CPUCoolerTester

Reference Manual

Important Notices

- Use this kit at your own risk.

- This kit is designed for electronics enthusiasts. Soldering of surface mount parts is required.

- This test system can apply excessive stress to CPU coolers, power supplies and the test board itself. Do not leave the powered-on test board unattended.

- The heater copper plate is non-isolated, and a strong 12 V appears on it. Do not short circuit with 0 V of the power supply metal case. Note that when testing water coolers in PC case, they can be shorted via coolant water, which can cause electrolysis of the coolant water and bursting of the water cooler after a period of time.

Heater copper plate voltage
Non-isolated DC 12 V

Case voltage
Non-isolated DC 0 V

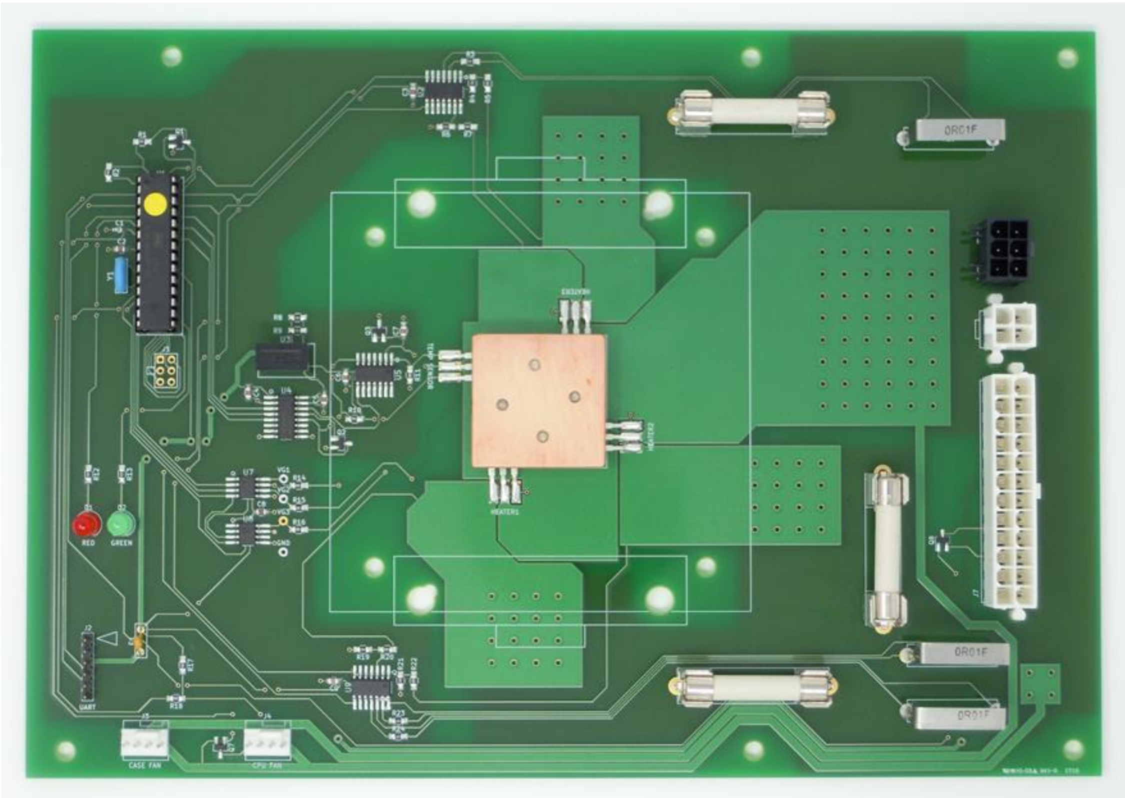
Do not short circuit!

- **Please take care not to accidentally short the heatsink and PC case with metal shaft of a screwdriver.**
- **If water-cooler is attached to the PC case, two electrodes will be shorted via coolant water, which may cause electrolysis of the coolant water and bursting of the water cooler.**

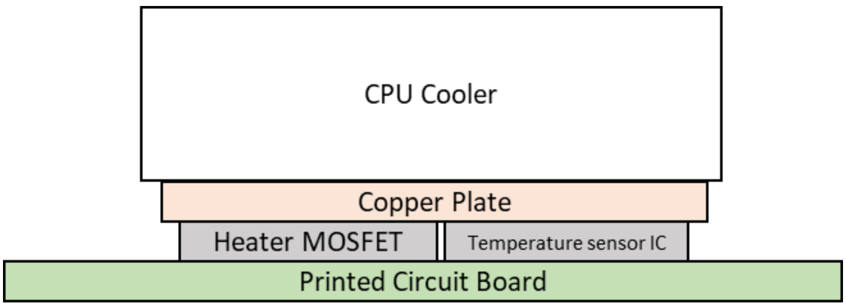
Table of Contents

1. Overview

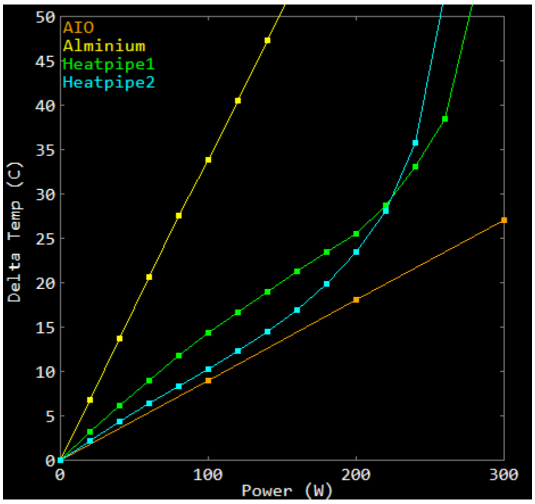
CPUCoolerTester is a DIY system kit to measure the performance of CPU coolers. This kit can apply constant power dissipation to a CPU cooler, and at the same time, it can measure temperature increases in the cooler. Heating and the temperature measurement function are implemented in one printed circuit board. The board is controlled by a Windows PC connected via a USB-UART cable. The electrical power for generating power dissipation is supplied by the ATX power supply unit connected to the board.



The heating device is power MOSFET, which is attached to the copper plate at center of the board. The temperature sensor is also attached to the copper plate. By commands from the host PC, power dissipation wattage can be set to an arbitrary value ranging from 0 W to 300 W. By measuring the temperature change for power dissipation, the performance of the CPU cooler is evaluated.

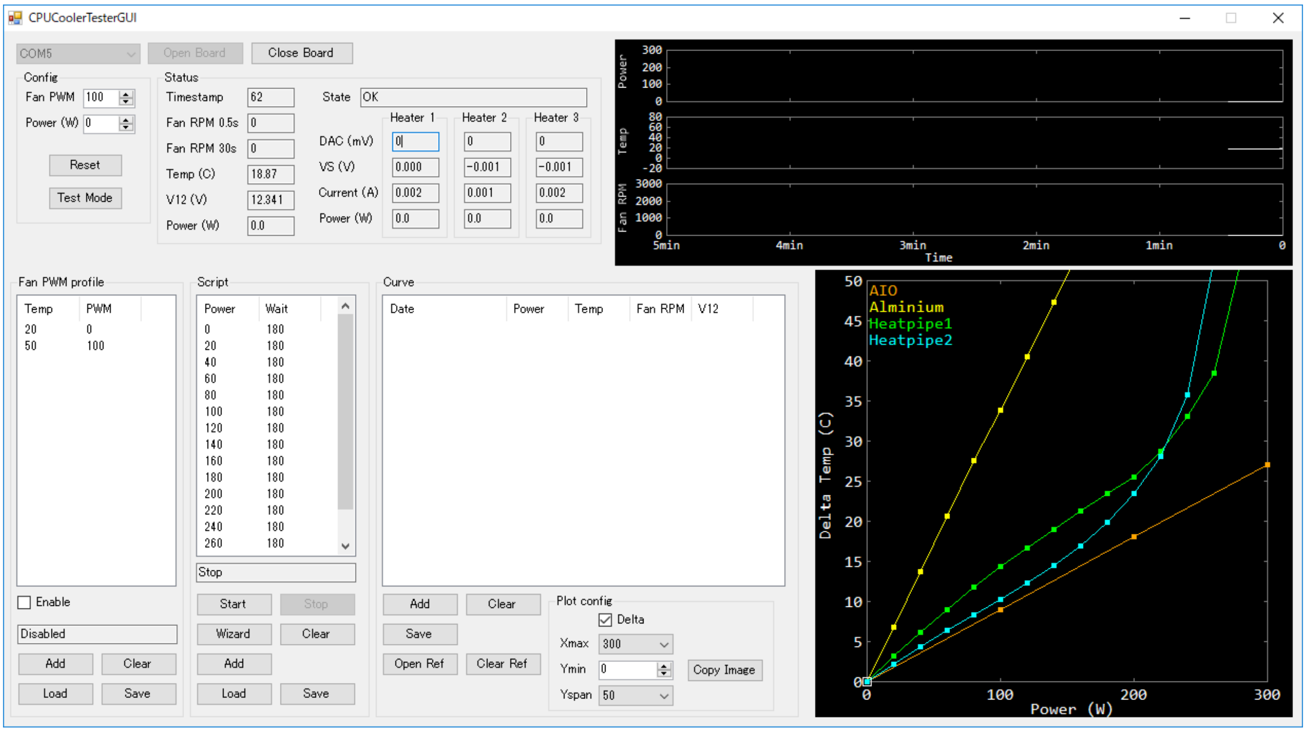


The figure below is an example of the measurement result. This plot is the temperature increase of the base plate of CPU cooler measured for power dissipation.



The board has mounting holes for LGA115x and Socket AM4 coolers. The board also has a connecter for the PWM fan, and the PWM duty ratio can be set by the host PC. The board has Micro-ATX compatible mounting holes such that the board can be mounted to the Micro-ATX compatible PC case.

The Windows application for automated measurement is provided. The application covers all of the board's functions.



2. Required tools

For building and using this kit, these tools are required.

Item	Note
ATX power supply unit	The 12 V outputs of the power supply **must be single-rail type.** The power supply unit must have at least 350 W capacity for the 12 V rail. The power supply unit must have at least one 24-pin ATX connector, one 4-pin CPU connector and one 6-pin PCIe connector.
Windows PC	Required to control the board. Windows 7 or later is supported.
Soldering iron	One that is temperature controlled with a thick iron tip (for high heat capacity) is recommended.
Solder	
Solder wick	
Solder flux	
Tweezer	
Scissor	
Nipper	
Marker pen	
Screwdriver	
Thermal grease	
CPU cooler	LGA 115x or Socket AM4 are supported.
AM4 backplate	Certain types of AM4 coolers need a backplate borrowed from the motherboard. Please refer to your AM4 cooler's manual.

3. Parts list

If package contains more parts than Qty number, they are spare.

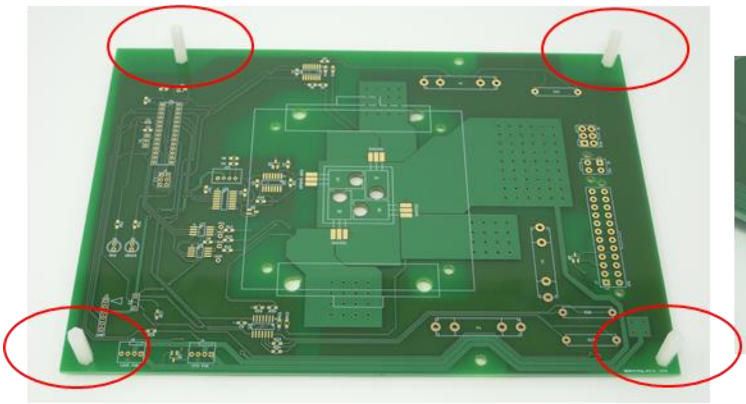
#	Bag	Description	Qty
1		Printed circuit board	1
2	A	Plastic spacer	4
3	A	Plastic nut	4
4	B	Resistor 10kΩ 0805	13
5	C	Resistor 1kΩ 0805	11
6	D	Ceramic capacitor 0.01µF 0603	1
7	E	Ceramic capacitor 0.1µF 0805	8
8	F	Small signal MOSFET 2N7002	5
9	F	Analog-to-Digital converter MCP3424	3
10	F	Digital isolator Si8631	1

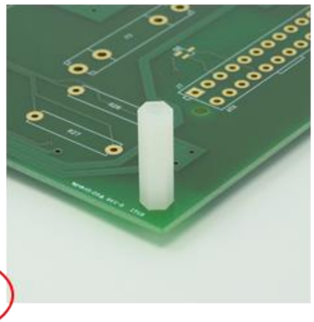
11	G	Digital-to-Analog converter MCP4822	2
12	H	LED (red)	1
13	H	LED (green)	1
14	I	Ceramic resonator 8 MHz	1
15	I	Polyswitch 100 mA	1
16	J	Isolated DC-DC converter RM-0505S	1
17	J	Resistor 10 mΩ 5 W	3
18	K	DIP28 socket	1
19	K	6-pin UART connector	1
20	K	Fan connector	2
21	K	24-pin ATX power supply connector	1
22	K	4-pin CPU power supply connector	1
23	K	6-pin PCIe power supply connector	1
24	L	Dummy fuse (glass tube)	1
25	L	Fuse clip	6
26	L	Fuse DC 10A (ceramic tube)	3
27	M	Plastic screw M3 5 mm (flame retardant)	4
28	N	Temperature sensor IC LM35DT	1
29	N	Power MOSFET IRL3713	3
30	O	Copper plate	1
31	O	Double sided tape (flame retardant)	1
32	P	Programmed AVR microcontroller	1
33		USB-UART cable TTL-232R-5V	1
34	Q	Rubber spacer (flame retardant)	1

4. Build

4.1. Installing plastic spacers

Take the plastic spacers and nuts in bag A, and then install them to the four locations on the board.





4.2. Soldering 10 kΩ resistor

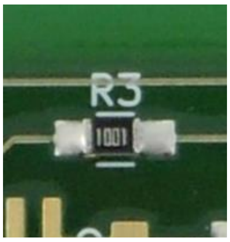
Solder the 10 kΩ resistor in bag B to the following locations.

R1, R2, R5, R7, R10, R11, R14, R15, R16, R17, R18, R20, R22

4.3. Soldering 1 kΩ resistor

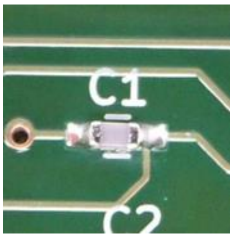
Next, solder the 1 kΩ resistor in bag C to the following locations.

R3, R4, R6, R8, R9, R12, R13, R19, R21, R23, R24



4.4. Soldering 0.01 μF ceramic capacitor

Solder the 0.01 μF ceramic capacitor in bag D to location C1.



4.5. Soldering 0.1 µF ceramic capacitor

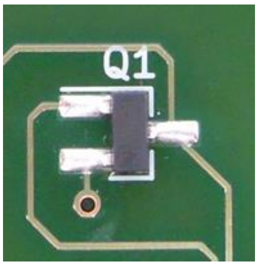
Next, solder the 0.1 µF ceramic capacitor in bag E to the following locations.

C2, C3, C4, C5, C6, C7, C8, C9

4.6. Soldering small-signal MOSFET 2N7002

Solder the small-signal MOSFET 2N7002 in bag F to the following locations.

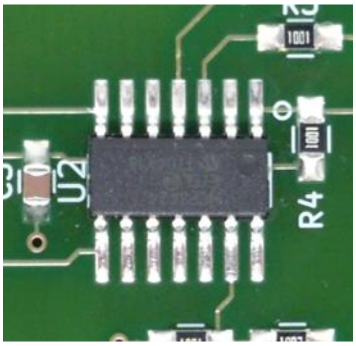
Q1, Q2, Q3, Q7, Q8



4.7. Soldering analog-to-digital converter MCP3424

Solder the analog-to-digital converter IC MCP3424 in bag F to the following locations.

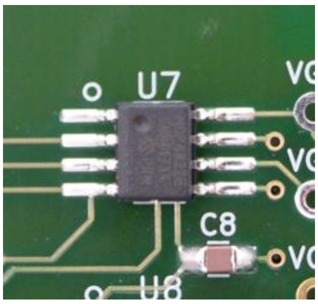
U2, U5, U9



4.8. Soldering digital-to-analog converter MCP4822

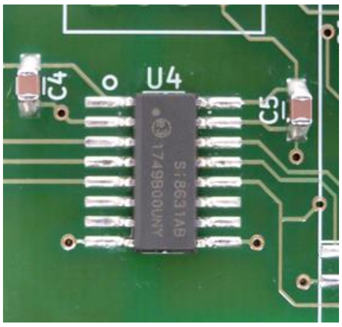
Next, solder the digital-to-analog converter IC MCP4822 in bag G to the following locations. (Note that bag G in the package may be rolled into cylinder form to make it more shock resistant.)

U7, U8



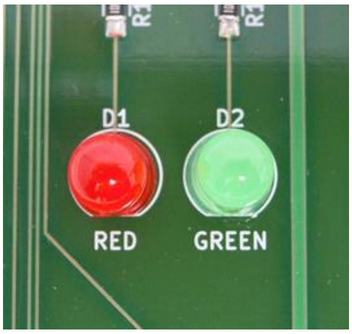
4.9. Soldering digital isolator Si8631

Next, solder the digital isolator IC Si8631 in bag F to the location U4.



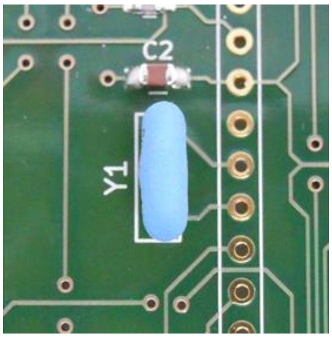
4.10. Soldering LEDs

Take two LEDs in bag H. Next, solder the red LED to location D1, and the green LED to location D2. To ensure the proper orientation of the part, match the silk-screen mark and the outline of LED.



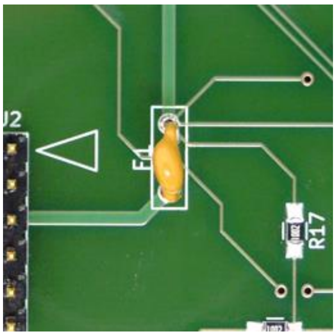
4.11. Soldering 8 MHz ceramic resonator

Solder the 8 MHz ceramic resonator in bag I to location Y1.



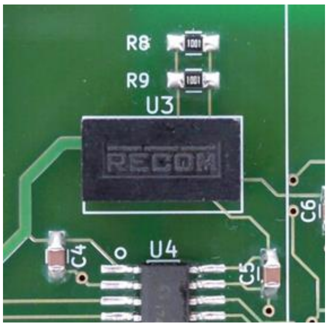
4.12. Soldering 100 mA polyswitch

Solder the 100 mA polyswitch in bag I to location F1.



4.13. Soldering DC-DC converter RM-0505S

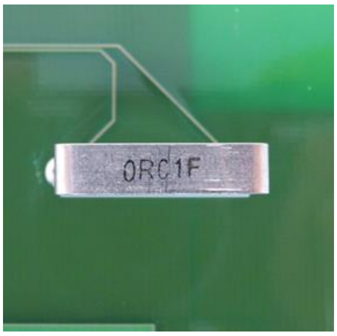
Solder the DC-DC converter RM-0505S in bag J to location U3. Match the silk-screen mark and the outline of the device.



4.14. Soldering 10 mΩ resistor

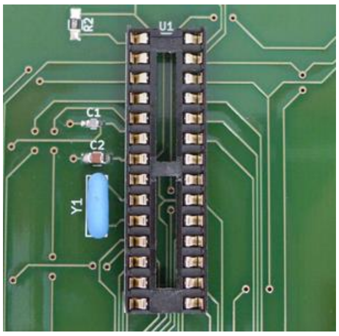
Solder the 10 mΩ 5 W resistor in bag J to the following locations.

R25, R26, R27



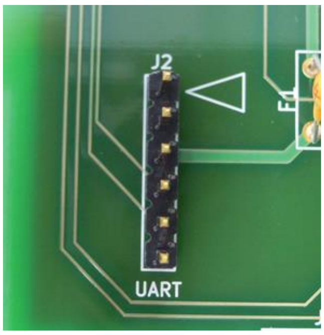
4.15. Soldering DIP28 IC socket

Solder the DIP28 IC socket in bag K to location U1.



4.16. Soldering UART connector

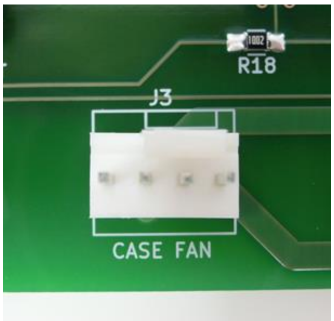
Next, solder the 6-pin UART connector in bag K to location J2.



4.17. Soldering fan connector

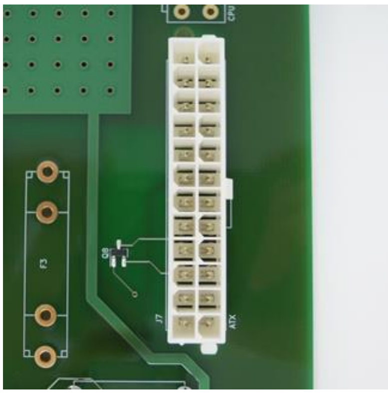
Next, solder the fan connector in bag K to the following locations.

J3, J4



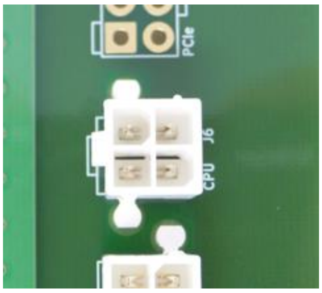
4.18. Soldering 24-pin ATX power supply connector

Next, solder the 24-pin ATX power supply connector in bag K to location J7.



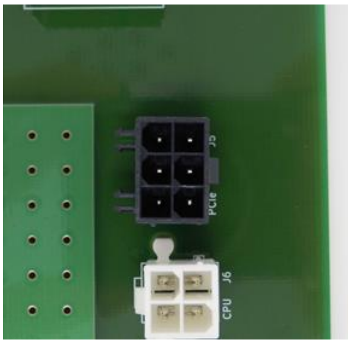
4.19. Soldering 4-pin CPU power supply connector

Next, solder the 4-pin CPU power supply connector in bag K to location J6.



4.20. Soldering 6-pin PCIe power supply connector

Next, solder the 6-pin PCIe power supply connector in bag K to location J5. **Ensure the proper orientation of the part** by matching the silk-screen mark and the outline of the part.

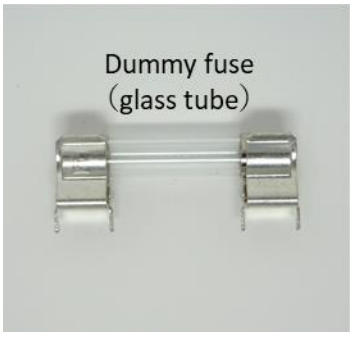


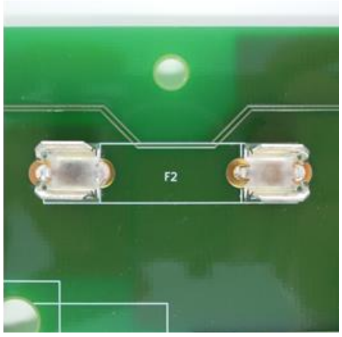
4.21. Soldering fuse clip

For the correct placement of fuse clips, use the dummy fuse (glass tube) in bag L. Attach the fuse clip to both ends of the dummy fuse, insert it into the board, and solder the leads of the fuse clip. This method provides the correct relative position of the pair of fuse clips to each other. Afterwards, remove the dummy fuse and reuse it to solder the next pair of fuse clips.

Solder the fuse clips in bag L to the following locations.

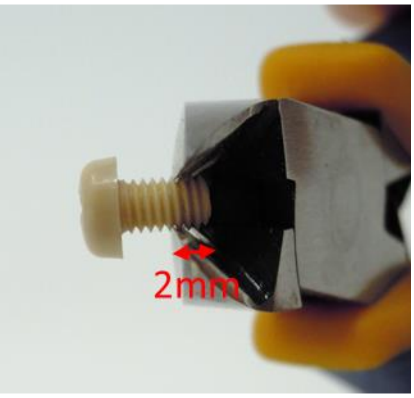
F2, F3, F4



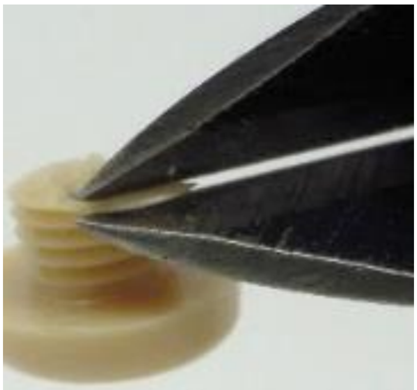



4.22. Assembling and soldering of heater unit

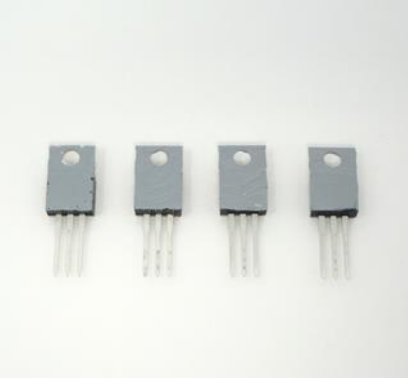
Take the four plastic screws in bag M, and trim the screw end 2 mm using nipper. Be careful of scattering fragments with high velocity.



Using the nipper, remove the burr on screw end.



Next, from bag N, take three heater MOSFET IRL3717 and one temperature sensor IC LM35DT, and apply thermal grease to them.



Next, take the copper plate from bag O. The copper plate has a top side (contact with CPU cooler) and a bottom side (contact with heaters and the temperature sensor).

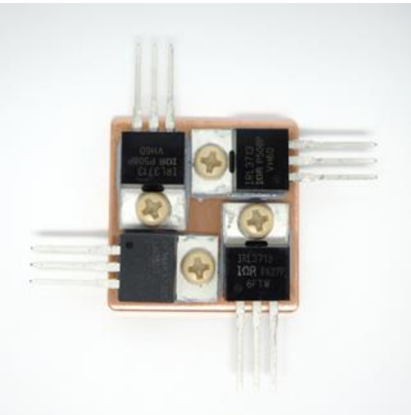
Top side
（contact with CPU cooler）

Bottom side
（contact with heater and
temperature sensor）

Now, attach heater MOSFET IRL3717 and temperature sensor IC LM35DT to the bottom side of the copper plate using trimmed plastic screws.

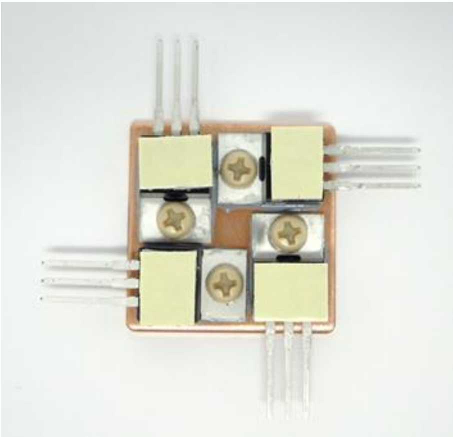
When the plastic screw is tightened too much, the screw thread will be stripped, and the screw will be loose. If this happens, use a new screw. The kit package contains spare screws.

Make sure that the end tip of the screw is not sticking out of the top side surface of the plate. If this happens, remove the screw and trim it slightly more using the nipper.



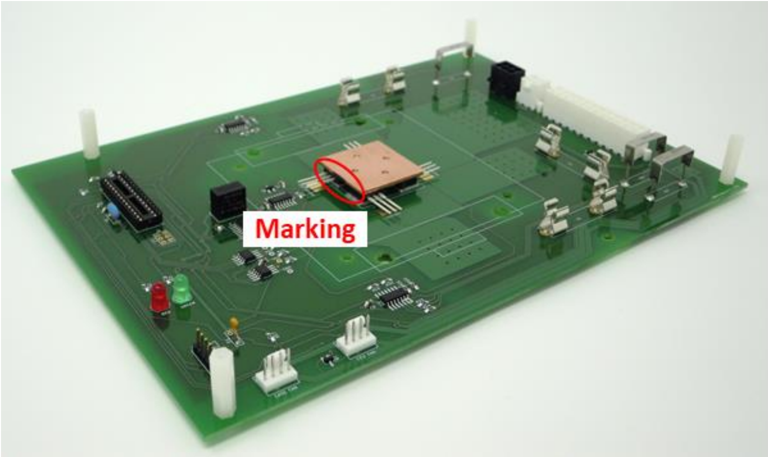
Next, apply marking, using the marking pen, to the edge of the copper plate where the leads for the temperature sensor IC LM35DT are located. (See figure below.)

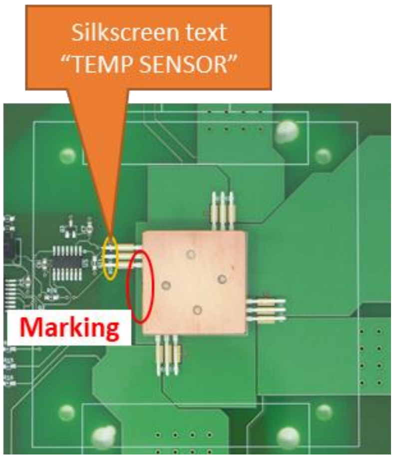
Next, take the double sided tape out of bag O. Cut it into the appropriate size using scissors and stick it to the top surface of the four ICs.



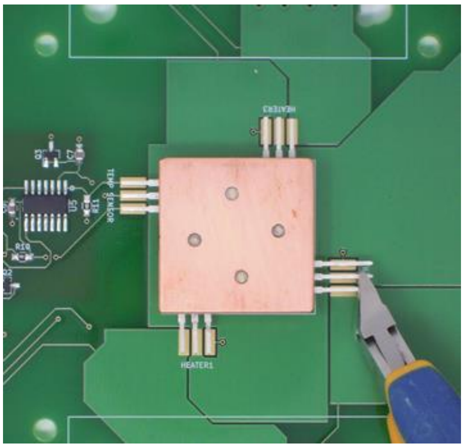
Next, temporarily place the heater unit on the board, keeping the tape sticker covered by siliconed paper, and ensure that the leads for the 4 ICs are aligned with lands on the board. If not, loosen the screw and fix it. Moreover, ensure that the orientation of heater unit is correct by matching the marking and silkscreen text “TEMP SENSOR” on the board.

Afterwards, remove the siliconed paper and attach the heater unit to the board. **Again, please ensure that the orientation is correct by checking the marking and silkscreen text “TEMP SENSOR”.**

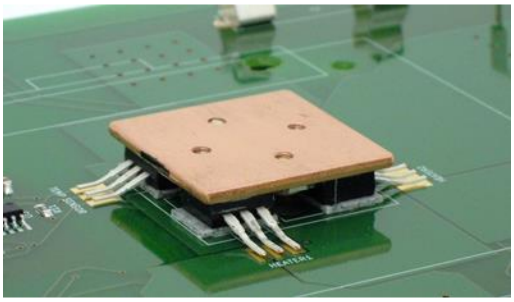




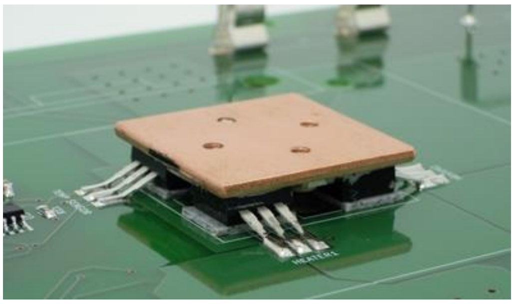
Next, trim the leads of the four ICs to match the length with the size of the land.



Push the leads for the four ICs using your fingers to fit them as close as possible to the land.

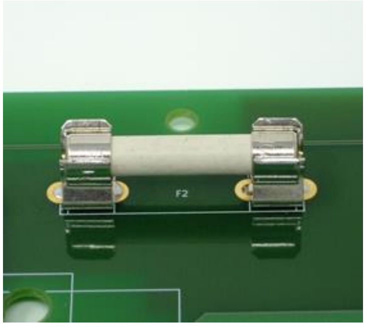


Solder the leads for the four ICs.



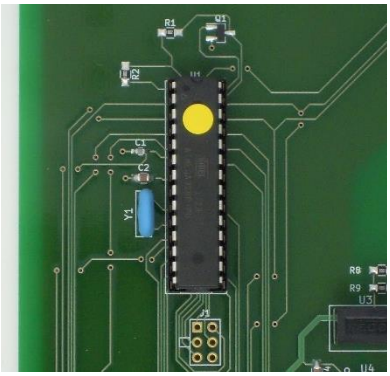
4.23. Installing DC 10A fuse

Install the DC 10A fuse (ceramic tube) in bag L to the fuse clips at locations F2, F3 and F4.



4.24. Installing programmed AVR microcontroller

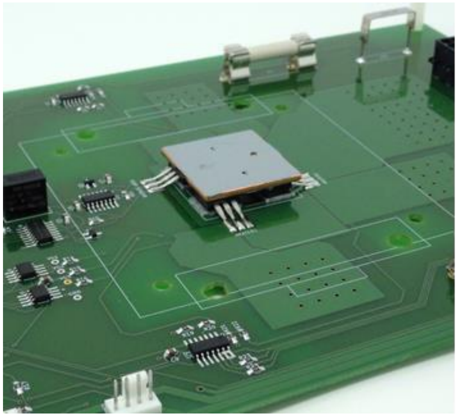
Install the programmed AVR microcontroller in bag P to the socket at location U1.



5. Measurement

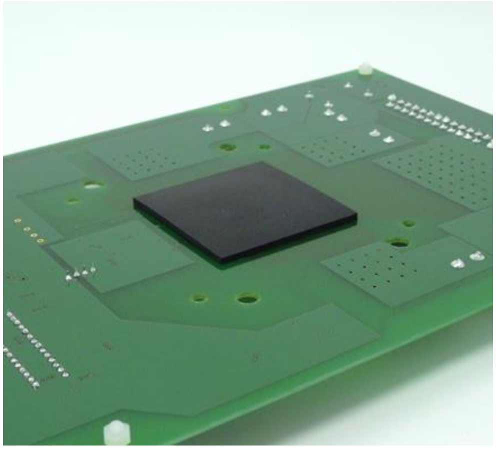
5.1. Installing CPU cooler

Apply thermal grease to the top surface of the copper plate. Afterwards, install the CPU cooler and connect the CPU fan to the CPU fan connector (J4).



5.1.1. Notes for LGA115x coolers

If your LGA115x cooler uses a backplate, insert the rubber spacer in bag Q between the backplate and the board.

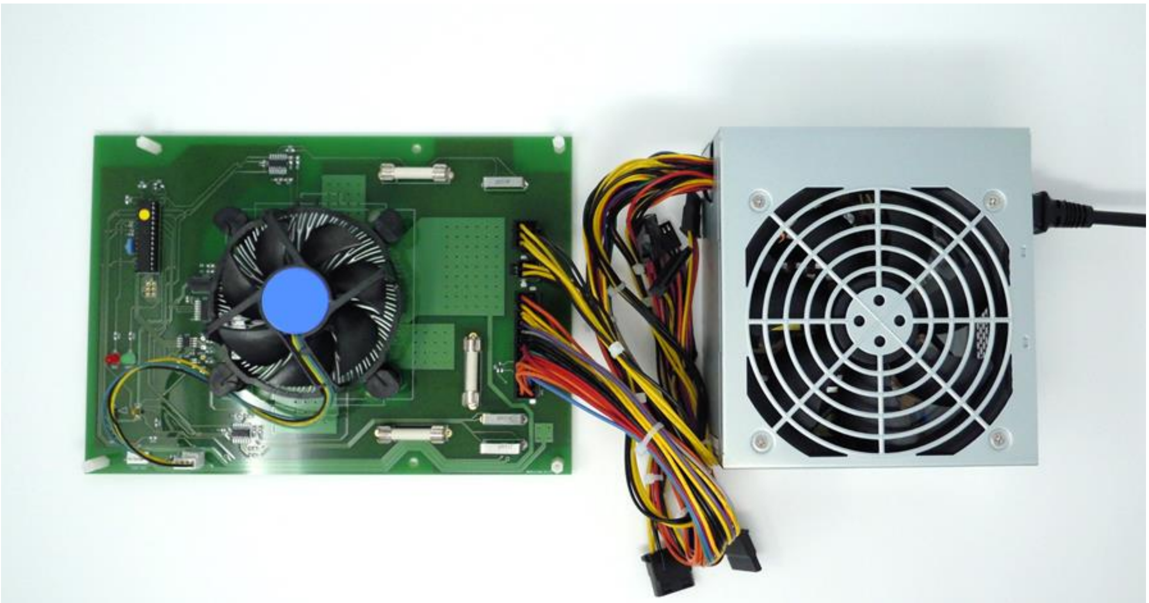


5.1.2. Notes for Socket AM4 coolers

If your Socket AM4 cooler needs a standard backplate, which is provided as part of the motherboard, please borrow it from motherboard and use it.

5.2. Connecting ATX power supply

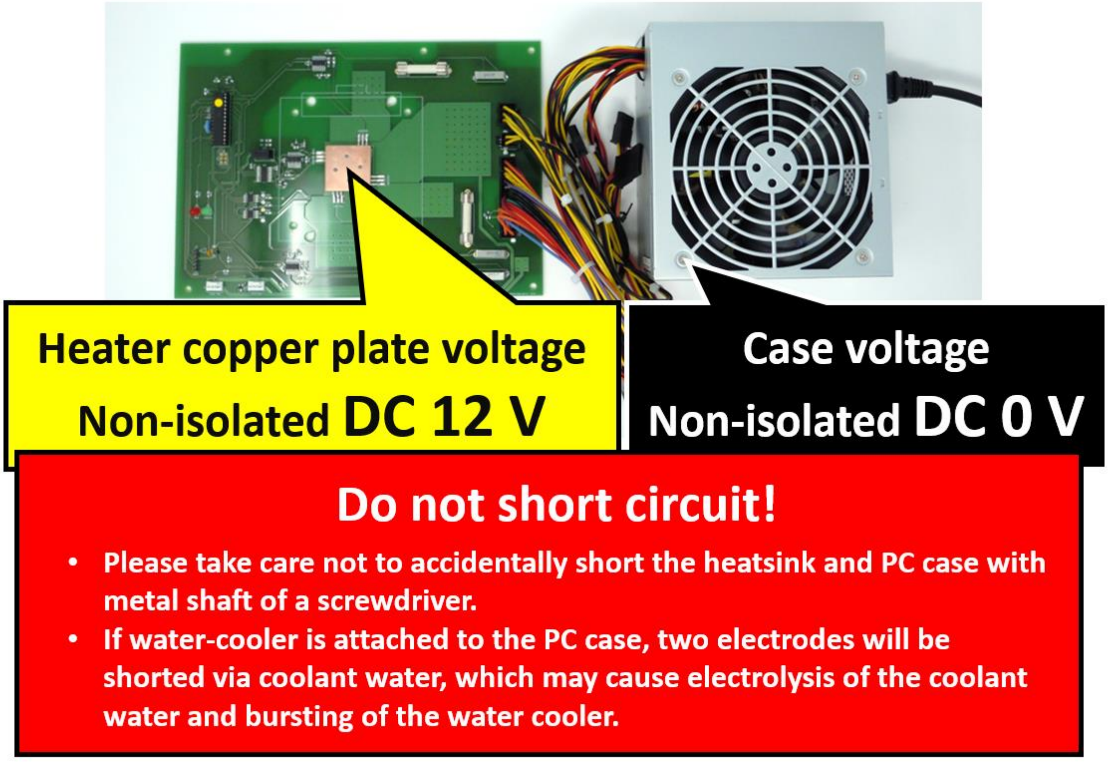
Next, connect the ATX power supply to the board. All three connectors must be connected, including the ATX power supply connector (J7), the CPU power supply connector (J6) and the PCIe power supply connector (J5). Please note that **it is unsafe to operate the board when some but not all of the power connectors are connected,** because a significant amount of current will be concentrated in a few wires, which causes overheating of the wire and it can catch fire.



Please note that **the heater copper plate is not isolated, and powerful DC 12 V appears on it. The metal case of the power supply unit is also non-isolated, and its voltage is DC 0 V. Thus, shorting these two electrodes will cause a direct short of the powerful DC 12 V rail.** Please take care not to do it.

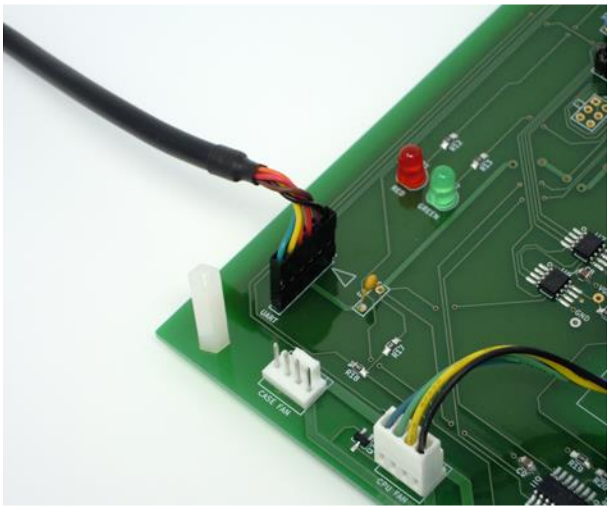
Short circuiting may be caused by accidentally shorting the PC case and the cooler with metal shaft of a screwdriver.

Moreover, shorting may be caused in the water cooler installed to the PC case through the coolant water. The short path in this case is the copper plate to the water block to the coolant water to the radiator to the PC case to the power supply metal case. If shorting through this path happens**, electrolysis of the coolant water will generate gas in a water loop, which will cause bursting of the water cooler after a period of time.** To prevent this outcome, measure the water cooler on a non-conductive floor outside of the PC case, or isolate the power supply metal case from the PC case. Note that maintaining isolation can be a complex task and even small mistake can lead to catastrophic result (the burst of the water cooler after a period of time). Therefore, please be careful when measuring the water cooler.



5.3. Connecting the USB-UART cable

Connect the USB-UART cable to the UART connector (J2) of the board with the correct orientation. Both the connector and the cable has triangle mark for pin 1. Black wire is pin 1.



5.4. Download and start CPUCoolerTesterGUI

Download the zip archive file for the application, CPUCoolerTesterGUI, from https://cpucoolertester.com/download. Unzip it and run the executable, CPUCoolerTesterGUI.exe.

5.5. Connecting the board to PC

Now, connect the USB-UART cable to the USB port of the PC.

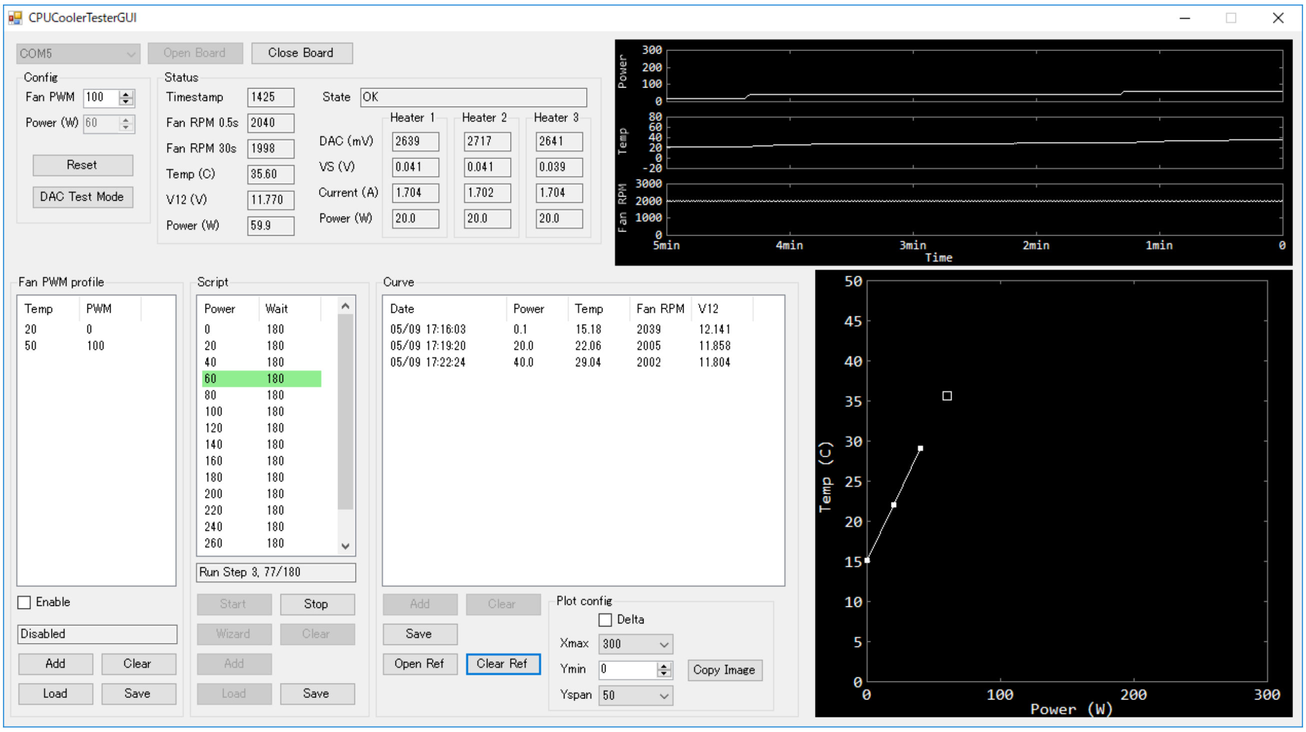
5.6. Opening the board in application

Next, click the “Open Board” button to open the board. The application will start periodic readings of voltage, current and temperature data. When the board is collecting sensor data using ADC, the green LED is turned on. When the board detects anything suspicious, such as no command from PC for some time or a loss of 12 V rail, it enters the alarm state. When the board is in the alarm state, the red LED turns on, and the power supply is turned off. To recover from the alarm state, the “reset” command has to be sent from the PC. To send it, click the “Reset” button.

5.7. Start automated measurements

Now, click the “Start” button to start automated measurements. With automated measurement, the power dissipation value is set and, after a specified wait time for converging, the temperature of the copper plate is captured. By default, the power dissipation is swept from 0 W to 300 W by 20 W intervals, and the wait time is 180 seconds. Captured temperature vs the power dissipation curve is plotted in the window in real time. When the copper plate temperature exceeds 80°C, automated measurement is stopped for safety.

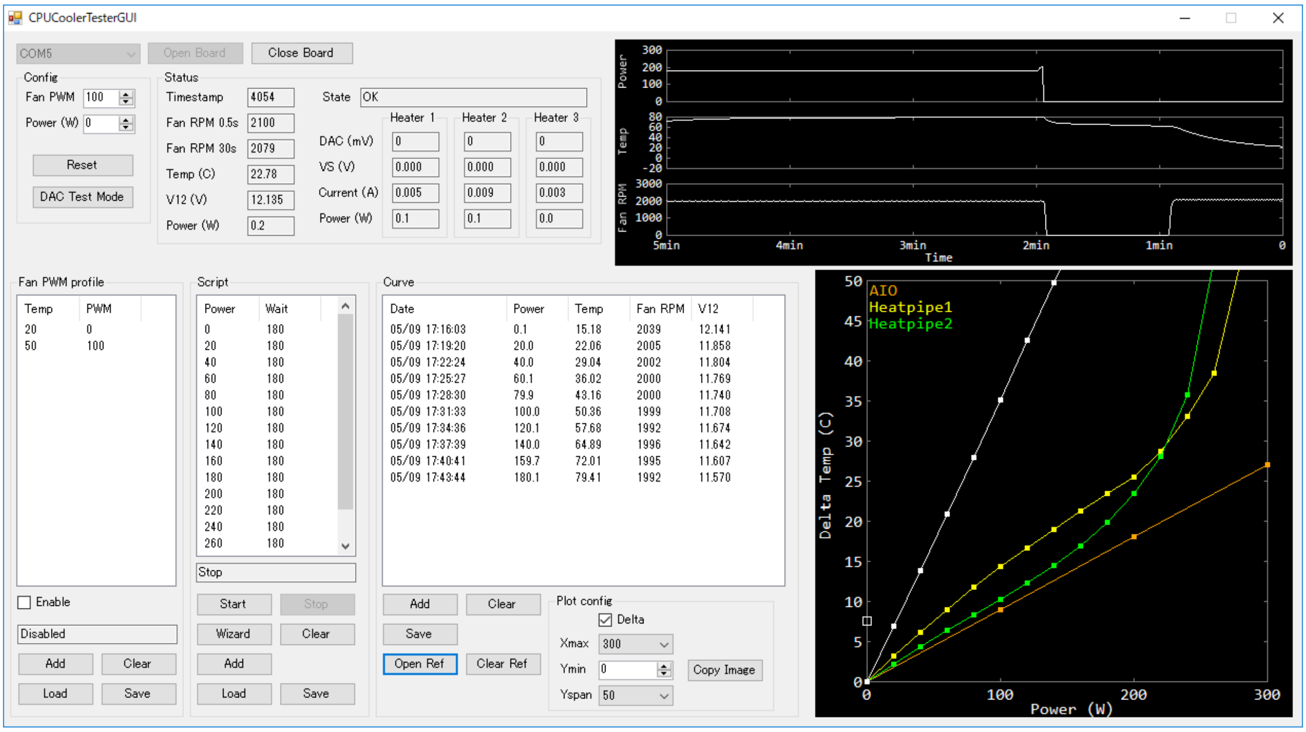
When the heater is enabled, excessive load is applied to the board, the cooler and the power supply, which can cause unexpected results. **For safety, please never leave the heater-enabled board unattended.**



5.8. Saving and comparing measurement results

To save measured data to a CSV file, click the “Save” button in the “Curve” group box. A CSV file can be read into external applications, such as Excel or a text editor.

CPUCoolerTesterGUI can load saved CSV files as reference data. Loaded reference data are plotted along with current measurement data. It can load multiple reference data at same time. By checking the “Delta” checkbox, delta temperature from the initial datapoint is plotted, which is useful to compare multiple data.



6. Testing board functions

This section describes methods for testing board functions.

6.1. List of state codes

When the board detects something suspicious, it enters the alarm state, and the ATX power supply is disabled. The state code can be read via serial interface. CPUCoolerTesterGUI displays this code in the “State” textbox.

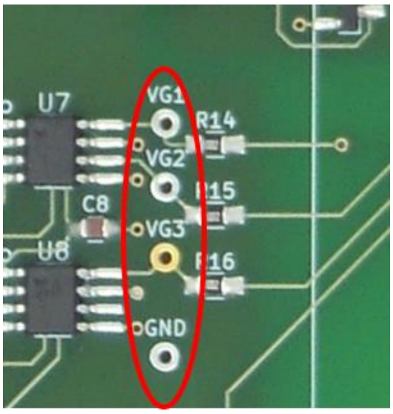
The state code represents the state of the board. In the alarm state, the state code represents the source of alarm activation. The following table lists all state codes.

State code	Description	Possible cause(s)
OK		
ALARM_WATCHDOG_TIMEOUT	The board hasn’t received a valid command from the host for more than 15 seconds.	
ALARM_V12_OUT_OF_RANGE	The voltage level of 12 V rail from ATX power supply is out of valid range (10.8 V to 13.2 V).	ATX power supply is not connected or powered off. Power supply is not capable of supplying

		sufficient power. Heater MOSFET has failed in short mode and PSU detects 12 V rail shorting and is shutting down.
ALARM_OVERHEAT	Copper plate temperature exceeded 80C.	The cooler could not handle the applied power dissipation.
ALARM_ADC_OVERRANGE	Input voltage overrange to ADC is detected.	Input circuit to ADC is not functioning properly.
ALARM_I2C_ERROR_ADC1	The microcontroller did not receive I2C ACK from ADC1(U9).	The I2C connection between microcontroller and ADC1(U9) is not functioning.
ALARM_I2C_ERROR_ADC2	The microcontroller did not receive I2C ACK from ADC2(U2).	The I2C connection between microcontroller and ADC2(U2) is not functioning.
ALARM_I2C_ERROR_ADC3	The microcontroller did not receive I2C ACK from ADC3(U5).	The I2C connection between microcontroller and ADC3(U5) is not functioning.
ALARM_DAC_TEST_MODE	DAC test mode	
ALARM_HEATER1_VS_OUT_OF_RANGE	The source voltage level of heater 1 is out of valid range.	Contact failure of fuse clip. Fuse blow (possibly due to failure of heater MOSFET in short mode).
ALARM_HEATER1_OVER_CURRENT	Current of heater 1 exceeded 10 A.	The heater was in a thermally unstable condition. Heater MOSFET has failed in short mode.
ALARM_HEATER1_UNCONTROLLABLE	The current of heater 1 did not respond to gate voltage change.	The current sense circuit is in a thermally unstable state. Heater MOSFET has failed.

6.2. Testing DACs

The special state for DAC testing is implemented in firmware. By clicking the “DAC test mode” button, a command is sent to the board, and it enters DAC test mode. The ATX power supply is not needed when performing this test.
In DAC test mode, the microcontroller outputs square wave signals to the DACs in 16-second periods. The amplitude is 0.1 V for ch1, 0.2 V for ch2 and 0.3 V for ch3. (The amplitude is low enough to prevent enabling heater MOSFET when 12 V rail exists.) If DAC outputs this waveform correctly, DAC is functioning properly.
The board has dedicated test pads.



6.3. Testing the temperature sensor

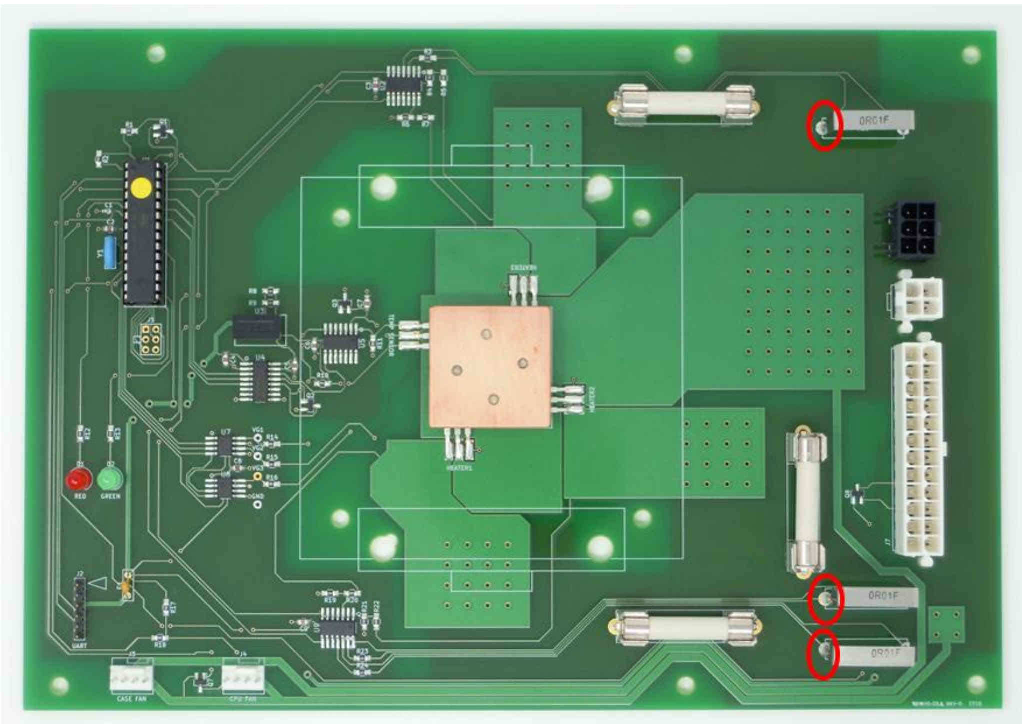
To test the temperature sensor, touch the copper plate by hand and check the temperature value on the application. The ATX power supply is not needed when performing this test.

6.4. Testing the current sense circuit

The application displays the heater current value in “Current (A)” text box. The current sense circuit monitors small voltage drop due to the heater current across 10 mΩ sense resistor. Since the amplitude of the signal is small, the measurement can fluctuate by the Seebeck effect.

Due to the Seebeck effect, small voltage that is proportional to the temperature difference between two terminals of resistor appears across the resistor. This voltage is a source of error in an actual measurement but this effect can also be used to test the current sense circuit.

To test it, touch the current sinking terminal of the current sense resistor (see figure below) by hand and check to see if the current read in application moves to positive by a few 10 mA. The ATX power supply is not needed when performing this test.



7. Controlling board with custom software

7.1. Serial interface configuration

CPUCoolerTester is connected to the PC via FTDI TTL-232R-5V USB-Serial cable. Therefore, the board is accessed as a COM port in Windows and can be controlled by any programming language that can access COM port. The following are parameters for accessing COM port.

Baud rate	38.4 kbps
Parity	None
Data bits	8
Stop bits	1
Flow control	None
Newline char	LF

7.2. Communication protocols

The board is controlled by commands from the host PC. A command is always a single line of text terminated by LF sent from the host to the board. There are many types of commands, which are determined by first character of the command. The board is case-insensitive when interpreting commands.

When the board receives a command, it sends a response back to the host. A response is always a single line of text terminated by LF. The board sends the response for the command without delay, regardless of the state of the board. For example, to perform the VERSION command, the host sends the text `“V¥n”` to the board, and the board sends text `“CPUCoolerTester 1.0.0¥n”` to the host without delay.

Depending on the type of command, arguments follow the first character. Moreover, responses can have return values. Each of arguments and return values are formatted as four-digit hexadecimal number, and its type can be uint16 or real.

The uint16 type value is formatted in conventional hexadecimal format.

The real type represents a numeric value from -32766 to 32766, positive infinity, negative infinity and NaN. They are formatted in four-digit hexadecimal number as listed in table below.

text	real
`“0000”`	0
`“0001”`	1
`“7FFE”`	32766
`“7FFF”`	Infinity
`“FFFF”`	-1
`“8002”`	-32766
`“8001”`	-Infinity
`“8000”`	NaN

For example, the CONFIG command starts with the `‘C’` character and takes two arguments. The first argument is the target power dissipation value in uint16, 1 LSB=1 W format. The second argument is the fan PWM duty ratio in uint16, 1 LSB=1% format. To set the power dissipation to 95 W and PWM to 50% by the CONFIG command, the host sends the command `”C005F0032¥n”` and the board sends the response `“A¥n”`. The response `“A¥n”` represents “acknowledge”.

When the board receives a line that cannot be interpreted as a valid command, it sends the response `“N¥n”`, which represents “not acknowledge”.

When the host sends command Y while the board is responding to command X, the command Y may be discarded.

7.3. Commands

7.3.1. CONFIG command

The CONFIG command sets the target power dissipation and the PWM duty cycle. The CONFIG command starts with `‘C’` character and takes two arguments, as listed in following table.

Offset	Name	Value	Type	Unit(LSB)
0	TargetPower	Target value of power dissipation	uint16	1 W
1	FanPWM	Fan PWM duty cycle	uint16	1%

The response is always `”A¥n”`.

7.3.2. STATUS command

The STATUS command reads the latest status/sensor values from the registers in the board.

The syntax of command is `“S¥n”`.

The response is concatenation of return values listed in following table.

Offset	Name	Value	Type	Unit(LSB)
0	Timestamp	Timestamp value which is incremented for each update	uint16	
1	State	State code of the board	uint16	
2	FanRPM_500ms	Fan RPM value averaged in last 0.5 seconds	uint16	RPM
3	FanRPM_30s	Fan RPM value averaged in last 30 seconds	uint16	RPM
4	Power	Total power dissipation (sum of three heater channels) calculated from sensor readings	real	0.1 W
5	Temp	Reading of temperature sensor in deg C	real	0.01K
6	V12	Voltage level of 12 V rail	real	1 mV
7	Heater1_DAC	DAC voltage level set as MOSFET gate voltage of heater 1	uint16	1 mV
8	Heater1_VS	Reading of source voltage of heater 1 MOSFET	real	1 mV
9	Heater1_Current	Reading of heater 1 current	real	1 mA
10	Heater1_Power	Power dissipation of heater 1 calculated from sensor readings	real	0.1 W
11	Heater2_DAC	DAC voltage level set as MOSFET gate voltage of heater 2	uint16	1 mV
12	Heater2_VS	Reading of source voltage of heater 2 MOSFET	real	1 mV
13	Heater2_Current	Reading of heater 2 current	real	1 mA

14	Heater2_Power	Power dissipation of heater 2 calculated from sensor readings	real	0.1 W
15	Heater3_DAC	DAC voltage level set as MOSFET gate voltage of heater 3	uint16	1 mV
16	Heater3_VS	Reading of source voltage of heater 3 MOSFET	real	1 mV
17	Heater3_Current	Reading of heater 2 current	real	1 mA
18	Heater3_Power	Power dissipation of heater 3 calculated from sensor readings	real	0.1 W

The following is an example of a response to the STATUS command.

”F18C0000083408100000071 82F530000000000001000000000000000100000000FFFF00010000¥n”

The register values returned as a response to the STATUS command are updated by 0.5 seconds interval. For each update, the Timestamp value is increased incrementally.

The State value is represented based on following enum definition.

```
public enum State : UInt16
{
    OK = 0x0000,

    ALARM_WATCHDOG_TIMEOUT = 0x0001,
    ALARM_V12_OUT_OF_RANGE = 0x0002,
    ALARM_OVERHEAT = 0x0003,
    ALARM_ADC_OVERRANGE = 0x0004,
    ALARM_I2C_ERROR_ADC1 = 0x0005,
    ALARM_I2C_ERROR_ADC2 = 0x0006,
    ALARM_I2C_ERROR_ADC3 = 0x0007,
    ALARM_DAC_TEST_MODE = 0x0008,

    ALARM_HEATER1_VS_OUT_OF_RANGE = 0x0009,
    ALARM_HEATER1_OVER_CURRENT = 0x0010,
    ALARM_HEATER1_UNCONTROLLABLE = 0x0011,

    ALARM_HEATER2_VS_OUT_OF_RANGE = 0x0109,
    ALARM_HEATER2_OVER_CURRENT = 0x0110,
    ALARM_HEATER2_UNCONTROLLABLE = 0x0111,

    ALARM_HEATER3_VS_OUT_OF_RANGE = 0x0209,
    ALARM_HEATER3_OVER_CURRENT = 0x0210,
    ALARM_HEATER3_UNCONTROLLABLE = 0x0211,
```

```
}
```

7.3.3. RESET command

The RESET command resets the state of the board. Thus, it deactivates alarm, sets the TargetPower to 0 W and the FanPWM to 100%.

The syntax of RESET command is `”R¥n”` and response is `”A¥n”`.

7.3.4. VERSION command

The VERSION command is used to read text that represents version of firmware.

The syntax is `“V¥n”` and a response is text that represents the version of firmware, for example `“CPUCoolerTester 1.0.0¥n”`.

7.3.5. TEST_MODE command

The TEST_MODE command sets the board into test mode for testing the DAC function. The syntax is `“T¥n”` and response is `“A¥n”`.

7.4. Watchdog timer

The board has the watchdog timer function. When the board hasn’t received a valid command from the host for 15 seconds, it enters the alarm state with the state code `ALARM_WATCHDOG_TIMEOUT`.

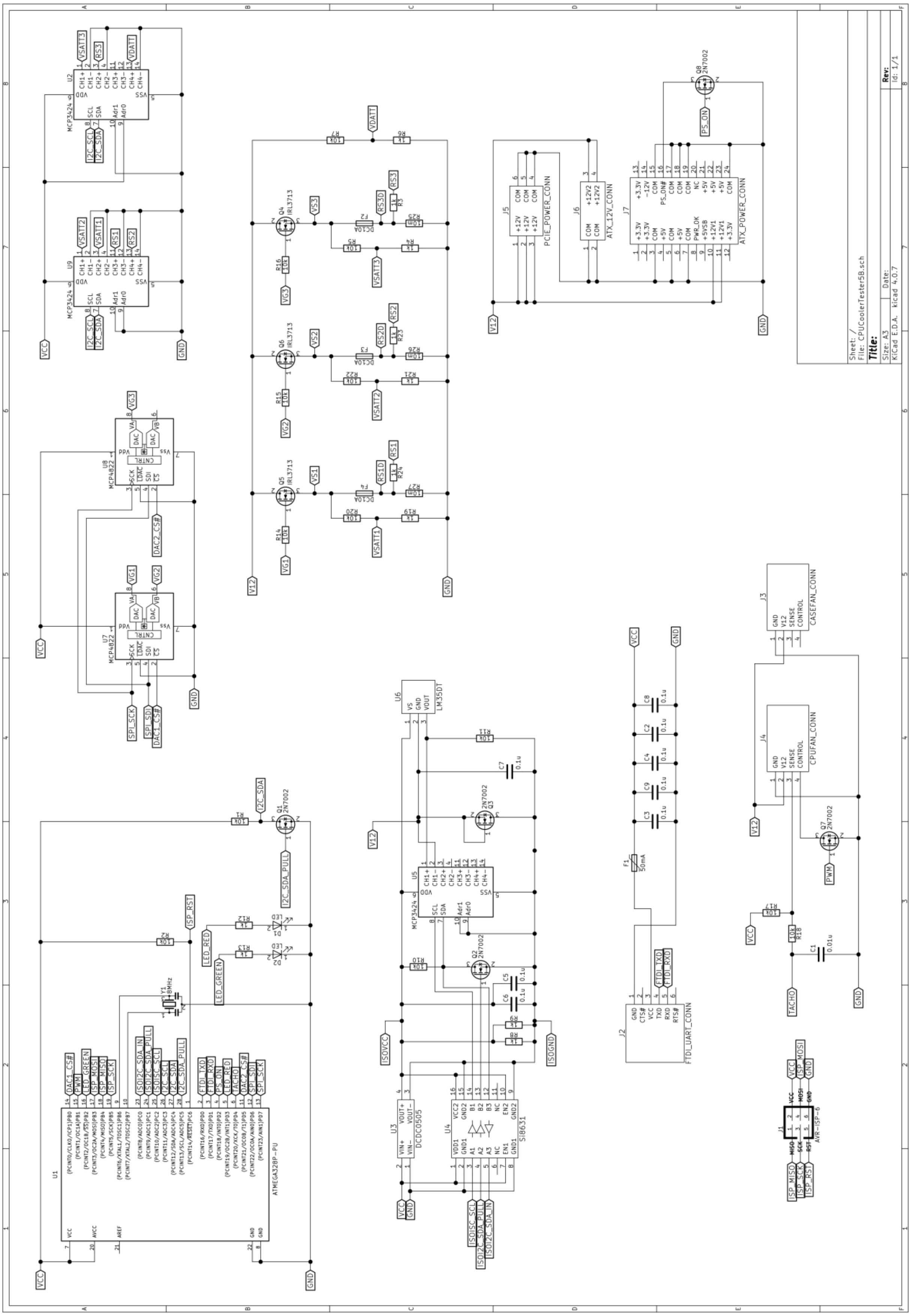
7.5. C# wrapper library

The wrapper library in C# language is provided. Please check https://cpucoolertester.com/download.

7.6. Python sample script

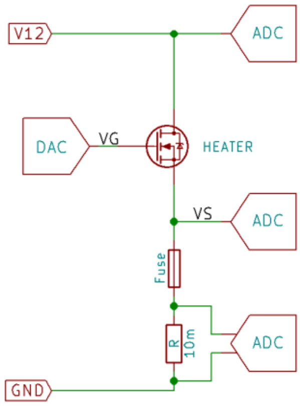
A sample script in the Python language to control the board is provided. Please check https://cpucoolertester.com/download.

8. Schematic



9. Method of controlling the power dissipation of heater MOSFET

This section describes the feedback control method employed to control the power dissipation of heater MOSFET. The figure below is a simplified schematic of the heater circuit.



The heater MOSFET is connected to 12 V rail. Heater current I_{DS} is controlled by gate voltage from DAC.
The heater MOSFET is operating in the saturation region. Therefore, the heater current has a positive temperature coefficient and is thermally unstable. To keep power dissipation stable, feedback control is necessary.
From voltages and currents measured by ADCs, the power dissipation of heater MOSFET can be calculated as follows.

$$P = (V_{12} - V_{\mathrm{S}}) I_{\mathrm{DS}}$$

The transfer characteristic of MOSFET in saturation region can be modeled as a quadratic function.

$$I_{\mathrm{DS}} = \beta (V_{\mathrm{GS}} - V_{\mathrm{T}})^2 \text{ for } V_{\mathrm{GS}} - V_{\mathrm{T}} > 0$$

where β and V_{T} are model constant.
Based on this model, the derivative of P can be approximated with respect to DAC voltage (V_{G}) at a given operating point.

$$\frac{dP}{dV_{\mathrm{G}}} \sim \frac{12}{\frac{1}{2\sqrt{\beta I_{\mathrm{DS}}}} + R}$$

where constant R is the sum of resistance of the fuse and the current sense resistor, and the constant 12 is the typical voltage of 12 V rail.
Using this value, the microcontroller performs a Newton-Raphson loop for the heater power dissipation. To stabilize the loop, constant β and R are chosen to overestimate $\frac{dP}{dV_{\mathrm{G}}}$. Moreover, to stabilize the loop when I_{DS} is rising, the expression $\frac{dP}{dV_{\mathrm{G}}}$ is rewritten to use $\mathrm{Max}(I_{\mathrm{DS}}, I_{\mathrm{T}})$ instead of I_{DS}, where I_{T} is the estimated target current given by $I_{\mathrm{T}} = \frac{P_{\mathrm{T}}}{12}$, where P_{T} is targeting power dissipation.

$$\frac{dP}{dV_{\mathrm{G}}} \sim \frac{12}{\frac{1}{2\sqrt{\beta \mathrm{Max}(I_{\mathrm{DS}}, I_{\mathrm{T}})}} + R}$$

In the Newton-Raphson loop, the updated DAC code is calculated as follows:

$$D_{n+1} = D_n - \mathrm{Round}\left(\frac{P - P_{\mathrm{T}}}{\frac{dP}{dV_{\mathrm{G}}} V_{\mathrm{LSB}}}\right)$$

where D_n is the current DAC code, D_{n+1} is the updated DAC code, Round is the rounding function and V_{LSB} is the LSB voltage of DAC.

Since rounding is used and the model constants are chosen to overestimate $\frac{dP}{dV_{\mathrm{G}}}$, D_{n+1} can still be a non-optimal value when D_{n+1} equals to D_n. To overcome this, when D_{n+1} equals to D_n for all three heater channels in the update cycle, the controller enters the "fine tuning mode". In this mode, the controller checks whether the total power is sufficiently close to the targeting total power. If so, the controller leaves the DAC codes unchanged. Otherwise, it chooses the heater channel farthest from the optimal power and incrementally increases (or decreases) the DAC code of that channel.

To stabilize the loop and avoid abrupt changes in the heater MOSFET junction temperature, the targeting total power in the update cycle is limited within the current total power plus 10 watts.