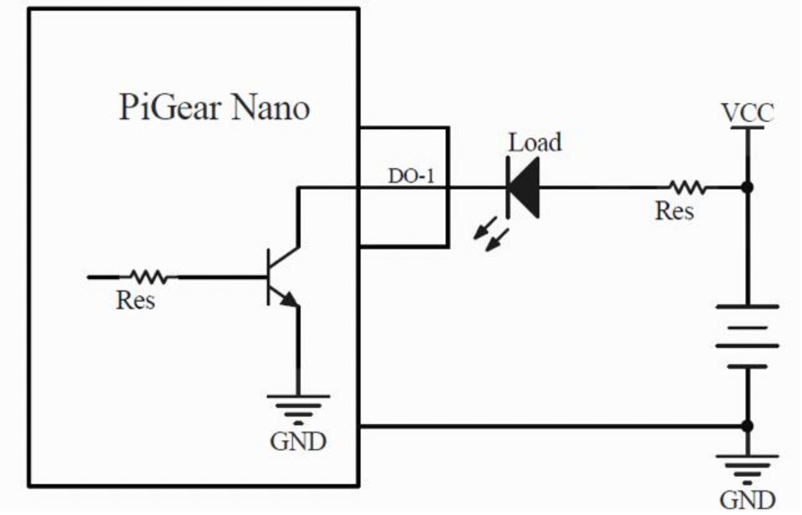


These 4 digital outputs are open drain outputs and they are driven by GPIO-11, GPIO-16, GPIO-17 and GPIO-22 in Raspberry Pi Compute Module 4 (please see table below).

CM4 GPIO	GPIO-11	GPIO-16	GPIO-17	GPIO-22
PiGear Nano D$_{OUT}$	DO-1	DO-2	DO-3	DO-4

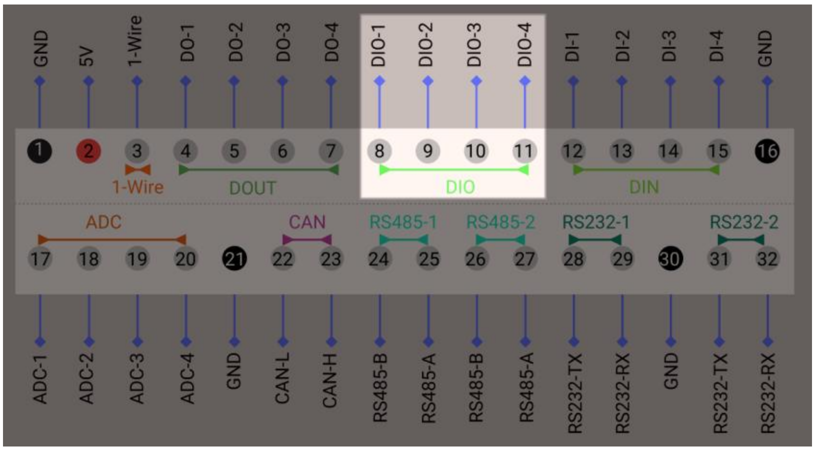
Remarks: every time you change the output value, you need to give a rising edge on bit-2 in the register in I2C-6 (address=0x20).

Each output can bear up to 30V DC and can drain up to 500mA current. Below is an example to drive a load with digital output.



10.7 Configurable Digital Inputs/Outputs

PiGear Nano also comes with four configurable digital I/O pins.



These four pins can be configured as either inputs or outputs. DIO-1 and DIO-2 are in one group, while DIO-3 and DIO-4 are in another group. The pins in the same group must be in the same input/output mode. Therefore, there are four possible configurations:

	DIO-1	DIO-2	DIO-3	DIO-4
Configuration 1	INPUT	INPUT	INPUT	INPUT
Configuration 2	INPUT	INPUT	OUTPUT	OUTPUT
Configuration 3	OUTPUT	OUTPUT	INPUT	INPUT
Configuration 4	OUTPUT	OUTPUT	OUTPUT	OUTPUT

When the pin is in OUTPUT mode, it works as open drain output, which can bear up to 12V and drain up to 500mA current.

The implementation uses two pieces of PCF8574 chips and they are connected to two I2C buses: I2C-1 and I2C-6. Both chips have the same 0x20 I2C address. The one on I2C-6 works as an internal controller and the one on I2C-1 provides the 4 DIO channels (mapped to bit 4~7 in I2C register).

10.7.1 Configuration 1 (four inputs):

In order to configure all of them as inputs, you need to set the register in both chips to 0xF0:

Bus	Address	Register Index	Set to Value
I2C-1	0x20	omitted	0xF0 (B11110000)
I2C-6	0x20	omitted	0xF0 (B11110000)

Then you can inquire the pin status via I2C-1 (address=0x20, omitting register index), and the pin status are stored in bit 4~7 in the register.

Bit	7	6	5	4	3	2	1	0
Value	0 / 1	0 / 1	0 / 1	0 / 1	X	X	X	X
Pin	DIO-4	DIO-3	DIO-2	DIO-1	N/A	N/A	N/A	N/A

Remarks: GPIO-23 (BCM naming) will be pulled down when new data is available on inputs, and you can monitor GPIO-23 and know when to query the new data.

10.7.2 Configuration 2 (DIO-1 and DIO-2 as inputs; DIO-3 and DIO-4 as outputs):

In order to set DIO-1 and DIO-2 as inputs, while DIO-3 and DIO-4 as outputs; you need to set the register in I2C-1 to 0x30, set the register in I2C-6 to 0xF8 and then 0xFA. This way you give bit 2 a raising edge to make the change take effect.

Bus	Address	Register Index	Set to Value
I2C-1	0x20	omitted	0x30 (B00110000)
I2C-6	0x20	omitted	0xF8 (B11111000) and then 0xFA (B11111010)

Then you can inquire the pin status via I2C-1 (address=0x20, omitting register index), and the pin status are stored in bit 4 and 5 in the register. You can also set the bit 6 and 7 to change the output state on DIO-3 and DIO-4.

Bit	7	6	5	4	3	2	1	0
Value	0 / 1	0 / 1	0 / 1	0 / 1	X	X	X	X
Pin	DIO-4	DIO-3	DIO-2	DIO-1	N/A	N/A	N/A	N/A

Remarks: GPIO-23 (BCM naming) will be pulled down when new data is available on inputs, and you can monitor GPIO-23 and know when to query the new data.

10.7.3 Configuration 3 (DIO-1 and DIO-2 as outputs; DIO-3 and DIO-4 as inputs):

In order to set DIO-1 and DIO-2 as outputs, while DIO-3 and DIO-4 as inputs, you need to set the register in I2C-1 to 0xC0, set the register in I2C-6 to 0xF4 and then 0xF6. This way you give bit 2 a raising edge to make the change take effect.

Bus	Address	Register Index	Set to Value
I2C-1	0x20	omitted	0xC0 (B11000000)
I2C-6	0x20	omitted	0xF4 (B11110100) and then 0xF6 (B11110110)

Then you can inquire the pin status via I2C-1 (address=0x20, omitting register index), and the pin status are stored in bit 6 and 7 in the register. You can also set the bit 4 and 5 to change the output state on DIO-1 and DIO-2.

Bit	7	6	5	4	3	2	1	0
Value	0 / 1	0 / 1	0 / 1	0 / 1	X	X	X	X
Pin	DIO-4	DIO-3	DIO-2	DIO-1	N/A	N/A	N/A	N/A

Remarks: GPIO-23 (BCM naming) will be pulled down when new data is available on inputs, and you can monitor GPIO-23 and know when to query the new data.

10.7.4 Configuration 4 (four outputs):

In order to configure all of them as outputs, you need to set the register in I2C-1 to 0x00, set the register in I2C-6 to 0xFC and then set to 0xFE.This way you give bit 2 a raising edge to make the change take effect.

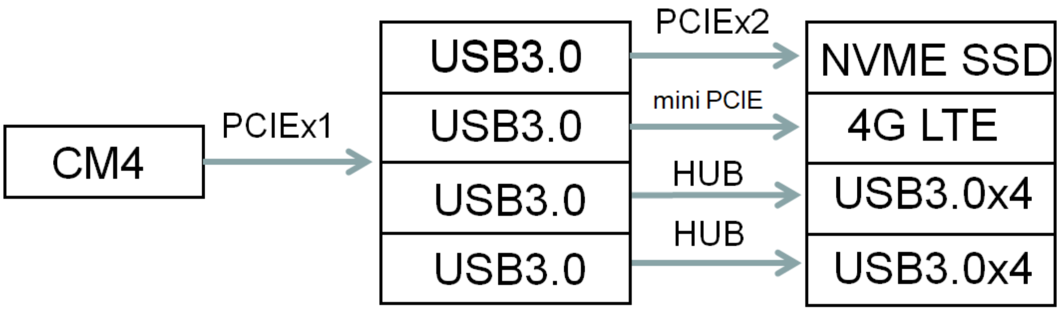
Bus	Address	Register Index	Set to Value
I2C-1	0x20	Omitted	0x00 (B00000000)
I2C-6	0x20	omitted	0xFC (B11111100) and then 0xFE (B11111110)

Then you set the output pin status via I2C-1 (address=0x20, omitting register index). More specifically set the value of bit 4 ~7 to change the output state on DIO-1, DIO-2, DIO-3 and DIO-4.

Bit	7	6	5	4	3	2	1	0
Value	0 / 1	0 / 1	0 / 1	0 / 1	X	X	X	X
Pin	DIO-4	DIO-3	DIO-2	DIO-1	N/A	N/A	N/A	N/A

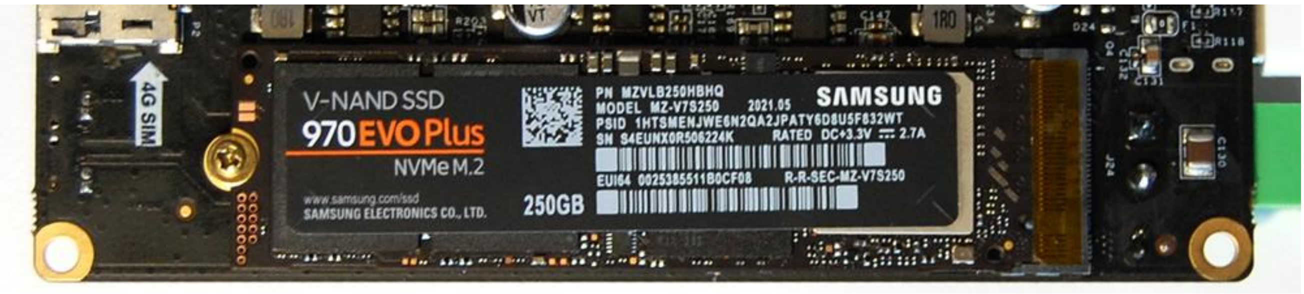
11. PCIE

PiGear Nano extends 4 channels of USB3.0 interfaces from the PCIE on Raspberry Pi Compute Module 4, and then extends a PCIE Gen3 x2 (via RTL9210B-CG) for NVME SSD, a Mini PCIE for 4G LTE module, and also 8 USB 3.0 ports.



11.1 NVME SSD

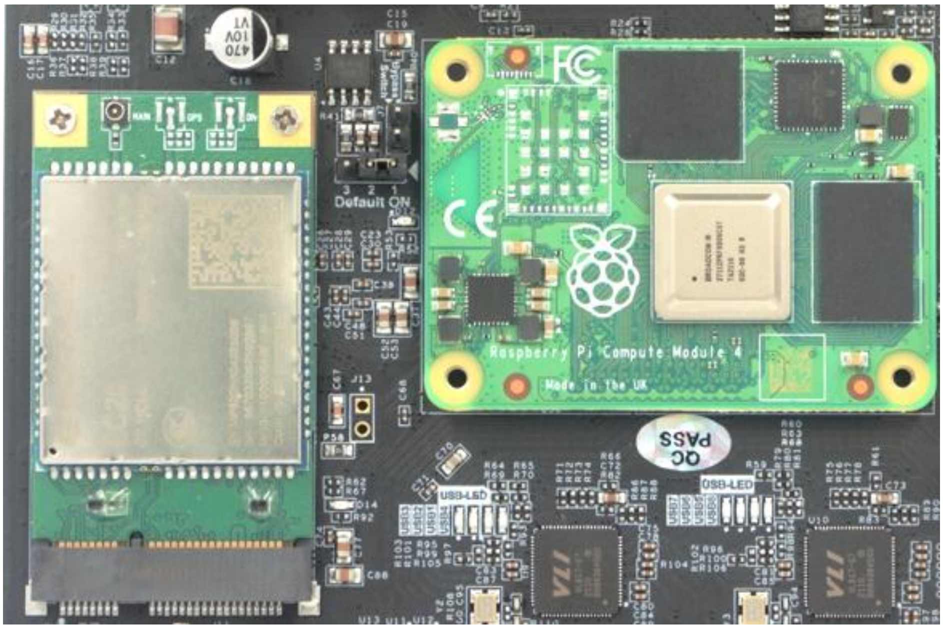
There is a NVME SSD M.2 connector on the back of PiGear Nano board, and you can connect an M.2-2280 NVME SSD to it.

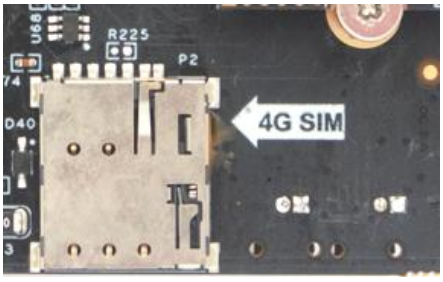


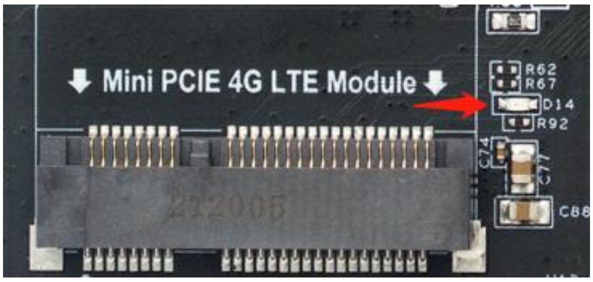
There is a green LED (D12) on the top surface of PiGear Nano board, and it will blink when SSD is transmitting data.

11.2 4G LTE Module

There is a Mini PCIE 4G LTE module connector on PiGear Nano board, and there is also a SIM card slot underneath. The green LED D14 works as 4G LTE module's status indicator.







If your 4G LTE module doesn't come with SIM card slot, you can insert the SIM card into the SIM card slot on board.

If your 4G LTE module already has SIM card slot, you can insert the SIM card into either SIM card slot: the one on 4G LTE module or the one on PiGear Nano board.

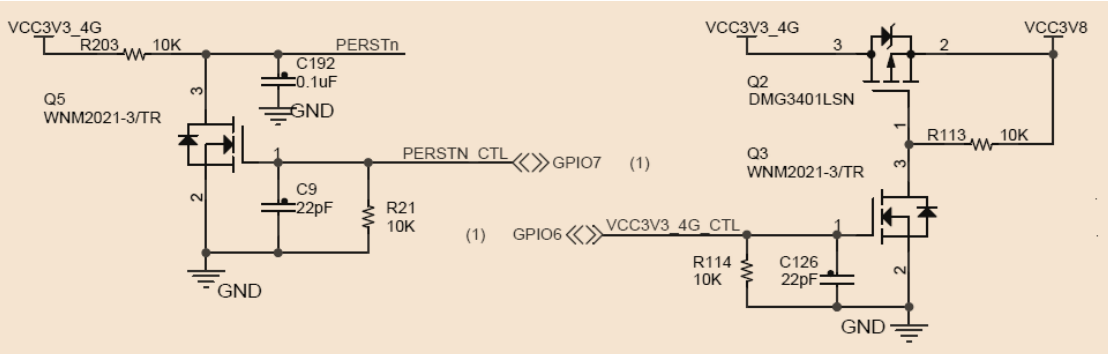
Remarks: connecting two SIM cards to both SIM card slots will not work.

PiGear Nano uses GPIO-6 to control the power supply to the 4G LTE module, and uses GPIO-7 to reset the module. If you install the software for PiGear Nano, the daemon.sh script will automatically configure GPIO-6 and GPIO-7 to enable the 4G LTE module (if present). If you want to use it without installing the software, you will need to set GPIO-6 to HIGH and GPIO-7 to LOW with these two commands:

```
pi@raspberrypi:~ $ sudo raspi-gpio set 6 op pn dh

pi@raspberrypi:~ $ sudo raspi-gpio set 7 op pn dl
```

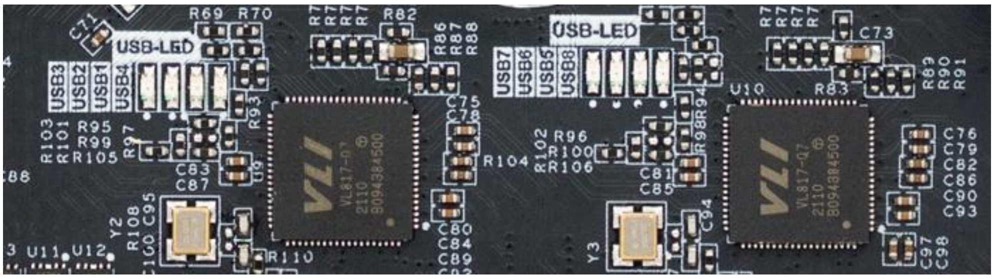
Remarks: you will then need to wait 5~10 seconds before running the "lsusb" command to see the 4G LTE module gets recongized.

When 4G LTE module is used in a remote site, and network is lost due to any reason, you may want to reset the 4G LTE module or even make a full power cycle to get it work again. You can set GPIO-7 to HIGH to reset the 4G LTE module, if it still doesn't work properly. You can set GPIO-6 to LOW to cut the power for 4G LTE module, and then set GPIO-6 back to HIGH to power the 4G LTE module again. This can also be done by a program running in the background, who checks the network with a fixed inteval and will reset/power cycle the 4G module when the network is not good.



11.3 USB3.0

PiGear Nano comes with 8 USB 3.0 ports, and each of them has its own activitiy indicator (green LED) near the VL817 chips.



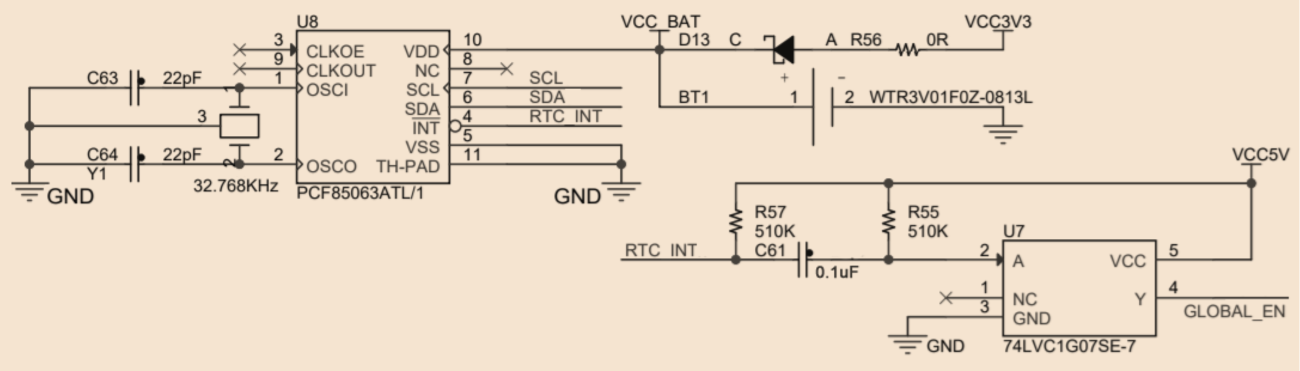
Limited by Raspberry Pi CM4's PCI Express Gen 2 lane, these 8 USB 3.0 ports share a total of 4Gbps of data bandwidth. However, there are scenarios that need to connect multiple USB 3.0 devices at the same time, while they don't have to work simultaneously. Providing 8 USB 3.0 ports indeed unlock many possibilities and offers more flexibility.

Four stacked USB connectors are used and each of them has a dedicated 1A current limiter (SY6280AAAC configured as 1A limiter), which means every two stacked USB ports can output up to 1A current (given the power supply can offer that current).

If you will use multiple USB devices or there will be a lot of data transation via USB bus, it is recommended to put the small heatsink (included in the package) on the VL805 chip (on the back of the board).

12. Realtime Clock (RTC)

PiGear Nano uses PCF85063 realtime clock for system time keeping, and the RTC talks to Raspberry Pi Compute Module 4 via I2C-1 interface.



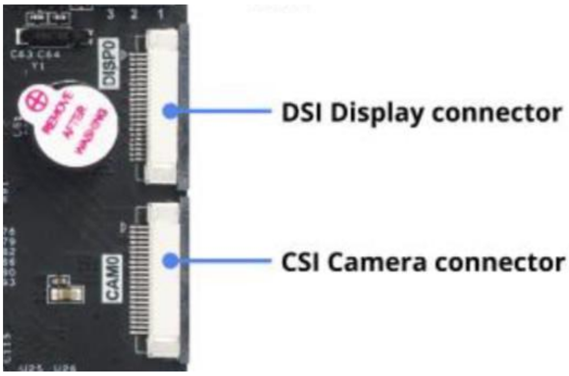
PCF85063 has very small quiescent current (0.22µA) and PiGear Nano uses a 3V 1F supper capacitor (BT1) for off-power time keeping. There is no need to use battery to keep the system time when the board is not powered.

PCF85063 has an alarm. When this alarm is triggered, it will pull down GLOBAL_EN on Raspberry Pi Compute Module 4, and will reboot the system. If the system has been shut down and the power is not cut (e-latch power switch is bypassed), this alarm can wake up the system again.

The software provides the options to use this alarm to reboot/wake up the system, or use it as a watchdog. You can find more details here.

13. DSI and CSI

PiGear Nano has one DSI connector (DISP0) and one CSI connector (CAM0).



Different than the official Raspberry Pi Compute Module 4 I/O board, PiGear Nano uses I2C-1 instead of I2C-0 for DSI and CSI.

If you will use them, please run these commands in home directory and then reboot:

```
pi@raspberrypi:~ $ sudo cp pgnano/bin/dt-blob.bin /boot/dt-blob.bin

pi@raspberrypi:~ $ sudo chown root:root /boot/dt-blob.bin

pi@raspberrypi:~ $ sudo chmod 755 /boot/dt-blob.bin
```

You can run this command to test your camera connected to CAM0:

```
pi@raspberrypi:~ $ sudo raspivid -t 0
```

14. SD Card Slot

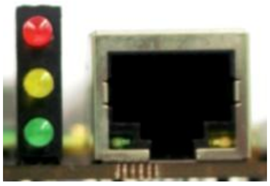
There is an SD card slot on PiGear Nano board. You can insert a micro SD card into it.

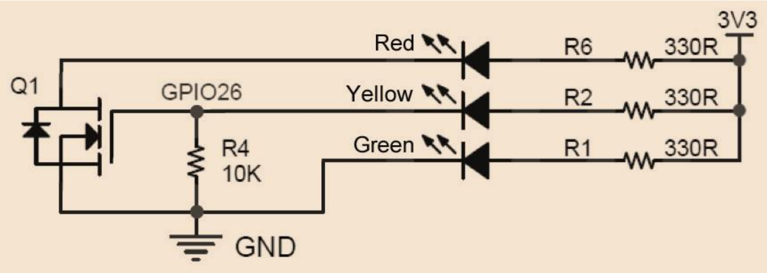
It is used only when you connect Raspberry Pi Compute Module 4 Lite (without eMMC) to PiGear Nano. In case your Raspberry Pi Compute Module 4 comes with eMMC, the SD card in this slot will be ignored.

15. LED Indicators

There are three 3mm LEDs aside the Ethernet connector, and they are in color red, yellow and green. The green LED serves as power indicator and will light up when PiGear Nano is turned on.

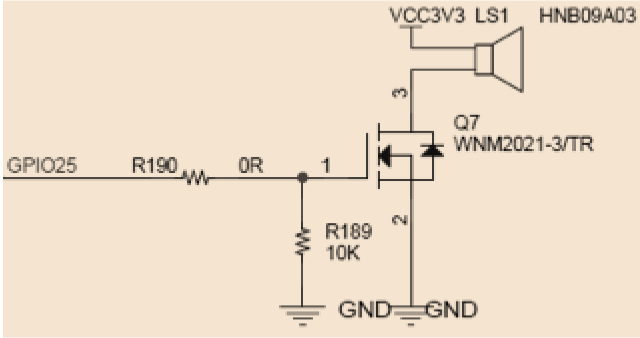
The red and yellow LEDs are both controlled by GPIO-26 (BCM naming). If GPIO-26 outputs HIGH, red LED will light up (while yellow LED is off); if GPIO-26 outputs LOW, yellow LED will light up (while red LED is off).





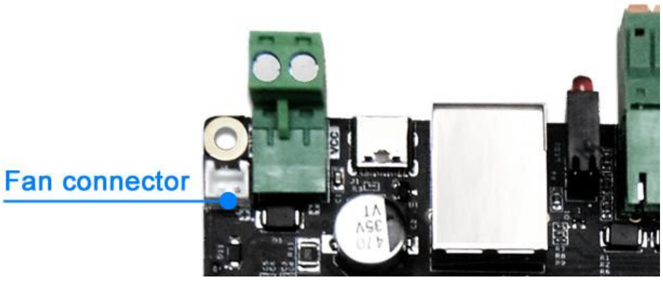
16. Buzzer

There is an on-board buzzer beside the DISP0 connector. This buzzer is controlled by GPIO-25 (BCM naming). If GPIO-25 outputs HIGH, the buzzer will turn on.



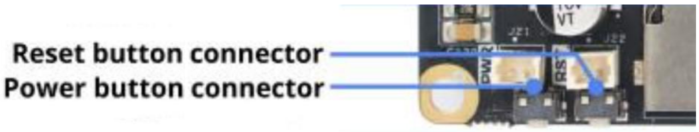
17. Fan Connector

There is a fan connector near the green power input connector. It is a 2-pin PH2.0 connector and you can connect a 5V fan to it. This connector can not control the fan speed.



18. Power and Reset Button Connectors

Aside the power button and reset button, there are two small connectors. Which allow you to connect external momentary push buttons and route them to different location.



19. Case Kit for PiGear Nano

It is possible to put PiGear Nano into a case. We have designed a black case kit for PiGear Nano, and the STL files will be published soon. You may also order this case kit from our e-shop.

20. Revision History

Revision	Date	Description
1.00	2022.01.09	Initial revision
1.01	2022.01.21	Fix some typo. Add description about how to enable 4G LTE module. Add description about the small heatsink in the package. Add description about fan connector and power/reset button connectors.
1.02	2022.01.24	Add more details about enabling 4G LTE module. Add more details about USB 3.0.
1.03	2022.01.25	Add note about powering via USB type C connector. Add information about case kit.
1.04	2022.01.26	Correct the USB output current limit to 1A for every two ports. Add output current information for internal DC/DC converter. Mention using RTL9210B-CG to extend PCIEx2 from USB 3.0.