Orbitool O2S v3.3

Author: Dr. Róbert Lőrincz

Should a tool-board *magic smoke* come out because of user error? Why?

Should a tool-board be destroyed by a fault in your 3D printer? Why?

My answer is NO! And this is what this tool-board is all about!

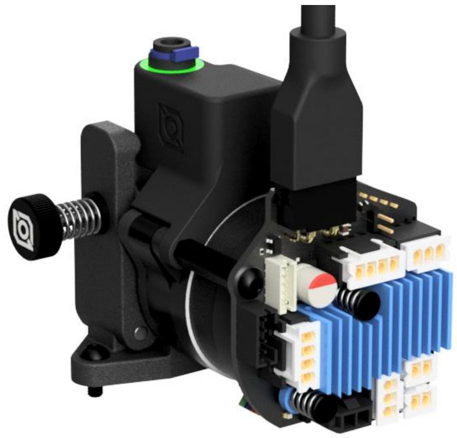
We are hobbyists and professionals, and let's face it, we make mistakes, and when the magic smoke shows up, everybody is disappointed, even if we know it's a user's fault. In this board, I promise you the *magic smoke* is kept inside extremely tight!

Smokeless Features

- Optimized for Orbiter v2 shape and features and not only
- STM32F072 microcontroller running at 48MHz
- Automotive USB communication with a Raspberry Pi
- Second onboard USB port
- Onboard ADXL345 accelerometer
- TMC2240 extruder stepper driver
- Direct connection to the Orbiter v2 sensor
- 4 wire-controlled part fan output with configurable supply voltage
- DC-DC converter based hot-end fan driver, compatible with 12/24V fan types with RPM speed input signal
- Hot-end temperature sensor input compatible with standard NTC or PT1000 temperature sensor types
- I2C port for bed scanner
- X-Stop sensor input
- RGB LED driver output
- Onboard temperature sensor
- Advanced thermal management

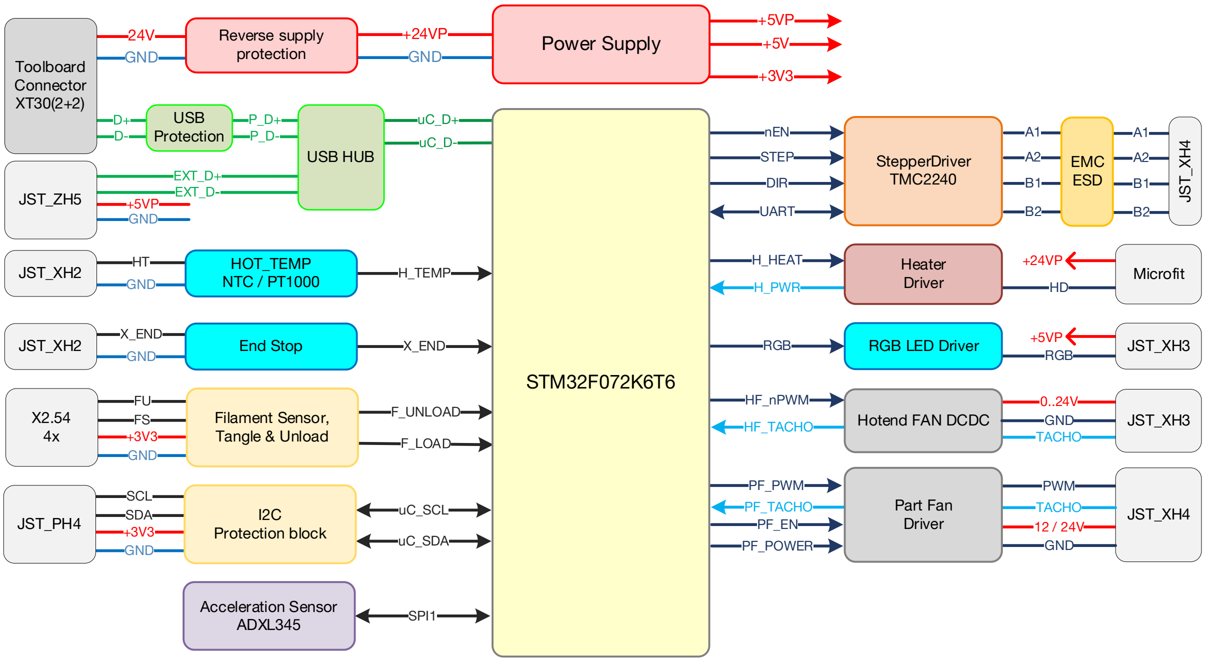
Protection Features

- Active short-circuit protected Hot-end output
- Active short circuit protected fan driver outputs
- Active protection circuit against reverse power supply connection
- Protection against loss of GND supply
- Analog and digital inputs protected against short to +24V supply voltage
- USB data lines are protected against short circuits to GND and +24V
- RGB LED power supply pin protected against short to GND
- EMI interference and ESD protection for all inputs and outputs
- Heater thermal runaway protection in case of a short between heater and sense thermistor wires

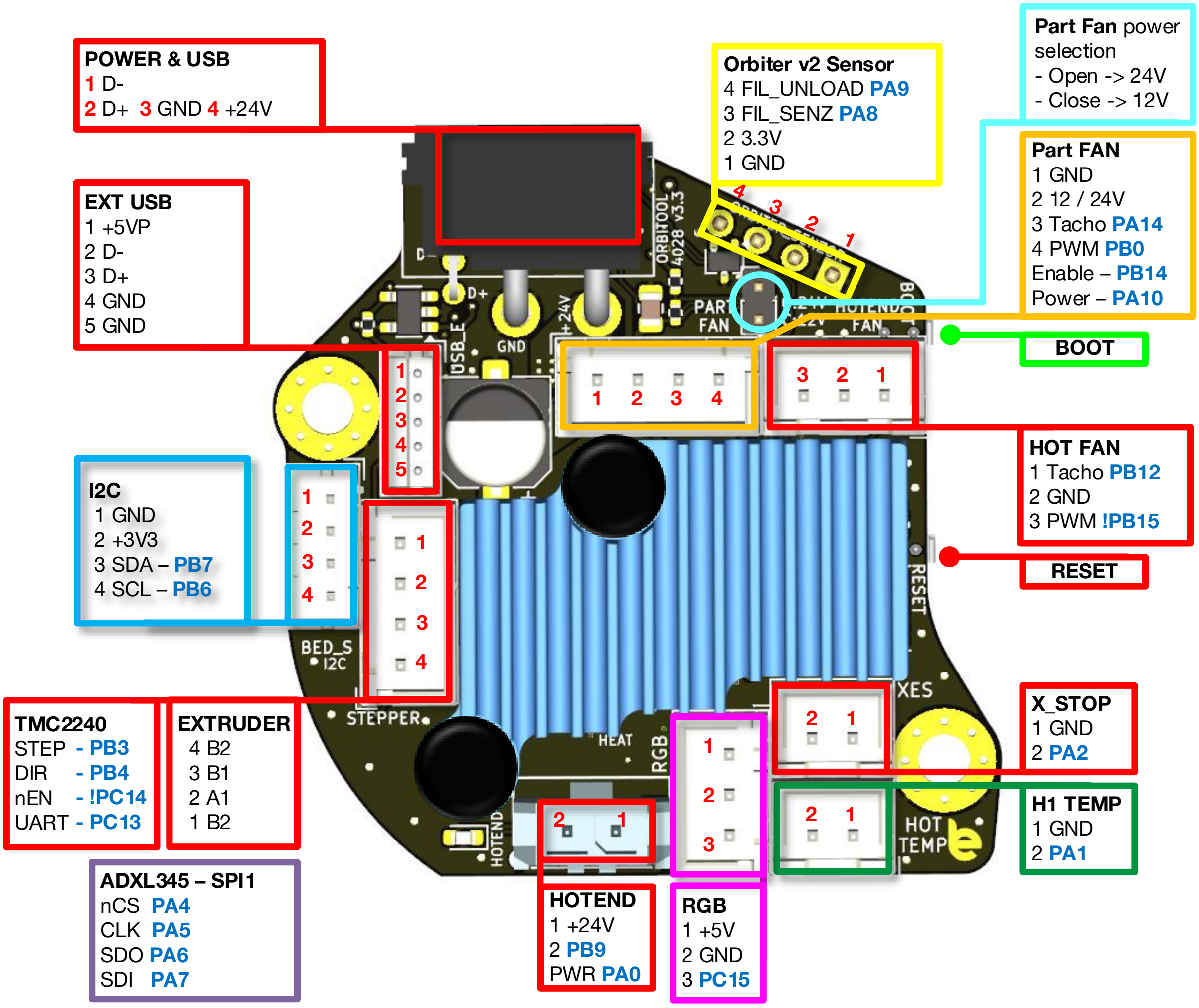


1 Block Diagram

Block diagram of the Orbitool 02S v3.3



2 Pinout definition



3 Features description

3.1 Introduction

For me, the electrical robustness of a design is part of the requirements. One of my main design goals was to integrate protection against common user errors and 3D printer defects.

In the end, nothing is more frustrating than powering up your new board and letting the *magic smoke* out. Even if it is due to user fault, the disappointment will still be present.

This design is based on automotive technology, where protection from all kinds of hazards is state-of-the-art a long time ago.

This document describes the main specific features of the tool-board.

Main changes compared to previous toolboard version v3.3:

- Secondary USB output for connection of a nozzle camera or bed level scanner, etc.
- Part fan supports 2, 3 and 4 wire fans with configurable power supply voltage. The power supply is able to deliver 2A of current, enabling the direct operation of the most powerful 4028 fans.
- I2C IO port for interconnection with I2C protocol sensors (like OrbiScan bed scanner). This port replaces the old BL_TOUCH connection of the previous board version
- Hotend heater connector changed from screw terminal to micro fit connector.
- Tool fan output has been removed, to make space for the 4wire part fan interface.
- The microcontroller has been changed to STM32F072K6T6, with a larger flash memory allowing bootloader flashing and auto update directly from the Mainsail interface.
- Onboard accelerometer changed to ADXL345
- Stepper driver changed to TMC2240 which has lower thermal losses compared to TMC2209
- USB adapter protection circuit supplied from onboard 5.5V power supply, not relying anymore on the 5V supply from the RPI, making the USB communication more stable.

3.2 Power supply concept

The board is supplied via an XT30 (2+2) connector, which includes +24V, GND, USB_DATA+ and USB_DATA- signals.

From the connector, the board is supplied via a reverse power supply protection circuit. This prevents tool-board damage. In the event of an accidental reversed supply by user error, the board will not start up and will behave like it's not powered.

The microcontroller is supplied from a 3.3V LDO supplied from the onboard 5V DC-DC buck converter for the highest power supply efficiency and lowest possible power consumption.

The RGB LED and the external USB port are supplied from a short-to-GND protected power supply.

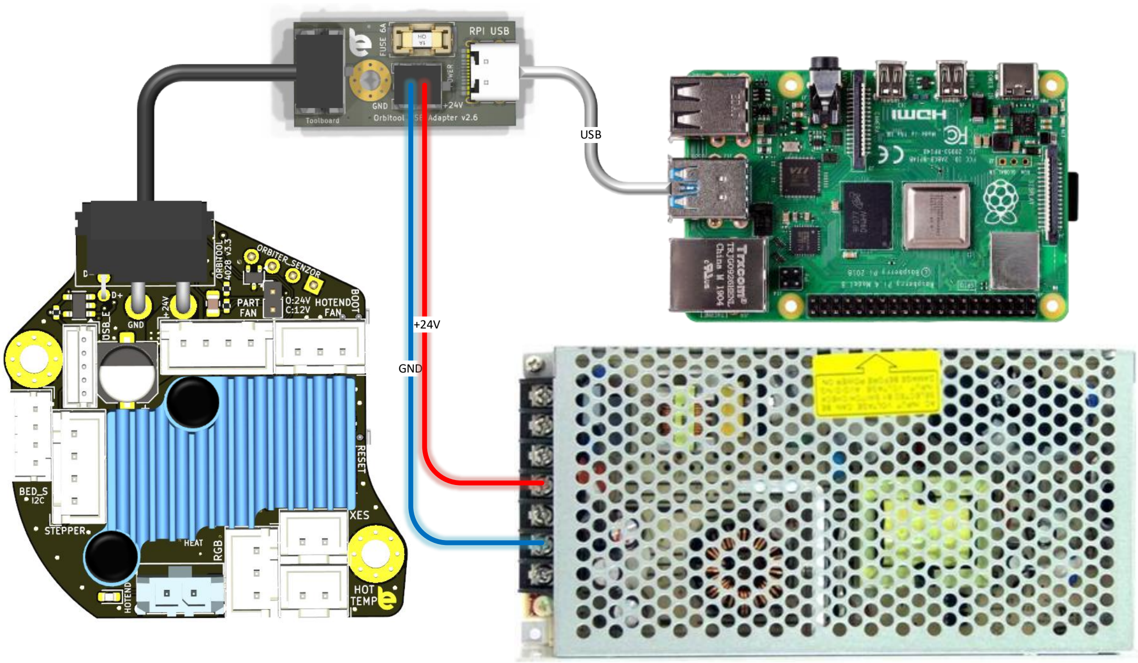
3.3 USB communication interface

The tool-board is equipped with an automotive grade USB interface. The main difference is the higher electrical robustness of the automotive USB, still maintaining the highest possible communication speed. Therefore, if the USB data lines are short-to-GND or up to +28V, it will not cause any destruction and will work properly after the short circuit is removed.

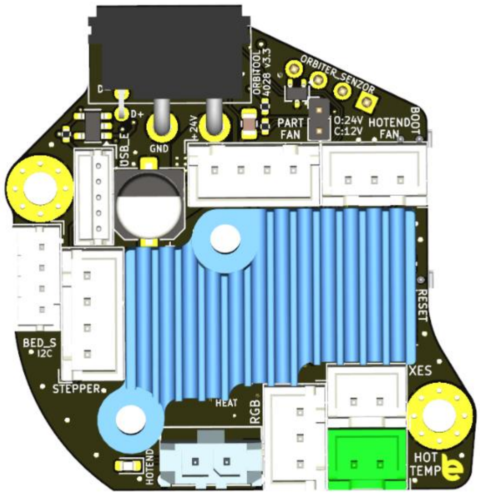
The same circuit offers protection in case of GND connection loss, which will lead to biasing all the communication signals to the supply positive line (in our case, +24V).

The USB interface of the Raspberry Pi is not robust against these electrical hazards that may occur in a 3D printer. Therefore, the tool-board shall be supplied and connected to the Raspberry Pi using the supplied adapter board, which has onboard protection circuits to protect the Raspberry Pi USB port.

The next picture presents the tool-board wiring connection principle.



3.4 Hot-end temperature sensor



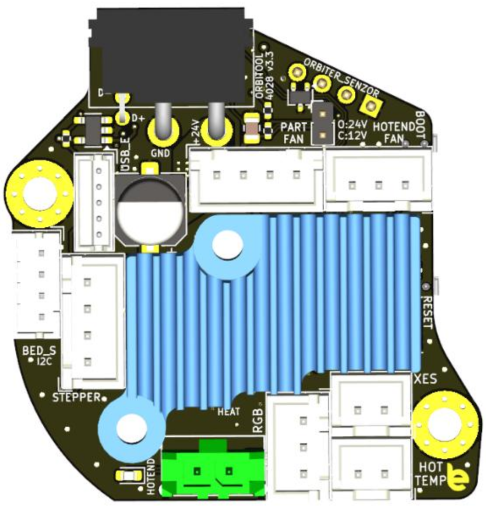
The hot-end input circuit accepts two types of thermistors: 100k NTC type similar to the ATC Semitec 104GT-2 and PT1000.

Pull-up resistor value: 2200Ω.

The sensor input is protected against short circuits to GND and supply voltage of 24V.

The GND connection of the sensor is current limited to avoid thermal runaway of the hotend in the event of a short circuit between the sensor and heater wires.

3.5 Driver circuit for hot-end heater



The hot-end heater driver circuit is implemented with active protection against short circuits.

The circuit measures the heater current and in case the current rises over a predefined threshold i_{OC_HOT}, it switches off the heater MOSFET until the next PWM cycle to prevent damage.

The heater max power shall be limited to 99.5%, meaning max_power: 0.995 to avoid the driver getting stuck in protection mode due to an unlikely event of a fake error detection.

```
max_power: 0.995
```

The heater current is fed to Klipper, which can calculate the actual power consumption of the heater element. This can be used to detect failures of the heater element, like a short circuit or a loss of heating power.

You may say yes; other boards also have protection with onboard fuses. Well, that is simply not enough to protect MOSFETs from being damaged because fuses have a long reaction time, about 1~2s. Their mission is to protect the circuit from catching fire in case of a defect, but it cannot protect the heater driver MOSFET from getting destroyed. This active protection reacts within 10ms, ensuring the driver stage is switched off before it gets damaged due to a short circuit or overload event.

To display the actual power of the heater, include the following section:

```
[adc_temperature HOT_P]
temperature1:120 # value in Watts
#voltage1:1.32
voltage1:1.91
temperature2:240 # value in Watts
voltage2:3.82
#voltage2:2.64

[temperature_sensor Hotend_power]
sensor_pin: orbitool02S:PA0
sensor_type: HOT_P
```

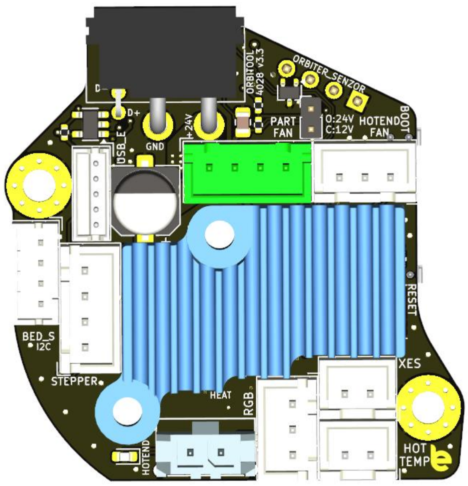
The hot-end power or current can be displayed based on preference. The power is calculated considering a 24V power supply.

To improve the heater power calculation accuracy, the heater PWM is forced to 100Hz (instead of the 10Hz default of Klipper).

```
max_power: 0.995 # limit heater power to 99.5% to enable autorecovery from short detection
pwm_cycle_time: 0.01
```

It is highly recommended to use ferrules to connect the heater wires to the tool-board screw terminal block.

3.6 Part fan driver



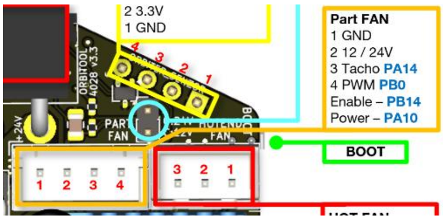
The part fan connector supports the classical two or three wire fans, and four wire fans.

It can even drive the most powerful 4028 fans supplied by 12V.

The fan driver is based on a powerful 2A capable DC-DC converter circuit. The part fan is directly driven by a DC voltage instead of PWM. This reduces the fan noise caused by the PWM chopping.

The power supply level of the part fan can be selected by the small 2mm jumper near the part fan connector as follows:

- Jumper open -> part fan supplied by 24V
- Jumper closed -> part fan supplied by 12V



Connector pins definition:

1- GND
2- 12/24V – positive supply for 2 and 3 wire fans
3- Tacho – speed signal
4- PWM for 4 wire fans

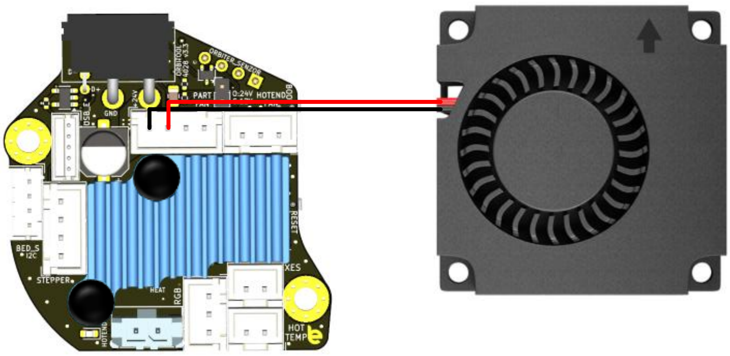
The fan driver has an enable signal connected to PB14.

3.6.1 Two wire fan connection and configuration

A regular two wire fan should be connected as follows:

- Negative terminal of the fan to pin number 1 – GND
- Positive terminal of the fan to pin number 2

Note: Pin number 4 – PWM is only used for 4 wire fans.



Warning: Please make sure the correct fan voltage is selected by the part fan voltage selection jumper. Supplying a 12V fan with 24V might damage the fan.

Since the driver applies a DC voltage instead of a PWM there could be a minimum voltage needed for the fan to start up.

This minimum value shall be defined experimentally (example: off_below: 0.2).

Klipper configuration example:

```
##################################################################
########### 2 and 3 wire part fan configuration ###############
##################################################################

[fan]
pin: !orbitool02S:PA10
enable_pin: orbitool02S:PB14
off_below: 0.2
tachometer_pin: orbitool02S:PA14
tachometer_ppr: 2
tachometer_poll_interval: 0.0005
shutdown_speed: 0.0
cycle_time: 0.0001 # 10KHz control signal frequency
kick_start_time: 0.2
hardware_pwm: False
```

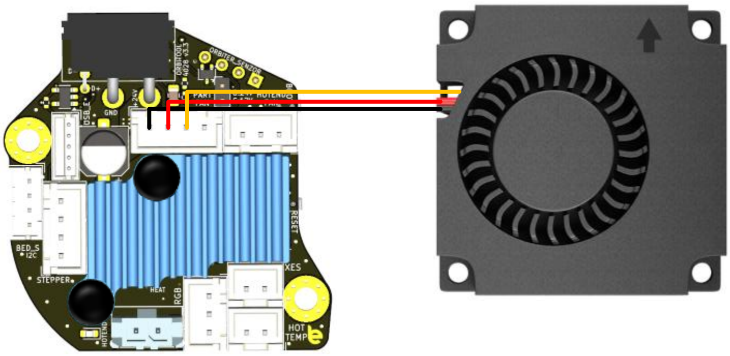
Note: The PWM frequency (cycle_time = 0.001) has nothing to do with the fan control. It is related to the fan control circuit, which is designed for a command signal of 10Khz.

3.6.2 Three wire fan connection and configuration

A three-wire fan should be connected as follows:

- Negative terminal of the fan to pin number 1 – GND
- Positive terminal of the fan to pin number 2 – 12/24V
- RPM terminal of the fan to pin number 3 - Tacho

Note: Pin number 4 – PWM is only used for 4 wire fans



Warning: Please make sure the correct fan voltage is selected by the part fan voltage selection jumper. Supplying a 12V fan with 24V might damage the fan.

Since the driver applies a DC voltage instead of a PWM there could be a minimum voltage needed for the fan to startup.

This minimum value shall be defined experimentally (example: off_below: 0.2).

Klipper configuration example:

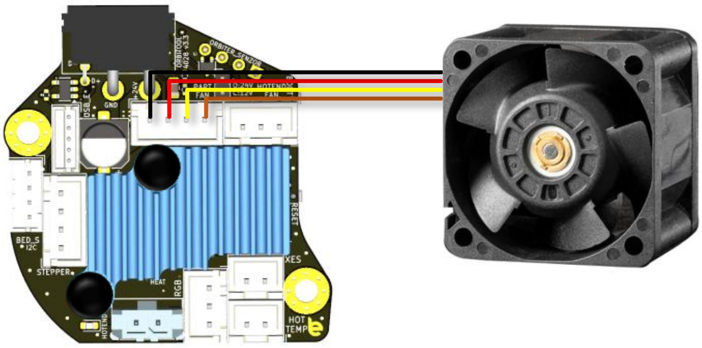
```
#################################################################
########### 2 and 3 wire part fan configuration ###############
#################################################################

[fan]
pin: !orbitool02S:PA10
enable_pin: orbitool02S:PB14
off_below: 0.2
#tachometer_pin: orbitool02S:PA14
#tachometer_ppr: 2
#tachometer_poll_interval: 0.0005
shutdown_speed: 0.0
cycle_time: 0.0001 # 10KHz control signal frequency
kick_start_time: 0.2
hardware_pwm: False
```

Note: The PWM frequency (cycle_time = 0.001) has nothing to do with the fan control. It is related to the fan control circuit, which is designed for a command signal of 10Khz.

3.6.3 Four wire fan connection and configuration

A four-wire fan should be connected as follows:

- Negative terminal of the fan to pin number 1 – GND
- Positive terminal of the fan to pin number 2 – 12/24V
- RPM terminal of the fan pin to number 3 – Tacho
- Speed control terminal of the fan to pin number 4 - PWM



Note: Wire colors might be different, please consult the fan manufacturer documentation for details.

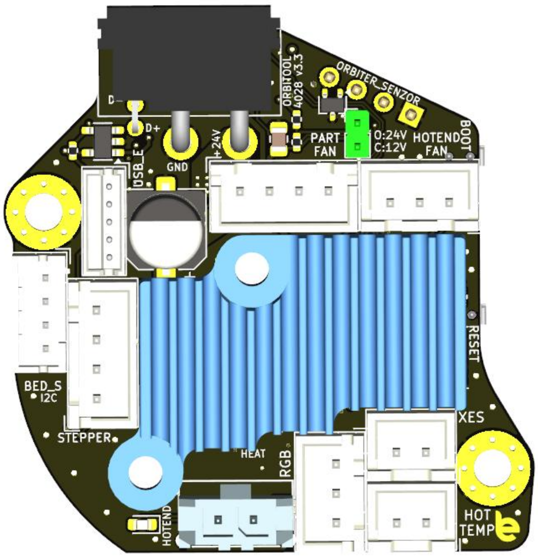
Warning: Please make sure the correct fan voltage is selected by the part fan voltage selection jumper. Supplying a 12V fan with 24V might permanently damage the fan.

Klipper configuration example:

```
################################################################
########### 4 wire part fan configuration ####################
################################################################

[fan]
pin: orbitool02S:PB0
enable_pin: orbitool02S:PB14
tachometer_pin: orbitool02S:PA14
tachometer_ppr: 2
tachometer_poll_interval: 0.0005
shutdown_speed: 0.0
cycle_time: 0.00004 # 25KHz control PWM frequency
hardware_pwm: False
```

Check out the fan datasheet for the correct speed control PWM frequency, most fans are working with 25Khz but it is not a universal standard.

3.6.4 Part fan supply voltage configuration



It is very important that the voltage selection jumper is fitted according to the used part fan.

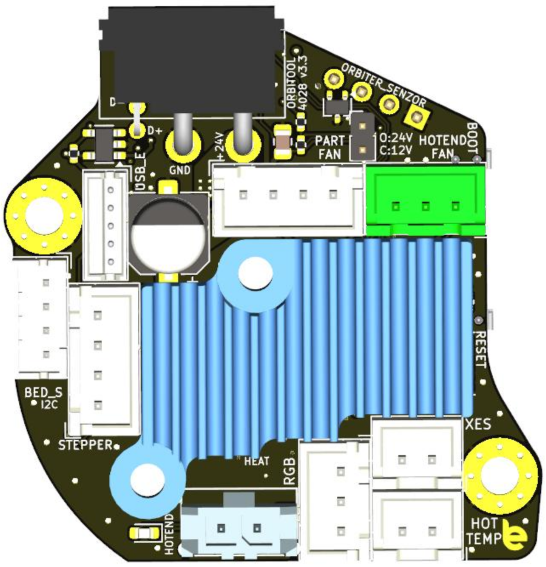
A small 2mm jumper placed near the part fan connector selects the part fan supply voltage as follows:

- Open - 24V fan type
- Closed – 12V fan type

Be aware that supplying a 12V fan with 24V might permanently damage the fan!

3.7 Hot-end fan output

The hot-end fan circuit is implemented using a microcontroller-controlled DC-DC converter.



PWM control of three wire fans will destroy the speed signal. PWM (Pulse Width Modulation) control means switching on and off the fan power supply several times every second. This means there are moments when the fan is actually not supplied with power. In those moments, the speed sensing electronic circuit is not supplied, therefore it cannot generate a reliable speed signal. Practically, the PWM signal will be superimposed over the speed signal.

Another advantage of this control circuit is the possibility of limiting the maximum voltage according to the fan type the user has.

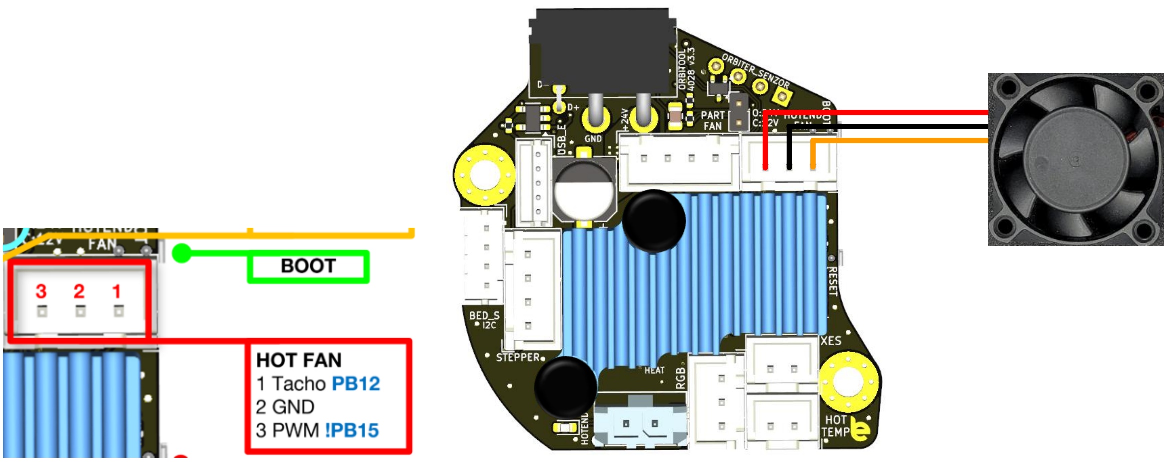
This output supports 12V or 24V fan types configured by the max power value in the fan configuration section:

- **12V** fan – max_power: 0.5
- **24V** fan – max_power: 1.0

Since the fan driver is based on a DC-DC converter, the PWM frequency required for correct operation is defined by the converter design, not by fan properties. Fan PWM speed control frequency shall be set to 10Khz, cycle time = 0.0001 – do not change this to a lower frequency.

Based on a specific printer design, the maximum power can be limited to reduce fan noise. Increase this to a higher level in case you experience hot end clogs.

The connection of a **three-wire** fan should respect the following connection scheme:



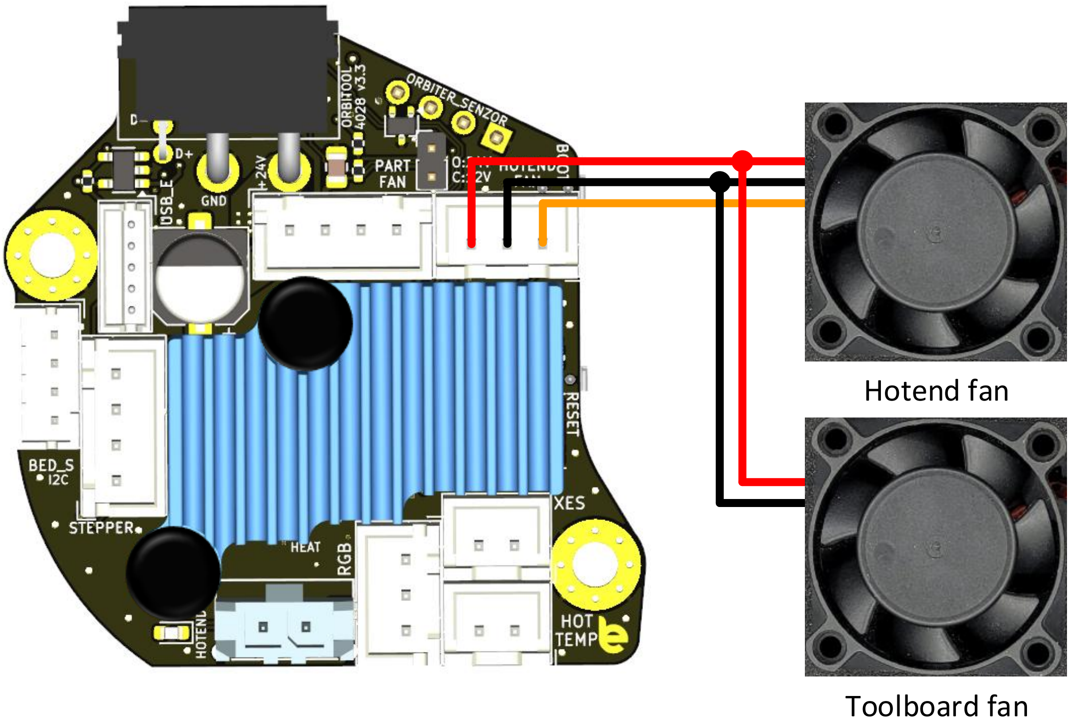
When a **two-wire** fan is connected, the orange wire shall be omitted and the pin left unconnected and the tachometer related configurations shall be commented out.

Klipper configuration example:

```
[heater_fan hotend_Fan]
pin: !orbitoolO2S:PB15
tachometer_pin: orbitoolO2S:PB12
tachometer_ppr: 2
tachometer_poll_interval: 0.0005
heater: extruder
cycle_time: 0.0001 #10KHz PWM frecvency
heater_temp: 75
fan_speed: 1.0 # can be lowered to reduce fan noise
hardware_pwm: false
shutdown_speed: 0.0
max_power: 1.0 # can be lowered to reduce fan noise
```

Note: The PWM frequency (cycle_time = 0.0001) has nothing to do with the fan control. It is related to the fan control circuit, which is designed for a command signal of 10Khz.

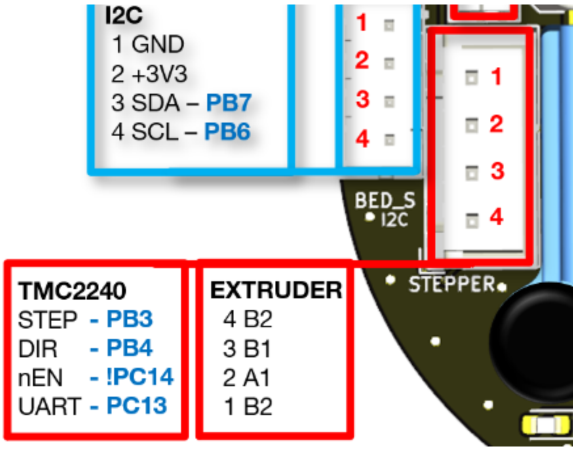
When a toolboard cooling fan is used, it is recommended to connect it together with the hotend fan as follows:



3.8 Extruder stepper driver

The onboard stepper driver type is TMC2240, this has reduced power loss compared to the TMC2209, therefore the board has better thermal behavior and can be used with lower current limitation in a hot chamber.

Pinout of the extruder stepper connector:



The stepper driver configuration is part of the extruder configuration, in this section, only the stepper driver related parts are defined.

Klipper configuration example:

```
[extruder]
step_pin = orbitool02S:PB3
dir_pin = !orbitool02S:PB4
enable_pin = !orbitool02S:PC14
```

```
[tmc2240 extruder]
uart_pin: orbitool02S:PC13
interpolate: false
run_current: 0.85 # Orbiter v2.5
rref: 12000
stealthchop_threshold: 99999
```

3.9 ADXL345 Accelerometer

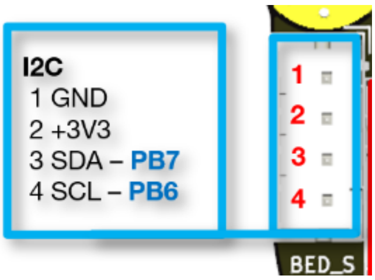
The toolboard is equipped with ADXL345 accelerometer connected to SPI1 of the microcontroller.

Klipper configuration example:

```
##############################################################
################## Accelerometer Sensor ######################
##############################################################

[adxl345 Tool0]
cs_pin: orbitool02S:PA4
spi_bus: spi1
axes_map: y, z, x # Assumes back-facing vertical toolboard mounting
spi_speed: 5000000
```

3.10 I2C interface

The onboard I2C port can be used to connect any type of external sensors with I2C communication protocol, like load cells, temperature sensors, eddy current based bed scanners (like Orbiscan or BTT Eddy Coil), etc.



The I2C bus name is: i2c2_PB10_PB11

The next picture presents a configuration example of an eddy current bed scanner:

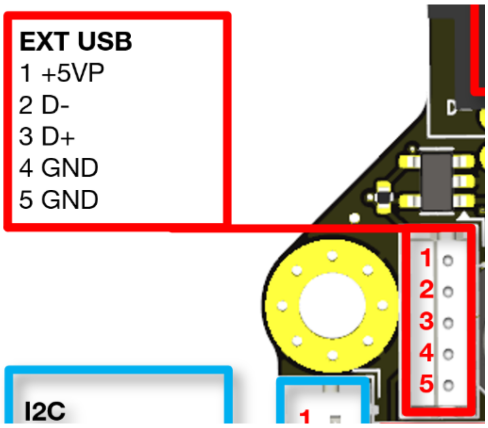
```
##############################################################
###################### Bed scanner ###########################
##############################################################

[probe_eddy_current orbiscan]
sensor_type: ldc1612
i2c_mcu: orbitool02S
i2c_bus: i2c2_PB10_PB11
i2c_speed:100000
```

3.11 Additional USB port

The additional onboard USB port is designed to support the USB1.0 protocol (12Mbit/s).

This USB port can be used to connect bed surface scanners like Beacon, Cartographer, BTT eddy with USB or a nozzle camera.

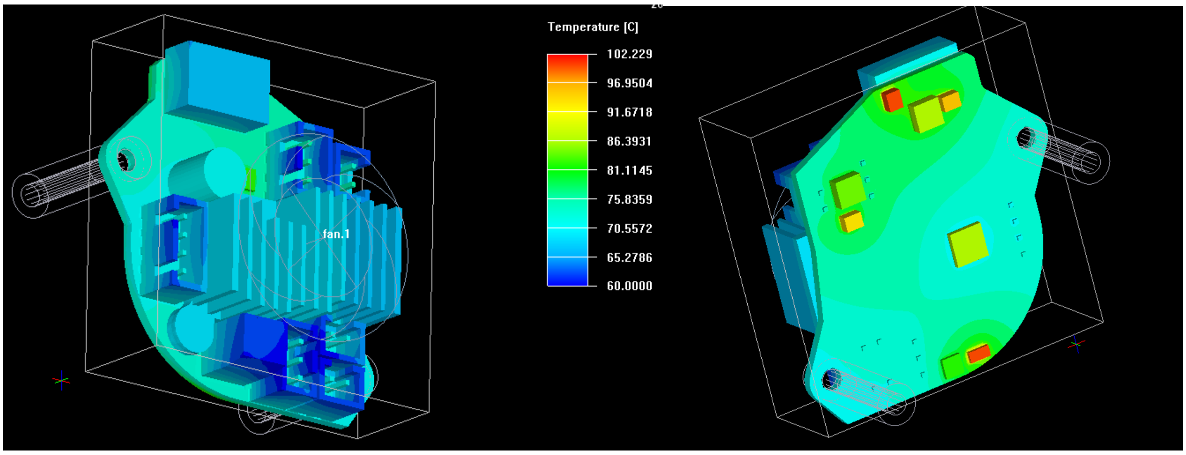
Note pin 4 and 5 are GND connections. The aim is that Pin 4 is used to connect the black USB wire and pin 5 to connect the USB cable shield. However, it is not mandatory to use both GND pins. The black wire and the USB cable shield can be tied together and connected to either of the GND pins.



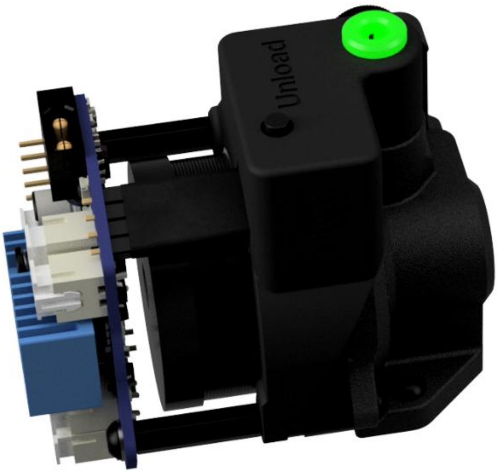
3.12 Thermal considerations

A closed printer chamber requires tool-board operation in high ambient temperature conditions. In addition, this tool-board is mounted on the backside of the stepper motor, which transfers a considerable amount of heat. Therefore, the circuits of this tool-board are designed to have low temperature drift and efficient thermal management to make sure none of the components heat up over their maximum allowed operating temperature.

I performed lots of thermal simulations using finite element simulations and real-life measurements using a Flir thermal camera.



The simulation results show that in a 60°C chamber temperature environment, the tool-board components do not heat over their maximum junction temperature rating.



To reduce heat transfer from the stepper towards the tool-board a minimum distance of 4-5 mm shall be maintained. Orbiter v2 and v2.5 equipped with the LDO-36STH20-1004AHG stepper and the Orbiter smart sensor, standoffs with 18 mm length is recommended. This way, the distance between the toolboard PCB and the stepper back side is about 4-5 mm.

Without the Orbiter smart sensor, the standoff length shall be minimum 20 mm.

The toolboard is designed for passive cooling, but even with the most optimized thermal design, stepper current derating over temperature still applies. This is mainly because there is simply not enough space for a bigger heatsink. Even if the heatsink size were doubled, it would still not be enough.

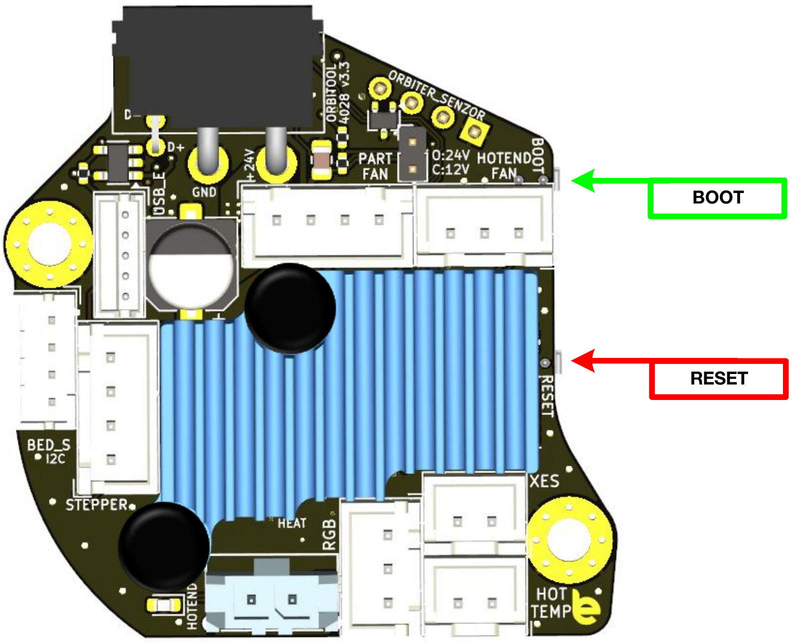
Therefore, we have two possible solutions:

1. Reduce stepper current for high ambient temperature operation. Example down to 0.55A for chamber temperature above 60°C.
2. Use active cooling to improve the thermal behavior. Adding a small 20x20mm or 25x25mm fan on the back of the toolboard that blows air toward the heatsink will improve a lot.

4 Microcontroller programming

1. Connect the board to the host Raspberry Pi via a USB adapter cable.
2. This step depends on whether your board has firmware on it or not:
   a) If your board is pre-flashed with Klipper, first must enter DFU mode as follows:
      - press the BOOT and RESET buttons,
      - release the RESET while keeping the BOOT button pressed,
      - release the BOOT button after about three seconds.

   b) If your board is a new, un-flashed board:
   - The MCU is already in DFU mode; confirm this at step 4



3. Connect to your host Raspberry Pi via SSH
4. Run lsusb from the command prompt
   - Make sure you see an STM32 in DFU mode listed

```
Bus 001 Device 007: ID 0483:df11 STMicroelectronics STM Device in DFU Mode
```

5. Run dfu-util --list from the command prompt
   - note the text inside the [xxxx:yyyy] (default is -> 0483:df11)
6. Run cd ~/klipper from the command line to enter the Klipper directory
7. Run make menuconfig settings should be:
   - Cristal oscillator – **12Mhz**

```
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F072)  --->
    Bootloader offset (No bootloader)  --->
    Clock Reference (12 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    USB ids  --->
[ ] Optimize stepper code for 'step on both edges'
()  GPIO pins to set at micro-controller startup
```

- Set custom USB ID to **Orbitool_O2S**

```
                    Klipper Firmware Configuration
(0x1d50) USB vendor ID
(0x614e) USB device ID
[ ] USB serial number from CHIPID
(Orbitool_O2S) USB serial number
```

- Hit Q to Exit and Save

8. Run make clean to clean up the make environment
9. Run make flash FLASH_DEVICE=0483:df11
   If you encounter some error at the end but still have the message of File downloaded successfully, you are good to proceed to the next step.

```
Downloading element to address = 0x08000000, size = 37072
Erase           [=========================] 100%        37072 bytes
Erase     done.
Download        [=========================] 100%        37072 bytes
Download done.
File downloaded successfully
Submitting leave request...
Transitioning to dfuMANIFEST state
dfu-util: can't detach
Resetting USB to switch back to Run-Time mode

Failed to flash to 0483:df11: Error running dfu-util

If the device is already in bootloader mode it can be flashed with the
following command:
  make flash FLASH_DEVICE=0483:df11
  OR
  make flash FLASH_DEVICE=1209:beba

If attempting to flash via 3.3V serial, then use:
  make serialflash FLASH_DEVICE=0483:df11

make: *** [src/stm32/Makefile:107: flash] Error 255
```

10. Press the RESET button to restart the MCU and enter normal operating mode
11. Run ls /dev/serial/by-id/* should return a device with /dev/serial/by-id/usb-Klipper_stm32f072xb_Orbitool_O2S-if00
    Copy this virtual serial port name to the **Orbitool_O2S.cfg** file in the MCU section

```
[mcu orbitool_O2S]
serial:/dev/serial/by-id/usb-Klipper_stm32f072xb_Orbitool_O2S-if00
restart_method: command
```

Manual definition of USB ID, flashing the MCU with Klipper will always result in the same serial ID.

Your tool-board should now be usable with Klipper. Use the example config file to get started.
The best option is to copy the config file into the same directory as printer.cfg.
Add [include Orbitool_O2S.cfg] to the beginning of your printer.cfg to include the file.
Comment out the unused I/O features of the OrbitoolO2S config section.

5 Assembly instructions

Based on the user setup, we can have several assembly combinations, as described below. As a recommendation, some kind of bracket with strain release for the USB cable is a must. Fixing the cable is necessary to prevent constant bending forces over the PCB during printhead movements. Even if the forces are not very high, it still stresses the PCB, and it could develop component or solder cracks. A well-designed strain release will avoid this issue.

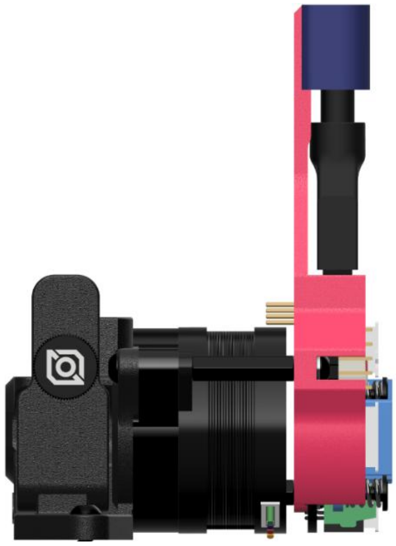
5.1 Orbiter v2 + Orbitool + Bracket

Mount the two M3 L=18mm standoffs to the Orbiter v2 stepper's backside.

Print the braket and the wire lock (downloadable from GitHub).

Insert the toolboard into the bracket, and using the two M3 L=8 mm and two washers fix it to the two standoffs.

Insert the USB cable and fix it into place using the printed wire lock.



5.2 Orbiter v2 + Orbitool

Change the two screws holding the Orbiter v2 together with the longer ones provided in the toolboard package (M3 L=25mm).

Add two M3 nuts over the screws on backside of the stepper, after which mount the two M3 L=18mm standoffs as shown in the picture.

Attach the toolboard and secure it in place with the M3 L=8mm screw sets and washers provided in the toolboard kit.

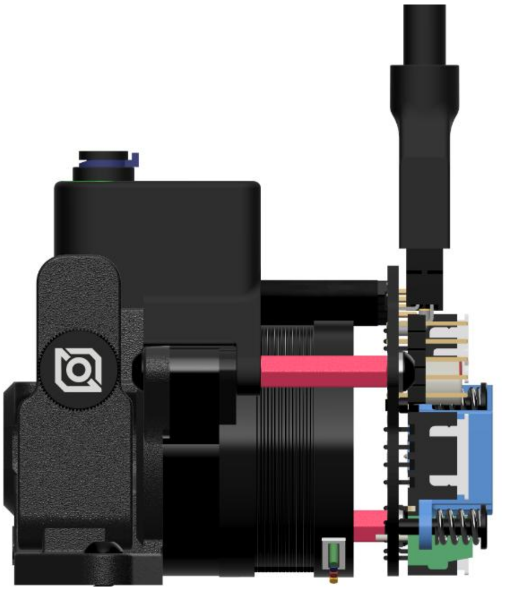
5.3 Orbiter v2 + Orbitool + Orbiter sensor

Mount the Orbiter sensor over the top of the Orbiter v2, using the two longer M3 screw sets provided in the sensor kit.

Insert the small interconnection part between the sensor and the toolboard into the sensor connector.

Screw on the two M3 standoffs with 18mm length over the sensor backside.

Attach the toolboard and secure it in place with the M3 L=8mm screw sets and washers provided in the toolboard kit.



5.4 Orbiter v2 + Orbitool + sensor + bracket

Mount the Orbiter sensor over the top of the Orbiter v2, using the two longer M3 screw sets provided in the sensor kit.

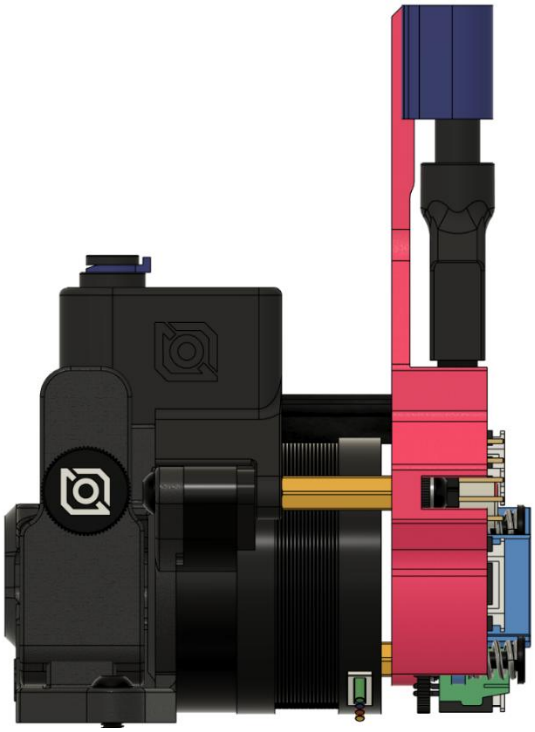
Print the bracket and the wire lock (downloadable from GitHub).

Insert the small interconnection part between the sensor and the toolboard into the sensor connector.

Screw on the two M3 standoffs with 14mm length over the sensor backside.

Insert the toolboard into the bracket, and using the two M3 L=8 mm and two washers, fix it to the two standoffs.

Insert the USB cable and fix it into place using the printed wire lock.



5.5USB adapter board bracket

To fix the small USB adapter board to the printer frame, you may use my simple mount, as shown in the next picture. All you need is to print the mount, an M3 heat insert (L = 4~5 mm), and an M3 screw (L = 6~8 mm). The PCB mount can be fixed to the printer frame using two M3 screws and nuts, or attach it simply using strong foam tape.

The new USB adapter board version 3.0 is an improved version compared to the last version, v2.72. It has an onboard 5V regulator that supplies the onboard USB protection circuit. This means the board is not relying anymore to receive a proper 5V supply voltage via the USB connector, resulting in more reliable USB communication, less probability of toolboard disconnection due to noise or instability of the 5V power supplied via the Raspberry PI USB connector.

5.6 Toolboard pack list

Orbitool O2S toolboard pack list				
Item	**Description**	**Nr pieces / length**	**Part number**	**LCSC**
1	Orbitool O2S	1	v3.3	
2	USB adapter board	1	v3.0	
3	24V USB cable	150cm	v2	
4	24V power supply cable	50cm		
5	Short USB A to USB C cable	30cm		
6	Orbiter Sensor connecting bridge	1		C2718488
7	XH-2 pin connector housing	2	XHP-2	C144401
8	XH-3 pin connector housing	2	XHP-3	C144402
9	XH-4 pin connector housing	2	XHP-4	C144403
10	XH crimp pins	25	SXH-001T-P0.6N	C385122
11	PH-4 connector housing	1	PHR-4	C111514
12	PH crimp pins	10	SPH-002T-P0.5S	C111515
13	ZH-5 connector housing	1	ZHR-5	C160378
14	ZH crimp pins	10	SZH-002T-P0.5	C246761
15	Molex Micro-Fit 2 pin connector housing	1	436450200	C114089
16	Molex Micro-Fit crimp pins	5	430300001	C259786
17	2mm Jumper	2		C5664
18	M3 threaded aluminum spacer **L=14mm**	2	M3 L14	
19	M3 threaded aluminum spacer **L=18mm**	2	M3 L18	
20	M3 screw L = 8mm	3	M3x8	
21	Me screw L = 25	2	M3x25	
22	M3 washers	2		
23	M3 brass insert nut L=6mm	2		
23	M3 nut	2		

6 Tips and tricks

Correct operation of the toolboards and basically any 3D printer electronics depends on the environment they are used in. Even if these toolboards have lots of protection circuits, it does not mean that improper installation or usage cannot cause damage or malfunction.

Below is a list of recommendations when wiring DIY 3D printers:

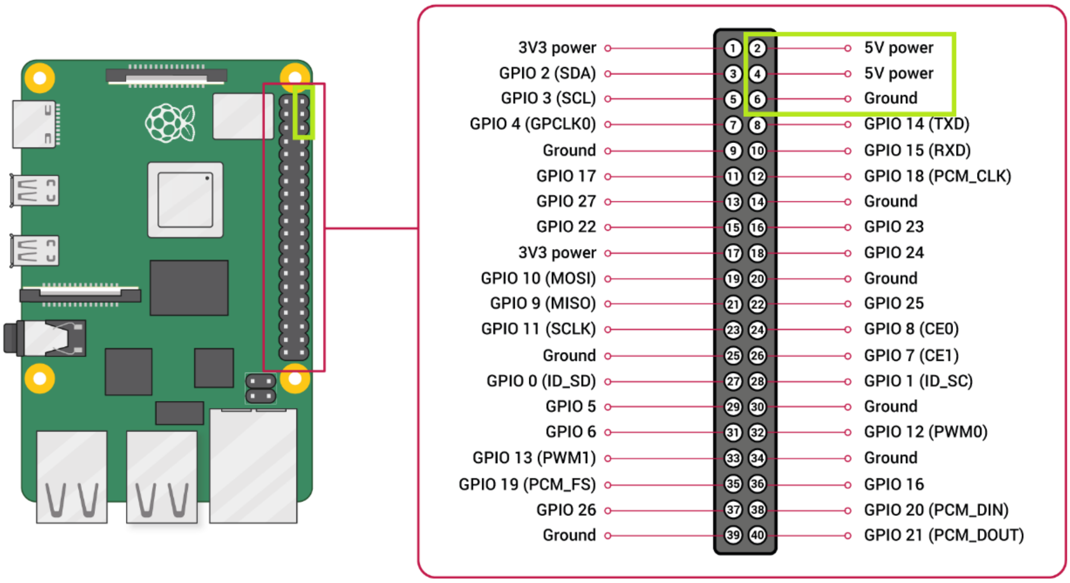
1. The printer frame must be connected to mains earth and GND.

2. Aluminum bed shall be connected to frame and GND. A severe crash of the toolhead into the bed or print piece causing the toolhead to break may cause short circuits between the bed and any input or output of the toolboard (like heater, fan outputs end-stop, temperature sensor inputs, etc.). If the bed is not connected to GND, then it can have any voltage potential level, even in the range where the toolboard protection circuit is not effective anymore, leading to loss of functions or total damage.

3. Toolhead made with metal parts shall be connected to GND with extra wire using proper electrical connection washer (a washer with ribs or teeth, and hole for wire soldering). Not grounded toolheads will build up static charge, eventually, this will arc over the toolhead, which may cause electrical damage to the toolboard (less likely) or disturb communication between the toolboard and RPI causing disconnection errors (very likely scenario).
4. When connecting wires to printer frame or any aluminum parts, special washers with toots or ribs shall be used! Aluminum oxide is very good electrical insulator, therefore, using only a flat washer connector will not ensure proper electrical contact.



5. USB cables are not the best choice to supply the RPI due to high voltage drop over them. The best is to supply the RPI with 5V via the GPIO pin header pins with a proper cable. This ensures the supply voltage of the RPI is stable during sudden current load increases. Remember that when the adapter board is connected to the RPI, the supply voltage of the USB protection circuit is taken from the RPI, any disturbance in that is a potential communication disturber.


3V3 power
GPIO 2 (SDA)
GPIO 3 (SCL)
GPIO 4 (GPCLK0)
Ground
GPIO 17
GPIO 27
GPIO 22
3V3 power
GPIO 10 (MOSI)
GPIO 9 (MISO)
GPIO 11 (SCLK)
Ground
GPIO 0 (ID_SD)
GPIO 5
GPIO 6
GPIO 13 (PWM1)
GPIO 19 (PCM_FS)
GPIO 26
Ground
5V power
5V power
Ground
GPIO 14 (TXD)
GPIO 15 (RXD)
GPIO 18 (PCM_CLK)
Ground
GPIO 23
GPIO 24
Ground
GPIO 25
GPIO 8 (CE0)
GPIO 7 (CE1)
GPIO 1 (ID_SC)
Ground
GPIO 12 (PWM0)
Ground
GPIO 16
GPIO 20 (PCM_DIN)
GPIO 21 (PCM_DOUT)

7 Electrical characteristics

7.1 Absolute maximum ratings

Important Note: The Orbitool O2S board can withstand these limits without electrical damage; however, long-term exposure to these limits is not recommended. The absolute maximum rating is not the same as the functional range.

The maximum ratings may not be exceeded under any circumstances!

Table 1. Orbitool O2S absolute maximum ratings

Nr.	Parameter	Min	Max	Unit
A1	Ambient temperature range	-20	85	°C
A2	Supply voltage	-30	30	V
A3	Extruder stepper current		1	A
A4	Hot-end heater current		6.8	A
A5	Part fan output current		3	A
A6	Hot-end Fan		1	A
A7	RGB LED 5V supply current		0.5	A
A8	RGB data line output	-1	7	V
A9	USB Data lines	-1	28*	V
A10	Hot-end temperature sensor input	-1	28*	V
A11	End stop sensor input	-1	28*	V
A12	I2C interface	-1	28*	V
A13	Orbiter v2 sensor interface inputs	-1	3.3	V
A14	ESD-Protection level for handling (Human Body Model, HMB)	-8	8	kV

*Should not exceed the supply voltage

7.2 Full functional operational limits

Table 2. Orbitool O2S electrical characteristics

Nr.	Parameter	Symbol	Min	Typ	Max	Unit
	General					
P1	Operating temperature range	t_o	-20		65	°C
P2	Max ambient temperature with active cooling	t_{o_cooled}			80	°C
P3	Power supply voltage	v_{PWR}	22		28	V
	Extruder stepper					
P5	Extruder stepper current (max 60°C chamber temp)	i_{MOT_60}			0.85*	A
P6	Extruder stepper current (max 80°C chamber temp)	i_{MOT_80}	0.5		0.6	A
	Hot-end heater					
P7	Hot-end heater nominal current	i_{HEATER}			5	A
P8	Hot-end heater overcurrent switch OFF threshold	i_{OV_HOT}	5.9	6.8	7.2	A
P9	Hot-end overcurrent detection time	t_{OV_HOT}			10	ms
P10	Hot-end power measurement accuracy	a_{P_HOT}	-6		6	%
	Part Fan					
P11	Part fan output current	i_{PART_FAN}			2	A
P12	Part fan overcurrent switch OFF threshold	i_{OC_PART}	4.2	5	5.8	A
P13	Part fan PWM frequency	f_{PART_FAN}	10		20	KHz
	Hot-end Cooling fan					
P15	Hot-end fan output current	i_{HOT_FAN}			0.5	A
P16	Hot-end fan current limitation	$i_{CL\_HOT\_FAN}$	2	3	3.9	A
P17	Hot-end fan PWM frequency	f_{HOT_FAN}	10		20	KHz
	External 5V supply					
P18	RGB LED + External USB current	i_{5V_Prot}			0.5	A
	Internal supplies					
P19	Internal 5V supply protection current limitation	i_{SC_5V}	2	3	3.9	A
P20	Internal 3.3V supply protection current limitation	i_{SC_3V3}	220	350	550	mA
P21	Weight			18		g

*Parameter derating with ambient temperature