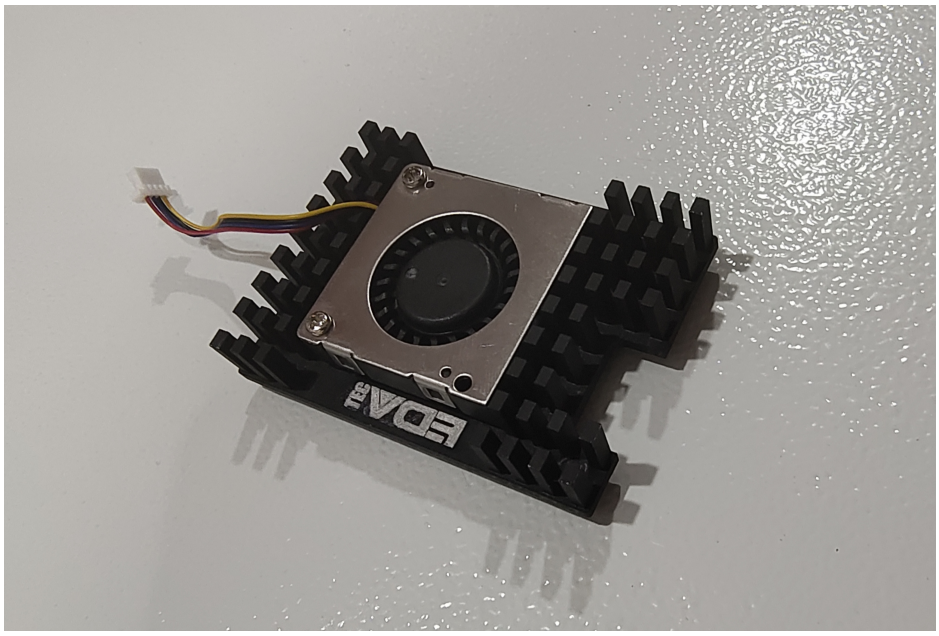
ED-CM5ACOOLER Cooling Performance Testing

1 Overview

This chapter introduces the Test device, Test purpose, Required Test Peripherals, and Test Program.

1.1 Product Overview

The ED-CM5ACOOLER is based on the Raspberry Pi CM5 active heatsink with an integrated blower, which is designed to accelerate heat dissipation and effectively reduce the operating temperature of the Raspberry Pi CM5 through the combination of a radial blower and heat sink.



1.2 Test Purpose

Test the cooling performance of ED-CM5ACOOLER.

1.3 Test Overview

By reading the temperature and frequency of Raspberry Pi CM5 CPU, the cooling performance of ED-CM5ACOOLER can be judged.

At the same ambient temperature, the lower temperature of Raspberry Pi CM5 CPU and the higher the frequency indicates the better cooling performance.

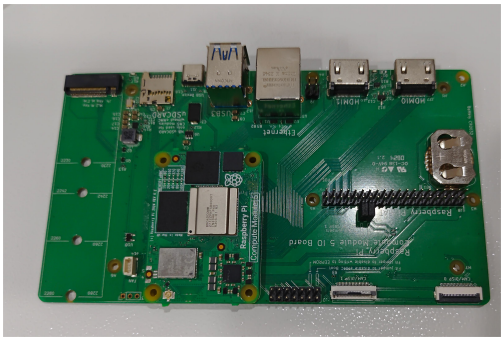
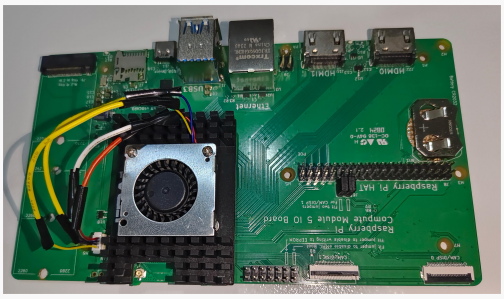
Test Content	Data Source
Cooling test of ED-CM5ACOOLER on Raspberry Pi CM5	The temperature of Raspberry Pi CM5 CPU

2 Cooling Performance Testing

2.1 Device Under Test

2.1.1 Hardware Configuration

The following two groups of equipment were configured to test and compare the cooling effect of ED-CM5ACOOLER.

Group	Configuration	
A	Raspberry Pi CM5 + Raspberry Pi CM5 IO Board	
B	Raspberry Pi CM5 + Raspberry Pi CM5 IO Board + **ED-CM5ACOOLER**	

2.1.2 Software Configuration

Operation System： 2024-07-04-raspios-bookworm-arm64.img

2.2 Test Equipment and Environment

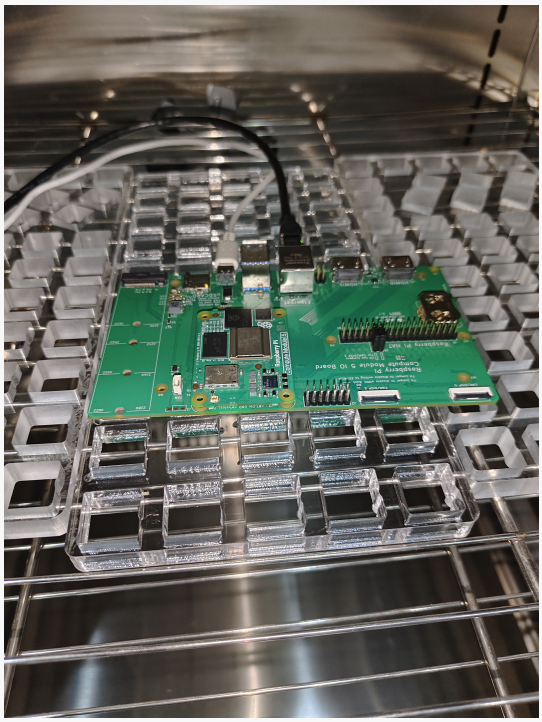
2.2.1 Test Equipment

Test Equipment	Quantity
Raspberry Pi CM5	1
Raspberry Pi 27W USB-C Power Supply	1
ED-CM5ACOOLER	1
Raspberry Pi CM5 IO Board	1
Thermostat	1
Network Cable	1

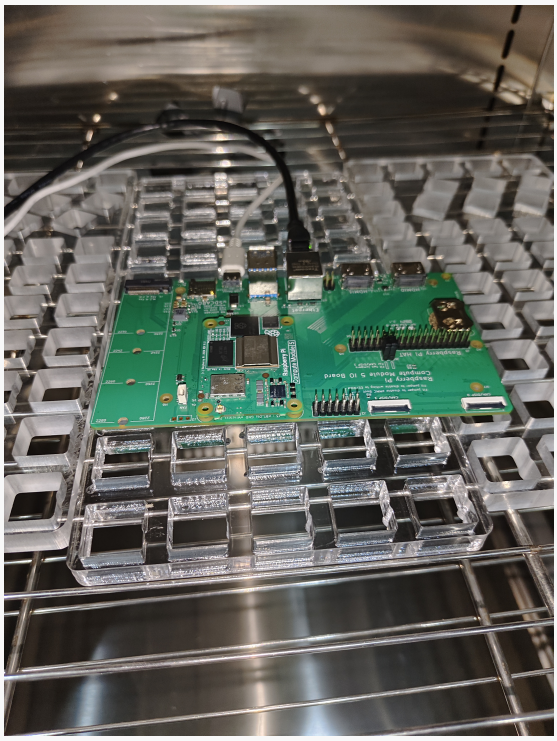
2.2.2 Test Environment

Temperature: Constant temperature (25℃,60℃)

Humidity: 20%



![



2.2.3 Test Software

This script is used to make the Raspberry Pi CM5 CPU 4 core run at full load, record the temperature data of Raspberry Pi CM5 CPU every 5s, save it in a ".csv" file and print it to the terminal.

```
#!/bin/bash
```

```
#

PID_BENCH=

FILE=./temp$1-bench.csv

[ "$(whoami)" == "root" ] || { echo "Must be run as sudo!"; exit 1; }

if [ ! `which sysbench` ]; then
    apt-get update -y
    apt-get install -y sysbench
fi

trap ctrl_c INT

ctrl_c() {
    echo "** CTRL-C Detected"
    echo "** Kill Bench and Exit"
    kill $PID_BENCH >/dev/null 2>&1
    exit 0
}

bench() {
    while true ; do
        sysbench --test=cpu --cpu-max-prime=20000 --num-threads=4 run > /dev/null
2>&1
        # sleep 1
    done
}

monitor() {
    Counter=14
    DisplayHeader="Time      Temp    CPU         Throttle       Vcore"
    echo "Time,      Temp,    CPU,         Throttle,       Vcore" >> ${FILE}
    while true ; do
        let ++Counter
        if [ ${Counter} -eq 15 ]; then
            echo -e "${DisplayHeader}"
            Counter=0
        fi
        Health=$(perl -e "printf \"%19b\n\", $(vcgencmd get_throttled | cut -f2 -
d=)")
        Temp=$(vcgencmd measure_temp | cut -f2 -d=)
        Clockspeed=$(vcgencmd measure_clock arm | awk -F"=" '{printf
("%0.0f",$2/1000000); }' )
        CoreVolt=$(vcgencmd measure_volts | cut -f2 -d= | sed 's/000//')
        echo -e "$(date '+%H:%M:%S'), ${Temp}, $(printf '%4s' ${Clockspeed})MHz,
$(printf '%020u' ${Health}), ${CoreVolt}" | tee -a ${FILE}
        sleep 5
    done
}

echo "********** Raspberry Pi Benchmark **********"
echo ""
echo "           Press 'CRTL + C' to Exit         "
echo ""
```

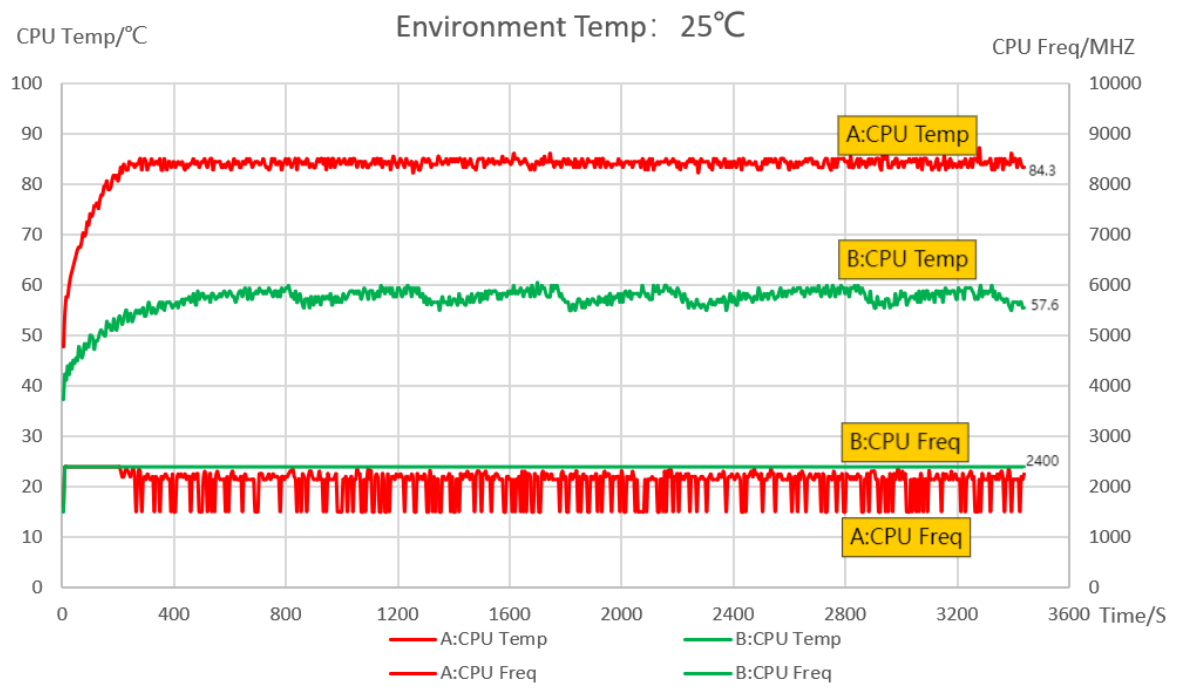
```
echo "*******************************************"
echo ""

touch ${FILE}

bench &
PID_BENCH=$!

monitor
```

2.3 Test Steps

1. Flashing the **2024-07-04-raspios-bookworm-arm64.img** image to two groups of devices, connect them with ssh, and update them using the following commands in turn `sudo apt updae` and `sudo apt upgrade` commands.
2. Configure devices in groups A and B to test each other. Each test lasts one hour.
3. Run the script in the test software to set the Raspberry Pi CM5 CPU to run at full load with 4 cores and read the CPU temperature and frequency data.
4. Evaluate the cooling effect of ED-CM5ACOOLER by comparing the temperature and frequency data of the Raspberry Pi CM5 CPU of the two groups of devices; the lower the temperature of the Raspberry Pi CM5 CPU and the higher the frequency of the device when it is running stably under the same ambient temperature, the better cooling effect will be.
5. The final test from a constant temperature of 25 ℃ conditions gradually in the thermostat increased by 5 ℃, read the CPU temperature and frequency data, test out the critical value of the group B device does not downclock the environment temperature.

2.4 Test Result and Analysis

Environment temperature 25°C

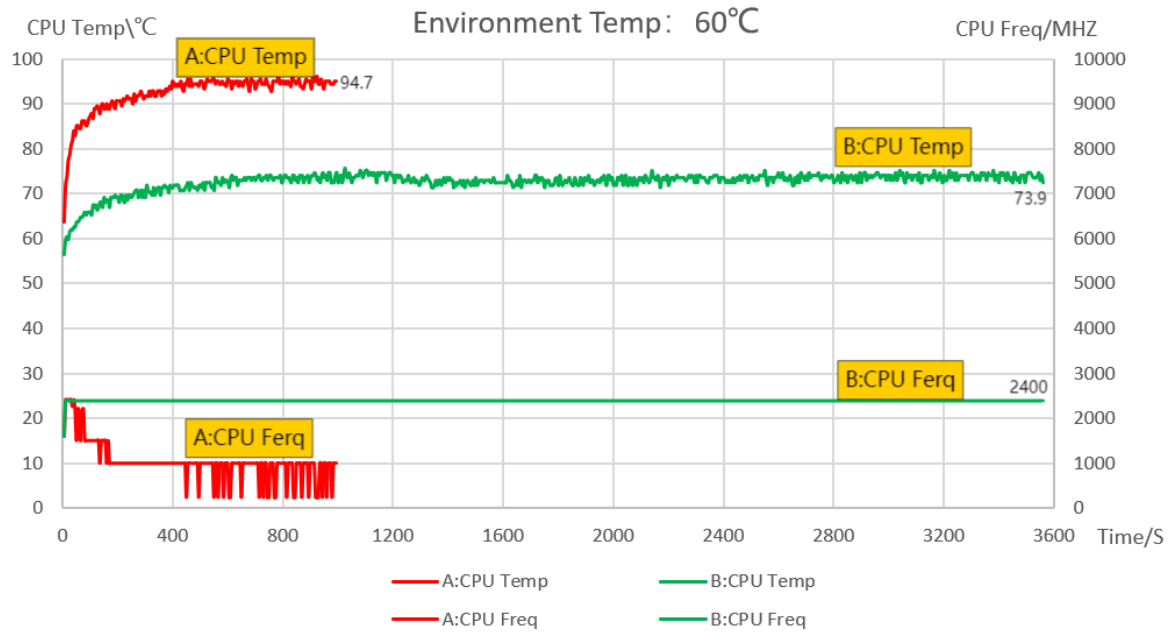


1. The following table shows the average temperature of the Raspberry Pi CM5 CPU during stable operation of the two groups of devices

Group	Configuration	Stable running temperature of CPU (℃)
A	Raspberry Pi CM5 + Raspberry Pi CM5 IO Board	84.3
B	Raspberry Pi CM5 + Raspberry Pi CM5 IO Board + **ED-CM5ACOOLER**	57.6

2. When the device is running at a steady state in a 25°C environment, the **ED-CM5ACOOLER** can reduce the temperature of the Raspberry Pi CM5 CPU by approximately 27°C, allowing the Raspberry Pi CM5 CPU to run continuously at its maximum mains frequency (2400MHZ).

Environment temperature 60°C



1. The following table shows the average temperature of the Raspberry Pi CM5 CPU during stable operation of the two groups of devices

Group	Configuration	Stable running temperature of CPU (℃)
A	Raspberry Pi CM5 + Raspberry Pi CM5 IO Board	Test termination
B	Raspberry Pi CM5 + Raspberry Pi CM5 IO Board + **ED-CM5ACOOLER**	73.9

Tip: The test was manually terminated due to high CPU temperatures while Group A devices were running.

2. At the environment temperature of 60 ° C, the Group B device runs for one hour, and the Raspberry Pi CM5 continues to operate at the maximum main frequency (2400MHZ) without downscaling.