Rototool v1.4

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!

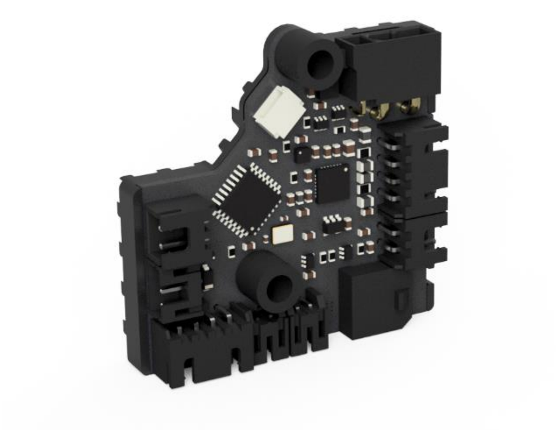
We are hobbits, professionals and let's face it, we do mistakes, and when the magic smoke shows up everybody is disappointed, even if we know it's a user fault. In this board, I promise you the *magic smoke* is kept inside extremely tight!

Smokeless Features

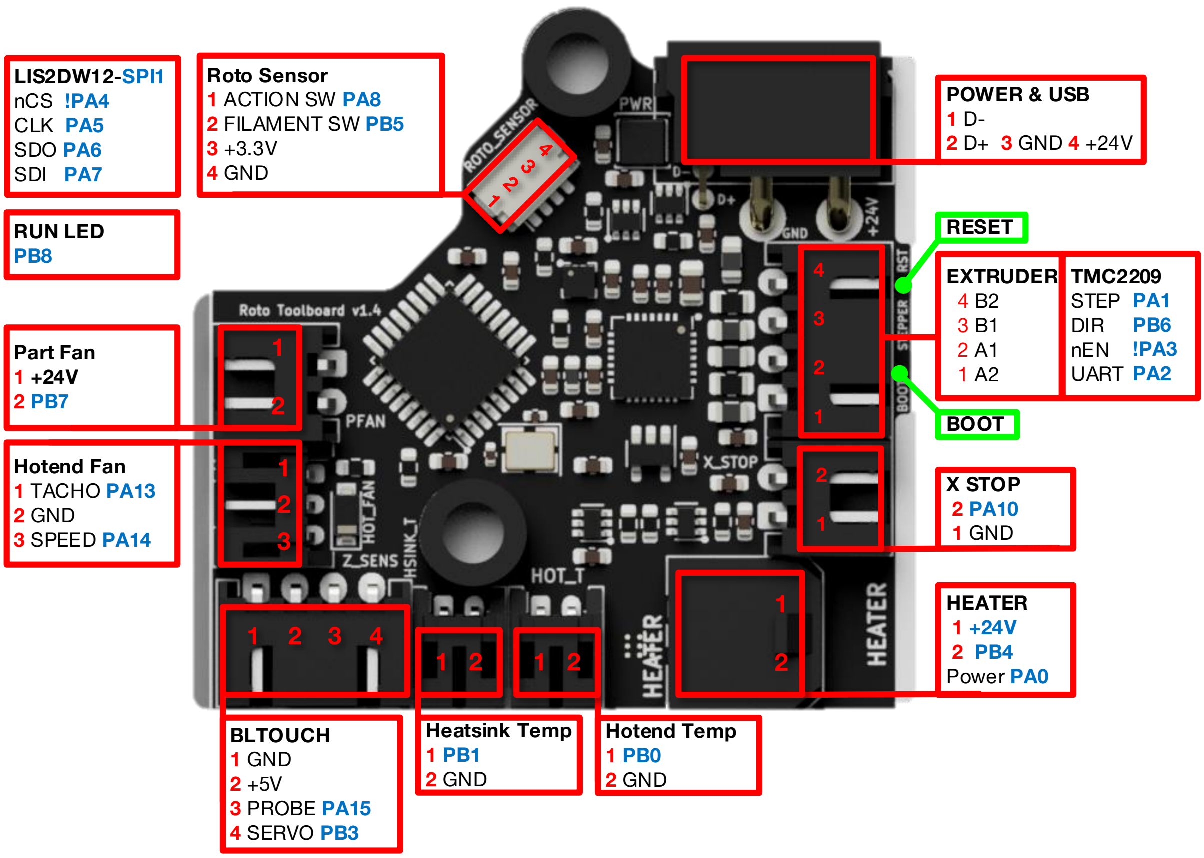
- Optimized for E3D Roto extruder shape and features
- STM32F042 microcontroller running on 48MHz
- Automotive USB communication with Raspberry PI
- Onboard LIS2DW12 accelerometer
- TMC2209 extruder stepper driver
- Direct connection to Roto
- 1 x PWM 24V controlled part fan output
- 1 x PWM 5V controlled extruder fan output with RPM speed input signal
- Hot-end temperature sensor input compatible with standard NTC or PT1000 temperature sensor types
- I/O for bed level sensing
- X-Stop sensor input
- Tool-board and extruder temperature sensor
- Advanced thermal management system

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 protected against short circuits to GND and +24V
- EMI interference and ESD protection an all inputs and outputs
- Heater thermal runaway protection in case of short between heater and sense thermistor wires



1 Pinout definition



2 Features description

2.1Introduction

For me 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 let the *magic smoke* out. Even if is due to user fault the disappointment will still be present.

This design is based on automotive technology where protection to all kind of hazards is state of the art long time ago.

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

2.2 Power supply concept

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

From the connector the board is supplied via a reverse power supply protection circuit. This prevents the tool-board damage in case of accidental reversed supply by user error, the board will not start up it 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.

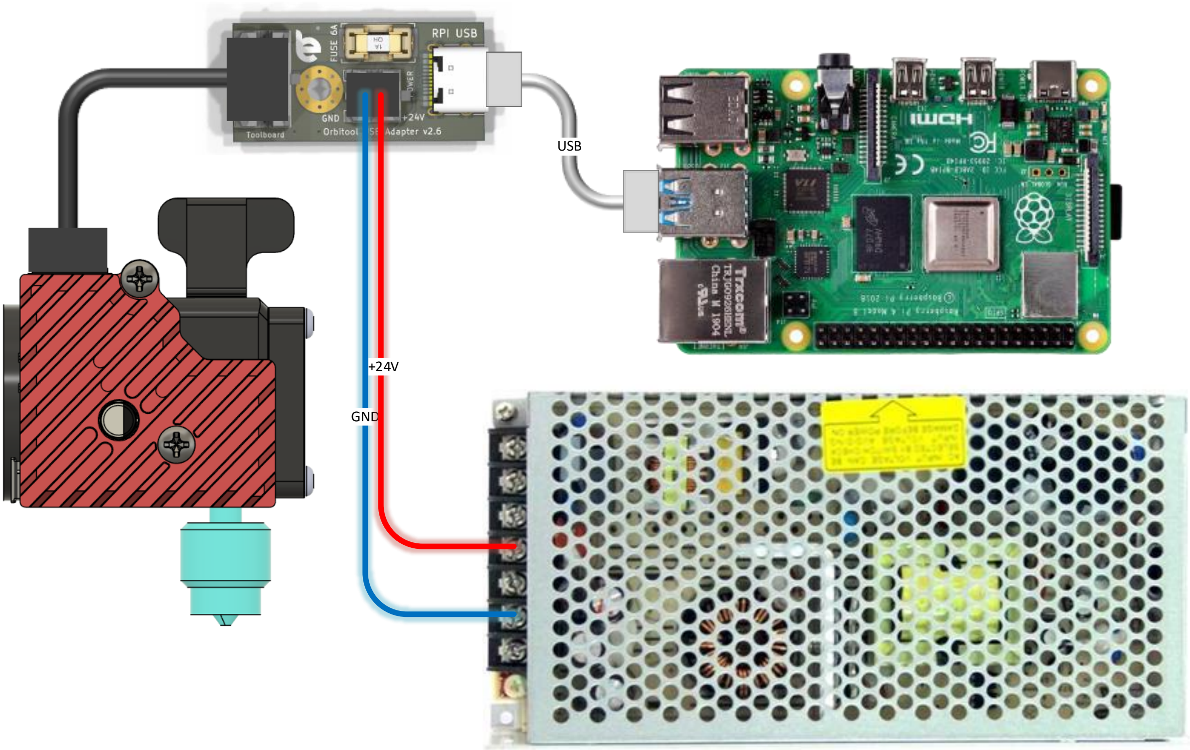
2.3USB communication interface

The tool-board is equipped with automotive grade USB interface. The main difference being the higher electrical robustness of the automotive USB, still keeping high communication speed possible. Therefore, in case of short to GND or up to +28V of the USB data lines 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 which 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 the onboard protection circuits to protect the Raspberry PI USB port.

Next picture presents the tool-board wiring connection principle.



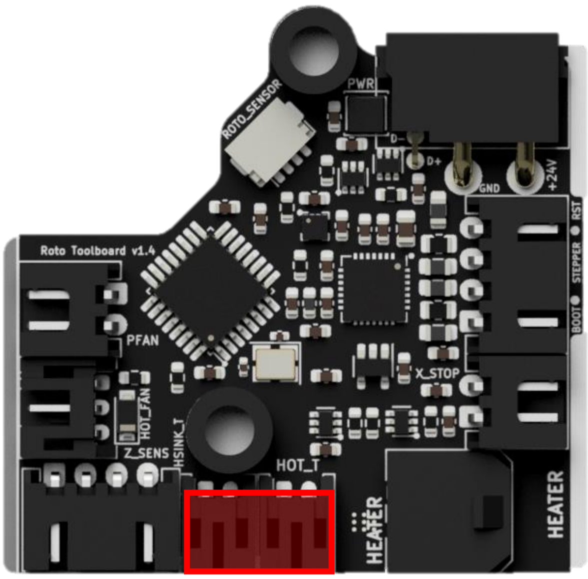
2.4 Roto Hot-end and heatsink temperature sensor

Both inputs accept 100K NTC type and PT1000 temperature sensor types.

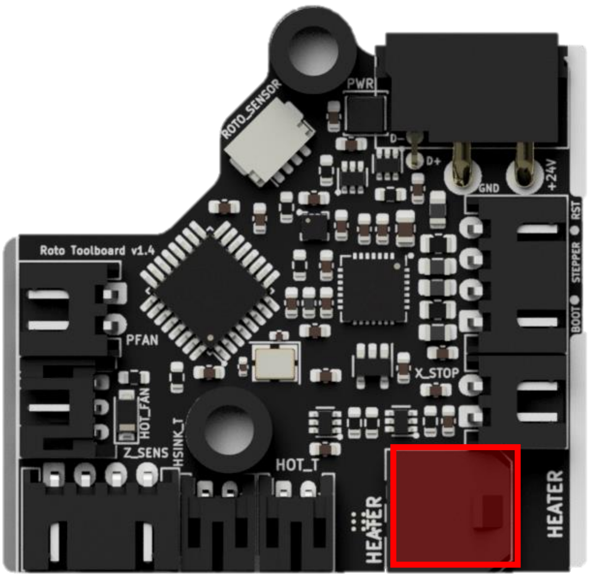
Pull up resistor value 2200.

The senor input is protected against short circuit to supply voltage of 24V.

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



2.5 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 driver stuck in protection mode due to an unlikely event a fake error detection.

```
max_power: 0.995
```

The heater current is feedback 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 short circuit or loss of heating power.

You may say yes, other boards have also protection with onboard fuses. Well, that is simply not enough to protect MOSFETs from being damaged because fuses have long reaction time, about 1-2s range. 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]
#temperature1:5 # value in Ampere
temperature1:119 # value in Watts
#voltage1:1.32
voltage1:1.91
#temperature2:10 # value in Ampere
temperature2:239 # value in Watts
voltage2:3.82
#voltage2:2.64

[temperature_sensor Hotend_power]
sensor_pin: orbitool02:PA2
sensor_type: HOT_P
```

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

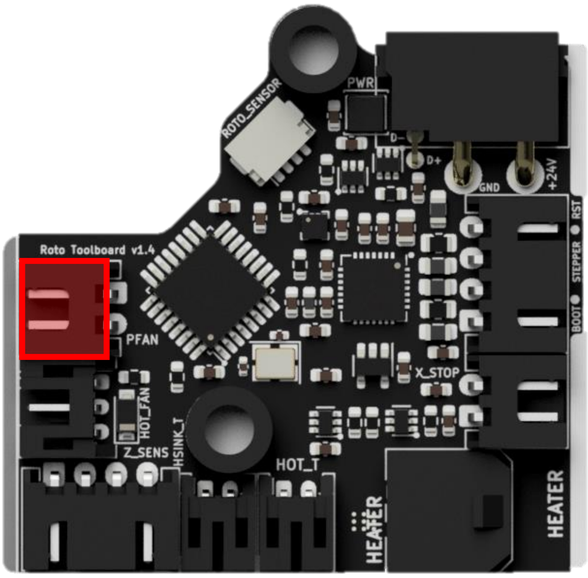
2.6 Part fan driver output

The part fan driver has a similar protection circuit like the hot-end driver, in case of overcurrent the fan driver is switched OFF until the next PWM cycle.

The PWM duty cycle of the fan driver shall be limited to 99.5% meaning max_power: 0.995 to avoid fan driver stuck in OFF state due to an unlikely event a fake error detection.

Configuration of the Fan driver:

```
[fan]
pin: orbitool02:PA10
max_power: 0.995
shutdown_speed: 0.0
cycle_time: 0.02
kick_start_time: 0.2
hardware_pwm: False
```

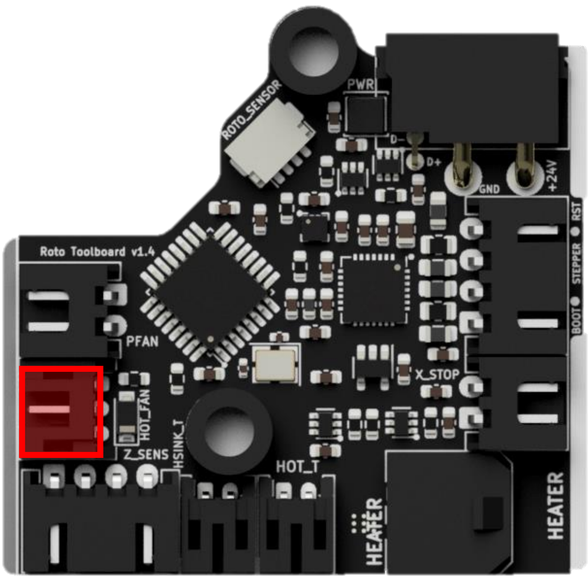


2.7Hot-end fan output

The Roto is equipped with a 5V supplied cooling fan.

The fan output current is limited to 500mA.

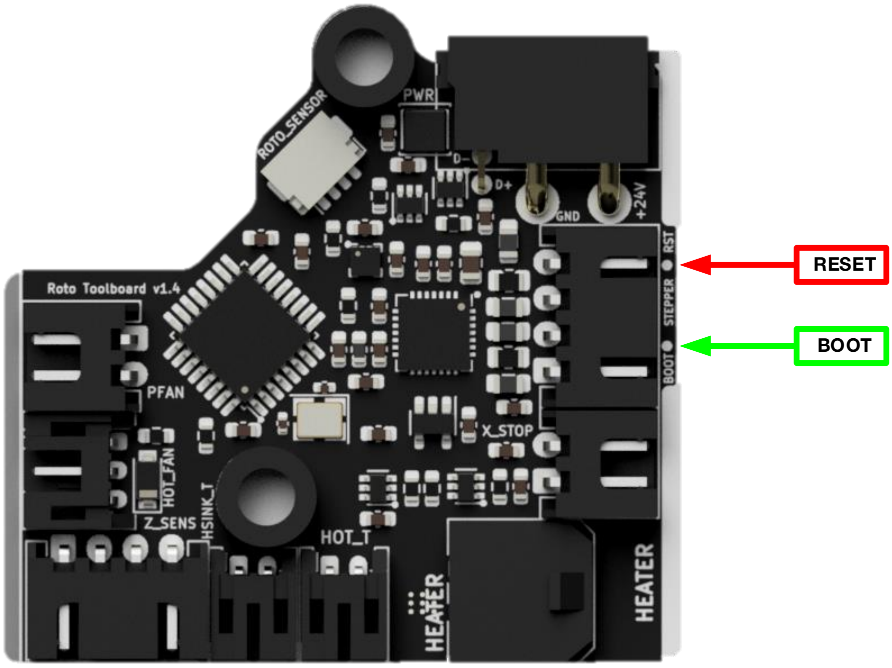
The fan speed input signal is feed back to the microcontroller for FAN RPM measurement, however for a reliable speed signal the Roto cooling fan shall be used in ON / OFF mode, without PWM control.



3 Microcontroller programming

1. Connect the board to the host Raspberry Pi via USB adapter cable.
2. This step depends on if your board has firmware on it or not already:
   a) If you have a pre-flashed board first must enter DFU mode as follows:
      - press the BOOT and RESET switch,
      - release the RESET while keeping the BOOT button pressed,
      - release BOOT switch button after 3 seconds.

   b) If you have a new un-flashed board:
   - The MCU default's to DFU mode if there is no other firmware installed, confirm the board enters DFU mode in 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
5. Run dfu-util --list from the command prompt
   - note the text inside the [xxxx:yyyy]
6. Run cd ~/klipper from the command line to enter the Klipper directory
7. Run make menuconfig settings should be:
   - Cristal oscillator – 8Mhz

```
(Top)
                    Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32F042)  --->
    Bootloader offset (No bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    USB ids  --->
    Optional features (to reduce code size)  --->
()  GPIO pins to set at micro-controller startup
```

- Set custom USB ID to *Rototool*

```
(Top) → USB ids
                    Klipper Firmware Configuration
(0x1d50) USB vendor ID
(0x614e) USB device ID
[ ] USB serial number from CHIPID
(SO3Tool) USB serial number
```

- Optional features (to reduce code size):

```
(Top) → Optional features (to reduce code size)
                    Klipper Firmware Configuration
[*] Support GPIO "bit-banging" devices
[ ] Support LCD devices
[*] Support external sensor devices
[*] Support lis2dw 3-axis accelerometer
[ ] Support software based I2C "bit-banging"
[ ] Support software based SPI "bit-banging"
```

- Hit Q to Exit and Save

8. Run make clean to clean up the make environment

9. Run make flash FLASH_DEVICE=xxxx:yyyy (using the xxxx:yyyy noted from step 5 usually 0483:df11)

In case you encounter some errors but still having the message of File downloaded successfully than you are good to proceed to the next step.

```
DFU mode device DFU version 011a
Device returned transfer size 2048
DfuSe interface name: "Internal Flash  "
Downloading to address = 0x08000000, size = 23004
Download        [=========================] 100%        23004 bytes
Download done.
File downloaded successfully
Transitioning to dfuMANIFEST state
dfu-util: can't detach
Resetting USB to switch back to runtime mode
roby@Machcube:~/klipper $
```

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_stm32f042x6_Rototool-if00

Copy this virtual serial port name to the Rototool config file MCU section

```
[mcu orbitool02]
serial:/dev/serial/by-id/usb-Klipper_stm32f042x6_Orbitool02-if00
restart_method: command
```

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

4 Electrical characteristics

4.1Absolute maximum ratings

Important Note: The Rototool board can withstand these limits without electrical damage, however long-term exposition to these limits is not recommended. Device absolute maximum rating is not the same with functional range.

The maximum ratings may not be exceeded under any circumstances!

Table 1 Rototool absolute maximum ratings

Nr.	Parameter	Min	Max	Unit
A1	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		1	A
A6	Hot-end Fan		0.5	A
A7	USB Data lines	-1	28*	V
A8	Hot-end and extruder temperature sensor input	-1	28*	V
A9	End stop sensor input	-1	28*	V
A10	Z sensor I/O interface	-10	+15	V
A11	Roto sensor interface inputs	-1	24	V
A12	ESD-Protection level for handling (Human Body Model, HMB)	-8	8	kV

*Should not exceed the supply voltage

4.2 Full functional operational limits

Table 2 Rototool electrical characteristics

Nr.	Parameter	Symbol	Min	Typ	Max	Unit
	General					
P1	Operating temperature range	t_o	-20		60	°C
P2	Max ambient temperature with active cooling	t_{o_cooled}			80	°C
P3	Power supply voltage	v_{PWR}	22		28	V
P4	Power supply voltage measurement accuracy	a_{PWR}	-3		3	%
	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}			1	A
P12	Part fan overcurrent switch OFF threshold	i_{OC_PART}	1.2	2	2.5	A
P13	Part fan overcurrent detection time	t_{OC_PART}			5	ms
P14	Part fan PWM frequency	f_{PART_FAN}		100		Hz
	Hot-end Cooling fan					
P15	Hot-end fan output current	i_{HOT_FAN}			0.5	A
P16	Hot-end fan current limitation	$i_{CL\_HOT\_FAN}$	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

*Parameter derating with ambient temperature