PERFORMANCE EVALUATION OF STARFIVE VISIONFIVE V1 BOARD

BY

KAM KAI HENG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Performance Evaluation of Starfive VisionFive V1 Board

Academic Session: January 2023

I KAM KAI HENG

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
No 4, Jalan Sejahtera 5, Taman
Nusa Damai, 81700 Pasir Gudang
Johor

Ts. Wong Chee Siang
Supervisor's name

Date: 23 April 2023

Date: 28 April 2023

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ____23 April 2023___

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that ______***Kam Kai Heng***_____ (ID No: __***19ACB03056***__) has completed this final year project entitled "*PERFORMANCE EVALUATION OF STARFIVE VISIONFIVE V1 BOARD*" under the supervision of ***Ts. Wong Chee Siang*** (Supervisor) from the Department of ___Computer Science__, Faculty of ___Information and Communication Technology__.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Kam Kai Heng*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**PERFORMANCE EVALUATION OF STARFIVE VISIONFIVE V1 BOARD**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ________________________

Name : _______Kam Kai Heng_____

Date : ______23 April 2023______

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts. Wong Chee Siang who has given me this bright opportunity to engage in single board computer evaluation project. It is my first step to establish a career in Single Board Computer field. A million thanks to you.

I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

The concept of a single board computer, as the name implies, involves a functional computer built on a motherboard with a CPU, RAM, and I/O devices, similar to a regular computer. Single board computers are becoming increasingly popular for Internet of Things (IoT) applications due to their affordability, energy efficiency, and ease of portability. One emerging instruction set architecture (ISA) is RISC-V, which is an open-source chip architecture designed to optimize power and performance for modern workloads such as IoT, artificial intelligence, and machine learning. This research aims to explore the performance advantages offered by the RISC-V ISA by benchmarking it against popular single board computers such as the Raspberry Pi (RPi), which is based on the ARM ISA. However, as RISC-V is a relatively new architecture, there are challenges to overcome in terms of compiler optimization, operating system and library support, and debugging tools. The main outcome of this research is to provide insights to developers regarding the performance advantages and benefits offered by the RISC-V ISA.

The success of this project will contribute to increased awareness and trust in the VisionFive V1 Board among the community. Developers can leverage the performance advantages of this board for their applications, ultimately benefiting the broader community.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IoT	Internet of Things
ISA	Instructions Set Architectures
RPi	Raspberry Pi
GCC	GNU Compiler Collection
PTS	Phoronix Test Suite
RAM	Random-Access Memory
IP	Intellectual Property
RISC	Reduced Instruction Set Computing
CISC	Complex Instruction Set Computing
CPU	Central Processing Unit
DNN	Deep Neural Network
CV	Computer Vision
HPL	High-Performance Linpack

CHAPTER 1 - Introduction

As technology advances, the use of single board computers becomes more prevalent. These computers, built with a microprocessor, memory, and an operating system on a single motherboard, are an essential component of the Internet of Things (IoT) field due to their compact size and nearly identical computing capabilities to traditional computers. There are several different instructions set architectures (ISA) used in single board computers, including ARM, x86, and RISC-V. While ARM and x86 are familiar ISAs used in single board computers for nearly 40 years, until recently, no developer had successfully created a RISC-V based single board computer [1]. The VisionFive V1 Board, the first RISC-V based single board computer, has been invented but its performance remains undefined. As a result, performance benchmarking is necessary to evaluate the board's performance for future industrial use.

To obtain the performance results for the VisionFive V1 Board, three primary tests were conducted in this project: Phoronix Test Suite Benchmarking, Thread Performance Test, and Power Consumption test. The same tests were also run on the Raspberry Pi (RPi) board, allowing for a clear comparison of performance between the two boards as a reference for other developers.

1.1 Problem Statement and Motivation

As the RISC-V architecture-based single board computer is still a relatively new invention, developers may encounter challenges when using it to develop products. Firstly, they may not know which applications are suitable for use with the VisionFive V1 board. Without any previous performance evaluation references, developers may find it challenging to develop products. Secondly, they may not fully understand the differences between RISC-V and ARM-based architecture single board computers. This lack of understanding may cause them to focus more on cost instead of performance when selecting a single board computer. While the specifications of the VisionFive V1 board are similar to those of the RPi 4B, developers may not know when to use the VisionFive V1 board over the RPi 4B board. As there are no proper benchmarking references available, it is difficult for developers to estimate the resources required to develop products using the VisionFive V1 board. Additionally, developers may be concerned about the board's security risks, which could impact the safety of their end product.

The aim of this project is to evaluate the performance of the VisionFive V1 board in terms of system, processor, and low-level hardware design. By doing so, we hope to increase awareness of this product in the industry and dispel any myths about RISC-V architecture-based single board computers. As the VisionFive V1 board was only released in December 2021, many people are still unfamiliar with it and cannot estimate its performance capabilities. This project will provide valuable insights into the architecture and help developers better understand the differences between ARM and x86 architecture.

Furthermore, this project will provide an alternative to the RPi product and may encourage organizations to replace it with the VisionFive V1 board if its performance is encouraging. As an open-source product, the VisionFive V1 board provides developers with greater freedom and power to develop products, and this project will help push developers to the next level of creativity.

1.2 Project Objectives

The primary aim of this project is to conduct a performance benchmarking analysis on the newly developed single board computer, the VisionFive V1 Board, which is based on the RISC-V architecture. Although there are numerous unique single board computers available in the market, this is the first RISC-V-based single board computer, making it critical to evaluate its performance before it can be widely used in the industry.

The objectives of proposing this project are:

1) To provide developers with a performance reference point when selecting a suitable single board computer for their project.
2) To compare and contrast the VisionFive V1 Board's performance with the current market leader, the Raspberry Pi model.
3) To contribute to the documentation and knowledge base in the field of single board computers.

1.3 Project Scope and Project Direction

The proposed project aims to evaluate the performance of the VisionFive V1 Board from Shanghai StarFive Company. The evaluation will cover low-level hardware component benchmarking, as well as benchmarking of the board's processor and system. Additionally, the project seeks to uncover the unique features of the VisionFive V1 board in the IoT field, where it has not been widely utilized. The characteristics of RISC-V architecture on the processor will also be analyzed and compared to modern architectures such as ARM and x86 processors from Intel or AMD. Upon completing the performance benchmarking of the VisionFive V1 board, a comparison between it and the Raspberry Pi model will be conducted to provide organizations with more options to choose the desired single board computer for developing their IoT products. The project will produce a performance score for the VisionFive V1 board that will serve as a benchmark reference for developers. The project will utilize several benchmarking tests, including CPU (single core) performance test, CPU (multi-core) performance test, RAM speed test, Flash Memory speed test, Graphics test, and Machine Learning Test.

1.4 Impact, Significance and Contributions

The completion of this project will have significant benefits for the community and developers. The performance score of the VisionFive V1 board will be determined, allowing developers to make informed decisions on which single board computer to use for their industry 4.0 related products. Additionally, developers will gain an understanding of how the performance of the VisionFive V1 board compares to other single board computers on the market. This will enable them to determine which product is best suited for their specific project needs.

The open-source nature of the VisionFive V1 board will further enhance its potential, as developers can collaborate and share ideas within the community to improve the product and its architecture. The availability of more options for single board computers will also benefit organizations, as they will have a greater choice of products to choose from when developing their products. Ultimately, the success of this project will contribute to the advancement of the industry 4.0 field and promote innovation among developers.

1.5 Background Information

As background information, RISC-V is the main ISA used in the VisionFive V1 board. An ISA is a model for computer architecture that defines things such as the register model and machine code instructions [2]. The two other commonly used ISAs in computers are ARM and x86, which have been widely used for generations. What sets RISC-V apart from these ISAs is its open-source property, which allows its use without any licensing fees. This is unlike ARM and x86, which have their own commercial IP licenses that developers must obtain to use their ISAs. RISC-V also includes different levels of instruction sets, including 32-bit and 64-bit variants [3]. This property allows RISC-V to be used in a wide range of applications, from small embedded systems to supercomputers with complicated processors. Additionally, RISC-V is supported by various language compilers, and one of them is the GNU Compiler Collection (GCC), which is a free software compiler [4]. GCC will be used in this project to optimize benchmarking tests with different flags, enabling accurate results for the VisionFive V1 board. Furthermore, designers can freely extend any cores and chips designed using RISC-V ISA, as there are no intellectual property restrictions as long as the designer does not register the system with other closed ISA products. This open ISA option from RISC-V provides designers with good flexibility when creating a custom processor.

The Phoronix Test Suite is one of the platforms that will be used to perform benchmarking on the VisionFive V1 board in this project [5]. It is an open-source framework for performing automated performance benchmarking tests, including uploading of test results, detection of tested system hardware or software, and other essential features. Most importantly, there are plenty of tests provided by different developers in the Phoronix Test Suite, most of which are available for free. As an example, FLAC Audio Encoding can be used to examine the performance of a single core in the CPU. Every user in the Phoronix Test Suite has the option to choose whether to upload their results onto OpenBenchmarking.org. All results uploaded onto this website can be viewed by other users and compared with each other.

In addition, the project will conduct a benchmarking test to assess thread scalability, measuring how the performance changes as the number of working threads increases or decreases. Additionally, a power meter will be used to record power consumption during the test, allowing for an evaluation of the efficiency of both boards.

1.6 Report Organization

The details of the project will be included in the next few chapters. Chapter 2 will discuss about the literature review on the research paper or the existing benchmarking technique to identify and analyse the strength, weakness of each of the reviewed system. In chapter 3, the exploratory research method flow for this project will be explained together with finalized of general workflow for this project. For chapter 4, the way to setup the experiment which is hardware and software required in this project will be discussed, the designation of the system will also be provided in this chapter. In chapter 5, the benchmarking result and performance comparison analysis of VisionFive V1 and RPi 4B model will be done. Lastly, conclusion of the project will be presented in chapter 6.

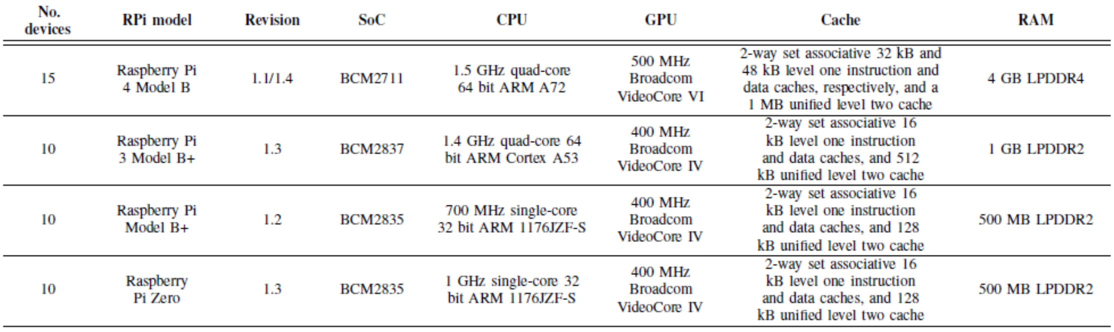
CHAPTER 2 - Literature Review

2.1 Previous Works on Single Board Computer

In this chapter, I will review some works that is done by previous researcher which is technique of benchmarking single board computer and characteristics of different ISA to expand my knowledge in this field. Finally, the specification of VisionFive V1 and RPi 4B model will be compared to have a clearer view of differences between them. Any benchmark result of RPi 4B model will be more focused in this section as the specification of RPi 4B is similar to VisionFive V1 Board.

2.1.1 LwHBench: A Low-level Hardware Component Benchmark and Dataset for Single Board Computers

A low-level hardware component benchmarking application which called LwHBench is introduced in [6]. This application is used to measure CPU, GPU, memory, and storage device performance test for Single Board Computer that will be applied in IoT. In this paper, different model of RPi product has been used to involve in benchmarking process. The model's specification is shown as figure below.



No. devices	RPi model	Revision	SoC	CPU	GPU	Cache	RAM
15	Raspberry Pi 4 Model B	1.1/1.4	BCM2711	1.5 GHz quad-core 64 bit ARM A72	500 MHz Broadcom VideoCore VI	2-way set associative 32 kB and 48 kB level one instruction and data caches, respectively, and a 1 MB unified level two cache	4 GB LPDDR4
10	Raspberry Pi 3 Model B+	1.3	BCM2837	1.4 GHz quad-core 64 bit ARM Cortex A53	400 MHz Broadcom VideoCore IV	2-way set associative 16 kB level one instruction and data caches, and 512 kB unified level two cache	1 GB LPDDR2
10	Raspberry Pi Model B+	1.2	BCM2835	700 MHz single-core 32 bit ARM 1176JZF-S	400 MHz Broadcom VideoCore IV	2-way set associative 16 kB level one instruction and data caches, and 128 kB unified level two cache	500 MB LPDDR2
10	Raspberry Pi Zero	1.3	BCM2835	1 GHz single-core 32 bit ARM 1176JZF-S	400 MHz Broadcom VideoCore IV	2-way set associative 16 kB level one instruction and data caches, and 128 kB unified level two cache	500 MB LPDDR2

Figure 2.1.1.1: Model and most relevant characteristics of the devices used for validation.

The author in this paper had also mentioned that data collection program is implemented using Python 3 so that it can be used to run on any device with the required libraries installed. 100 read and write operation of 100kB of data will be monitored in the SD card in which generating 200 features in total. Other than that, several implementations need to be done to prevent any unfair result occurred during the benchmarking process. The implementations included are fixed CPU/GPU/RAM frequency, kernel level priority, disable memory address space layout randomization, profiled guided optimization, fixed hash seed and core isolation. In this implementation, the numbers of sample generated from each device will be monitored and compared between each other. However, we will only pay more attention on result generated by RPi 4 model for this project as it's specification is quite similar with VisionFive V1 Board.

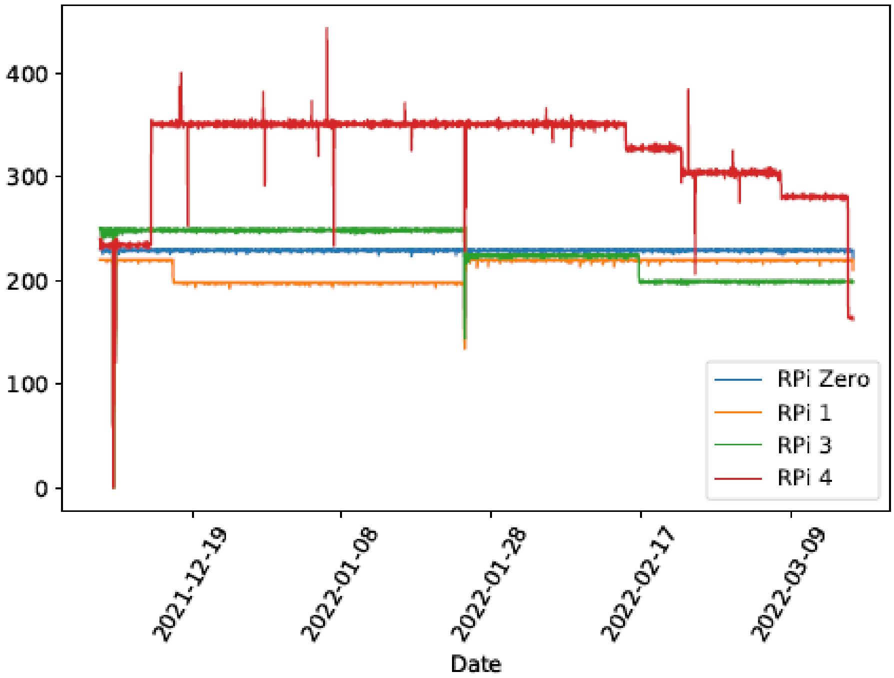


Figure 2.1.1.2: Sample generated by each model every hour.

There are two use cases that we have to consider which is the benchmark and dataset can only be applied in device identification based on the performance of its device. The second use case that we need to consider is the benchmark and dataset can be applied in the comparison of each device component. For first use case, it can be seen from two different perspectives which are Model Identification and Individual Identification. For model identification, machine learning and deep learning based dimensionality will be group together from each device. This method is chosen because of its proven accuracy for automated class inference and identification in many research areas. It can also be used to determine the stability of the type of device. As a result, there is 784095 samples generated by RPi 4 model in Cluster 0. For individual identification, this is used to uniquely identify each one of the 45 devices used for dataset generation. This is a perspective that more focusing on machine learning or deep learning algorithm to produce the device identification results. It uses different algorithms such as, k-Nearest Neighbors, Support Vector Machine and many else to compare between each other. For the second use case, it can also be seen from 2 different areas which are intra-device comparison and inter-device comparison. For intra-device performance analysis, the impact of temperature from each model will be taken and recorded. As we can see from figure 2.1.1.3, the temperature correlation for RPi 3 model is in high sensitivity for almost all the feature tested in the implementation. While for the other model including RPi 4 shows a stable performance for all the feature tested.

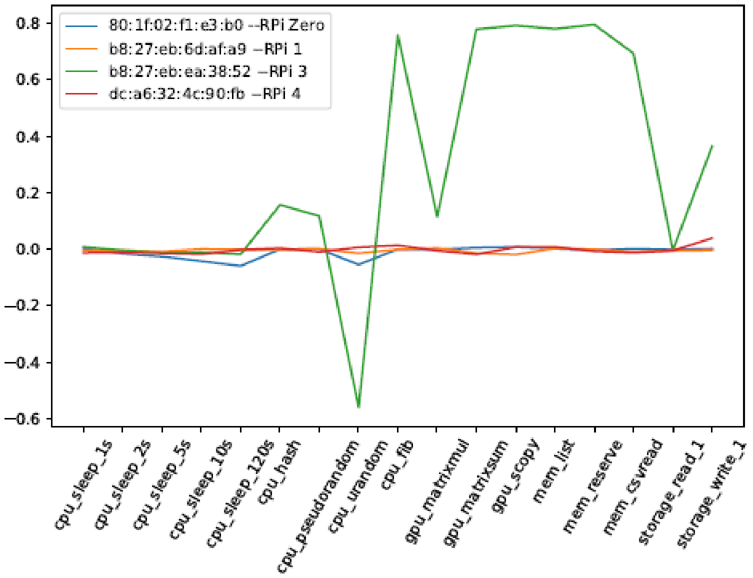


Figure 2.1.1.3: Temperature correlation plot.

Lastly, inter-device performance analysis is the present of the performance to analyse the result of different hardware components of the same model. Although the model taken is the same, but the result of each individual device can still be different. This can be proof by using the distribution densities of different features for devices as shown in figure 2.1.1.4.

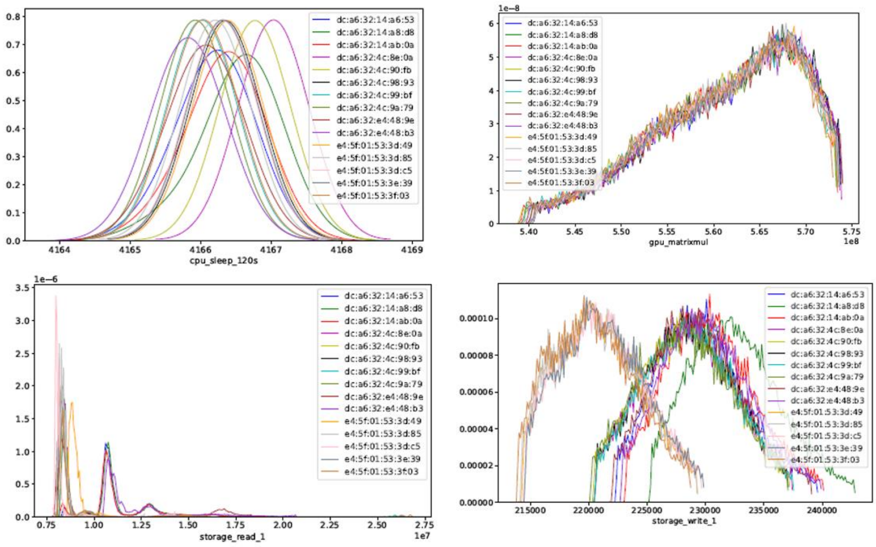


Figure 2.1.1.4: Raspberry Pi 4 feature density plots.

Therefore, we can conclude that even though the hardware specification is the same for each model, but the performance variation is still depending on the exact device. The chips in the device may be consist of variations that can be leveraged to perform fingerprinting or identification tasks.

2.1.2 Performance Evaluation of Different Raspberry Pi Models for a Broad Spectrum of Interests

[7] had conducted a performance evaluation of different RPi models in several scenarios by using different parameters. Areas that had been tested are including thermal test, CPU performance test, memory speed test, sequential access performance test, sequential write access performance on microSD Card test, audio conversion performance test, other performance test, TCP throughput test and TCP latency test. For the performance evaluation conducted, there are total of 5 RPi model are involved. Only RPi 4B is connected with a fan to reduce its temperature when it starts to operate. RPi operating system (32-bit version) had chosen as the main operating system for the model tested. The first test that was carried out which is temperature test by using Stressberry Test. It is common that processor for a single board computer will faced over-heating problem and if necessary, CPU throttling will occur in order to decrease the electrical energy being consumed so that the heat generated can be decreased as well. The Stressberry test mentioned above is used to determine whether the CPU can run as its fully performance without facing the overheating problem. The result for RPi 4B is shown as figure below:

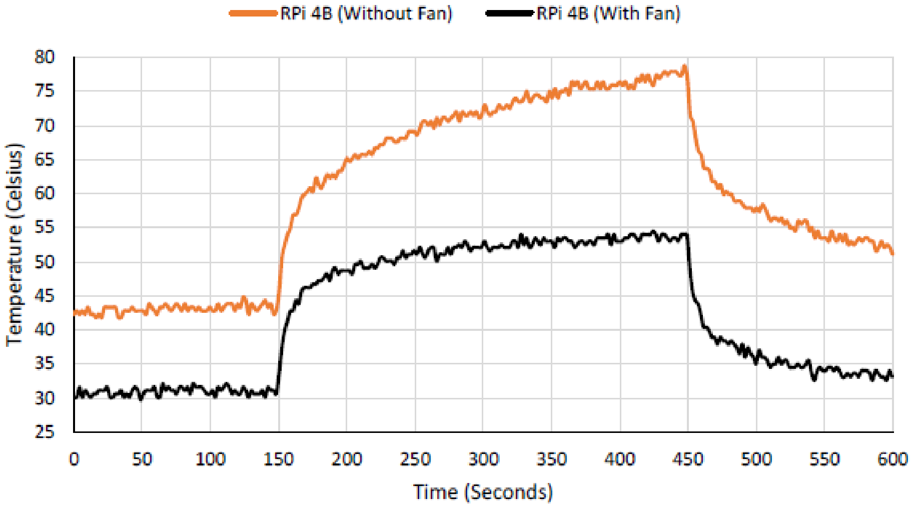


Figure 2.1.2.1: Result of Stressberry for Raspberry Pi 4B with/without fan.

The next test that had been carried out is CPU performance test by using 7-zip and Sysbench. 7-zip is a tool that packing and compressing files into archives, and it has a build-in benchmarking tool inside it. The tool will display the result of how well the CPU can perform compression and decompression on a dummy data.

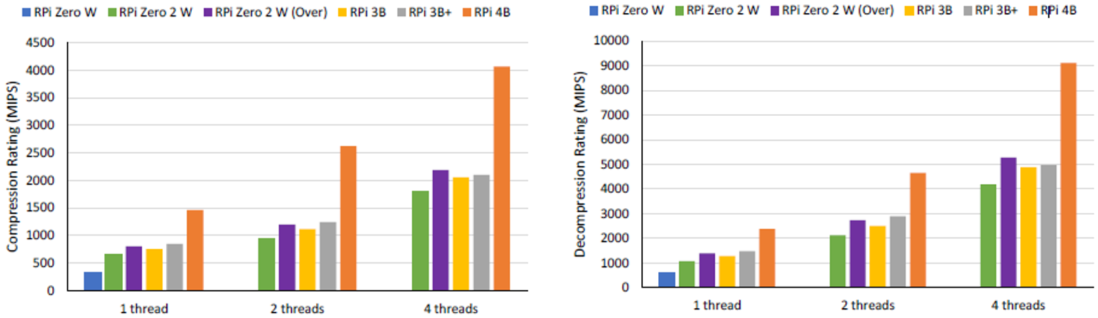


Figure 2.1.2.2: Result of Compression and Decompression Rating by using 7-zip.

While for Sysbench, it is a multi-threaded benchmark tool that can also be run in Linux system and it consist of multiple tests which are CPU performance, memory speed test, file input/output speed, threads performance and mutex performance. For the test conducted using Sysbench in this paper, the researchers are more focusing on CPU performance itself as the memory test conducted on Sysbench does not acquire a consistent result. The result of CPU performance is shown as figure below:

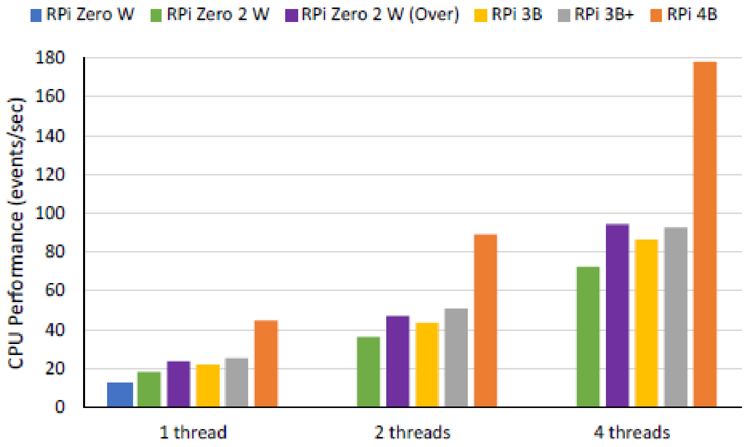


Figure 2.1.2.3: Result of CPU Performance by using Sysbench

Move to next test, memory speed test will be carried out by using Geekbench and STREAM. Geekbench is a cross-platform benchmark program that allow user to report the performance related to memory. There are 4 tests provided in Geekbench which are Read Sequential, Write Sequential, Stdlib Write, and Stdlib Copy and each of the model had undergo the tests. The result is shown as figure below:

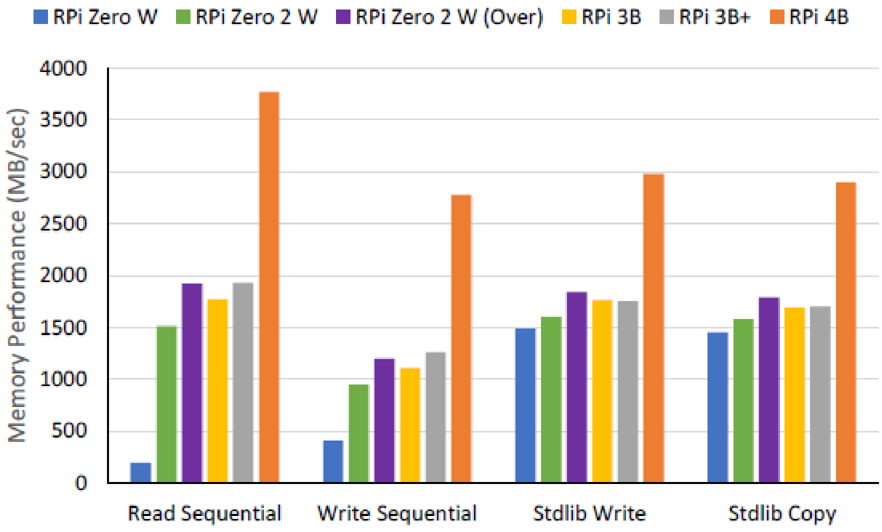


Figure 2.1.2.4: Result of Memory Performance by using Geekbench

While for STREAM, it is a benchmark developed to test the performance of sustainable memory bandwidth and computation rate for copy, scale, add and triad kernels. We can clearly see from the figure below that RPi 4B had outperformed other devices since it is the newest devices invented by RPi foundation.

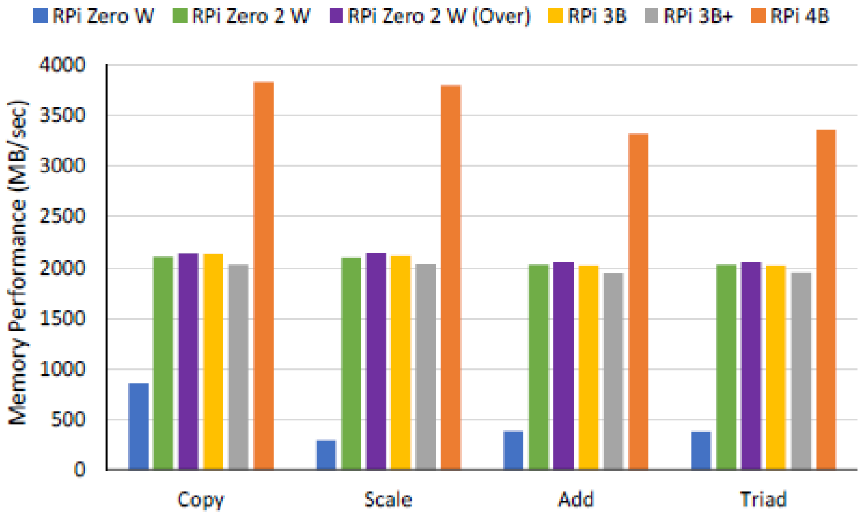


Figure 2.1.2.5: Result of Memory Performance by using STREAM

There is also lot of some other tests had been carried out in this paper and the result produced by RPi 4B undoubtedly had stand out among the RPi devices in this paper.

2.1.3 Single Board Computer Benchmarks

From the website [8], the test for single board computer benchmarks is done within Phoronix Test Suite (PTS). It is a free and open-source benchmarking platform for multiple operating systems and Linux. The user is also allowed to upload the result obtained from PTS to the Open Benchmarking website either privately or publicly. There are a few tests that had been carried out in this website, which are single core CPU test, multi-core CPU test, random-access memory (RAM) speed test, flash memory test, 2D graphics test, and power test. Every performance evaluation for single board computer is carried out by using PTS excluding power test as it is not included in PTS package.

For single core CPU test, it is a test that uses CPU to encode a WAV file into FLAC format. The encoder will only use up one thread, which means that the comparison will be done between single core speeds of different model.

For multi-core CPU test, Himeno benchmark will be used to examine the speed of multi-core processor. It is better for the CPU to run multiple threads simultaneously as it can speed up the time to get the job done for a computer.

Move to RAM speed test, we need to examine the speed of a computer to read and write data to RAM. RAM is used to store any short-term memory in a computer and how fast a computer can access to the data is indicating how good the performance it is. RAMspeed benchmark is used to carry out the test by copying bunch of integers to and from RAM.

The next test that is being carried out is flash memory test. Like I mentioned above, RAM is good for storing the short-term memory while hard disk drive allow computer to store the non-volatile data which means that the user will not lose the data even the power had been cut off. In most of the case, a single board computer does not really uses hard disk drive but they rely on SD cards which serve the same function as hard disk drive and we called it as flash memory. IOzone test from PTS is used to test the read and write non-volatile data speed to and from the flash memory.

2D graphic test will be carried out next. Some of the single board computer had even build-in with a low-end gaming machine as the community becoming more modern day by day. GtkPerf is used to run the graphic test on single board computer. We can indicate the graphic performance by examining how long the graphics card takes to finish drawing a set number of circles in a window.

Lastly, power test to test out the power consumption of each single board computer will be carried out. This is a test cannot be done using PTS platform. Therefore, the author creates a current logging device to compute idle and loaded power consumption.

2.1.4 RISC-V vs. ARM vs. x86 – What’s the difference?

In this website [9], the author had introduced about ARM, RISC-V and x86 from Intel. The first difference between them is ARM, x86 are actually based on its proprietary intellectual property (IP). However, RISC-V offers an open-source IP license to the public but there is also commercially licensed cores for RISC-V. Other differences for them are RISC-V and ARM processor are designed based on reduced instruction set computing (RISC) while x86 processors employ a complex instruction set computing (CISC). The figure below will show the licensing models for each architecture model.

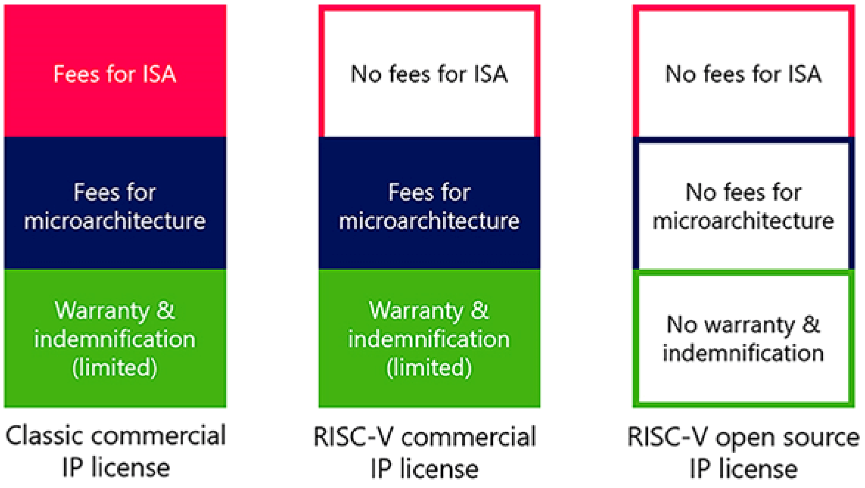

Fees for ISA
Fees for microarchitecture
Warranty & indemnification (limited)
Classic commercial IP license
No fees for ISA
Fees for microarchitecture
Warranty & indemnification (limited)
RISC-V commercial IP license
No fees for ISA
No fees for microarchitecture
No warranty & indemnification
RISC-V open source IP license

Figure 2.1.4.1: Licensing model for each architecture.

For power and performance comparison, RISC required less power to be consumed to execute a simple instruction and CISC processor will try to minimize the number of instructions per program by increasing the complexity of each instruction. However, RISC processors use an approach that execute the instruction in a single clock cycle by increasing the number of instructions. The computer performance can be calculated by using the formula below:

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \text{ x } \frac{\text{cycles}}{\text{instruction}} \text{ x } \frac{\text{instructions}}{\text{program}}$$

Figure 2.1.4.2: Formula of computer performance.

In terms of safety of each processor, x86 architecture includes 4 levels of safety privilege, ARM architecture consists of 3 privilege levels while RISC-V uses software-defined isolation domains. Due to RISC-V open-source property, the community is large and diverse with lots of designers and developers. Therefore, they will share instantly with the experts in this field about the solution when there is any security risk identified.

For three of them, ARM, x86 and RISC-V all supports virtualization of CPUs and memory. However, only RISC-V ISA does not support the virtualization of file input/output. For all architecture except RISC-V, they have a virtualization extension that allow hardware to virtualize a CPU using a hypervisor. For RISC-V, it uses a hypervisor extension that introduce a full duplicate of CPU state in which one copy for guest and another one is for host. Move to virtualization of memory, all x86, ARM and RISC-V architectures is supported using multi-stage page tables. I/O virtualization is not done yet in RISC-V as the technique for RISC-V is not mature yet. There is only a rough idea for it in which is will be a feature of Input-Output Memory Management Unit and platform-level interrupt controller.

In conclusion, there are lot of differences between x86, ARM and RISC-V processors. Most of the feature in x86 and ARM had been discovered during the past time and dominated most of the market. However, the era had changed now. RISC-V architecture had now increasingly used in single board computer, embedded system, artificial intelligence and many else due to its open-source property, everybody in community has a chance to discover its feature.

2.1.5 VisionFive V1 Board Official Website

By referring to this website [10], I able to know more about the new single board computer invented by StarFive Company. As we all know, there is no one else implemented a single board computer using RISC-V architecture and this is world's first single board computer from StarFive Company. Based on the specialty of RISC-V architecture, VisionFive V1 board provides the developer and designer more freedom to develop and design the industry-leading solution as it is an open-source architecture and discussion can be easily done among developers and designer. The comparison of specification between VisionFive V1 Board and RPi 4B model is shown as table below:

Table 2.1.5.1: Comparison of specification between VisionFive V1 Board **[10]** and RPi 4B model **[11]**.

	VisionFive V1 Board	RPi 4B model
Processor	• RISC-V SiFive U74 Dual-Core 64-bit RV64GC ISA SoC with 2MB L2 cache @ 1.0 GHz • Vision DSP Tensilica-VP6 for computing vision @ 600MHz • NVDLA Engine (configuration 2048 MACs @ 800MHz) • Neural Network Engine (1024MACs @ 500MHz)	• Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
Operating System	• Fedora OS	• Raspbian (Raspberry Pi OS)
Memory	• 8GB LPDDR4	• 1, 2, 4 and 8GB LPDDR4 RAM options
Wireless Connectivity	• 2.4 GHz Wi-Fi (IEEE 802.11b/g/n) • Bluetooth 4.2 with BLE	• 2.4 GHz Wi-Fi (IEEE 802.11b/g/n) • Bluetooth 5.0 with BLE
Video Processing	• 2 x MIPI-CSI (up to 4K@30 fps), 1 x MIPI-DSI (up to 4K@30fps) • 1 x HDMI 2.0 (up to 1080p@60 fps display) • Video Decoder (H264/H265) up to 4K@60fps; Support Dual stream decoding for 2K@30 fps each	• VideoCore VI 3D Graphics. • Supports dual HDMI display output up to 4Kp60 • H.265 (HEVC) hardware decode (up to 4Kp60) • H.264 hardware decodes (up to 1080p60)

	• Dual channels of ISP, each channel support up to 4K@30FPS • Support MIPI-CSI TX for video output after ISP and AI processing • JPEG encoder/decoder	• 2x micro-HDMI ports supporting dual display up to 4Kp60 resolution
Dedicated Audio Processing DSP and Sub-system	• Ultra-low power Voice Activity Detector for audio bit-stream as a Voice Trigger • On-chip Audio DAC • Support DMIC and AMIC, up to 4 channels	• Supports near-CD-quality analogue audio output and composite TV-output. • Can drive 32 Ohm headphones directly.
Peripherals	• 4 x USB 3.0 ports • 40 Pin GPIO Header (28 x GPIO, I2C, I2S, SPI, UART) • Gigabit Ethernet Connector • 3.5 mm Audio jack (4-pole stereo audio output) • Micro-SD card slot for system boot and data storage • Support TRNG and OTP • Support DMAC, QSPI, and other peripherals • Reset button and Power Button	• 2x USB2 ports • 2x USB3 ports • 40 Pin GPIO Header (28 x GPIO, I2C, I2S, SPI, UART) • Gigabit Ethernet port • Raspberry Pi camera port (2-lane MIPI CSI) • Raspberry Pi display port (2-lane MIPI DSI)
Power Supply	• Minimum: 5 V / 1.5 A • Recommended: 5 V / 3 A	• Minimum: 5 V / 2.5 A • Recommended: 5 V / 3 A

There is not many differences between VisionFive V1 board and RPi 4B model in terms of specification. But there is an important section that we must pay attention to, which is the processor of both single board computer. For GPIO Header of both single board computer is also almost the same. However, there is a unique DSP chip implemented in VisionFive V1 Board which is never be seen from RPi product. Therefore, we can conclude that most of the product that is done by using RPi 4B model can also be done by using VisionFive V1 Board.

2.2 Limitation of Previous Studies

Based on the literature review that completed in section 2.1, I found out that there are some limitations in the previous studies. Firstly, I found out that there is no one else had done the benchmark comparison between RISC-V and ARM architecture. This is because the RISC-V architecture is not common yet for single board computer to be used in the past time. Other than that, most of the processor used in single board computer are ARM, x86 from Intel or AMD but there is no RISC-V architecture based of processor implemented yet for benchmarking. This may cause the researcher in future not able to accumulate the resource needed for implementing a product based on RISC-V architecture single board computer. Besides, I had also realized that most of the benchmarking process does not count operating system as a factor that will affect the benchmarking result. This might lead to an inaccuracy of result as operating system takes care of all the process including background process and those process may affect the performance of CPU.

2.3 Proposed Solutions

To solve the limitations that mentioned in section 2.2, benchmarking on VisionFive V1 Board must be done as it provides the benchmarking reference of a RISC-V architecture based single board computer to future researcher. Since this product is also the first invented RISC-V based single board computer, we can examine the difference between RISC-V and ARM architecture in terms of their performance. The comparison between benchmark of RPi 4B model and VisionFive V1 board will be done so that we can examine the effect of different operating system bring to CPU performance. In this case, Fedora Linux operating system will be used to compare with Raspbian operating system. Other than that, more people will be explored to use RISC-V architecture based single board product to invent their own product. This is not only because the architecture is still new to single board computer, but also due to its open-source property. Everybody in community is given a chance to discover their own solution and share it to others in the community.

CHAPTER 3 – System Model

I will discuss the proposed method or approach to complete the project in this chapter.

3.1 Design Specification

3.1.1 Research Framework

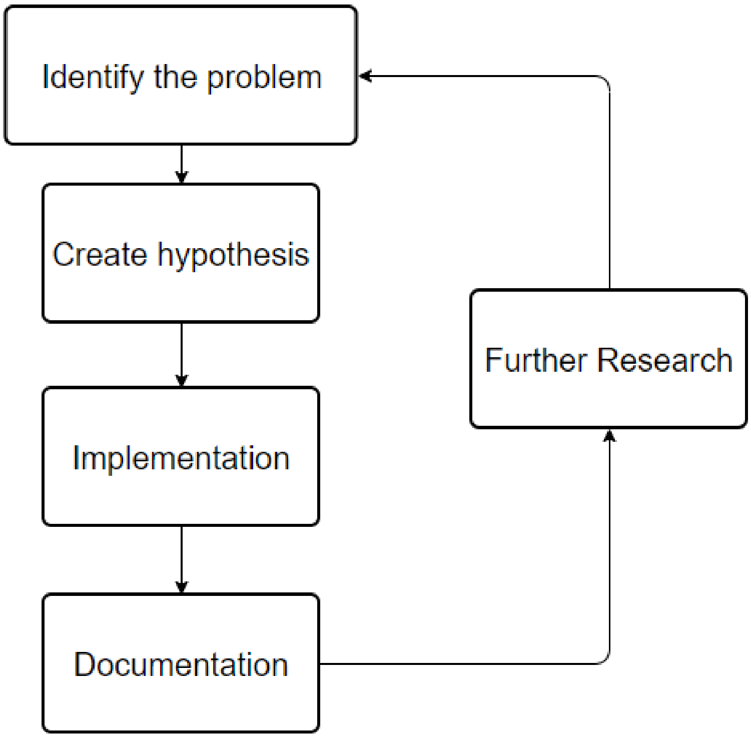


Figure 3.1.1.1: Research framework of project

The methodology that applied in this project is Exploratory Research Method. This research method is chosen for this project to investigate the performance of RISV-C architecture based single board computer in which this area had never been studied by the researchers in the past time [12]. This approach helps to identify the problem, gather information, and generate new ideas for further research.

The first step in exploratory research is to identify the problem. This involves finding a topic of interest and exploring it to see if there are any research gaps or questions that need to be answered. The problem can be identified through a variety of means, including literature review, personal observation, or experience. The different aspect of performance benchmarking need to be determined before any action is taken.

The next stage of exploratory research is to create a hypothesis. A hypothesis is an educated guess or assumption about the possible outcome of a study. It is a tentative explanation for a phenomenon that can be tested through research. The hypothesis should be based on the identified problem and be specific enough to test with data.

The third stage of exploratory research is to implement the study. This involves gathering data and analyzing it to test the hypothesis. The study can be conducted through various methods, including surveys, interviews, experiments, or observation. The data collected should be analyzed using appropriate statistical methods and techniques. The performance benchmarking will be carried out in this stage to verify the correctness of hypothesis.

The fourth stage of exploratory research is to document the findings. This involves summarizing the data collected and analyzing the results to draw conclusions. In this project, the result of each performance benchmarking test will be recorded down and listed on website which allow any public to access to it. The documentation should also include a discussion of the limitations and implications of the findings.

The final stage of exploratory research is to continue with further research. This involves using the findings of the study to generate new questions and hypotheses for further research. The process then returns to the first stage of identifying a new problem and starting the cycle again. By completing these stages, researcher should be able to gain insights and generate new knowledge that can contribute to the advancement of their field.

3.1.2 Methodologies and general work procedures

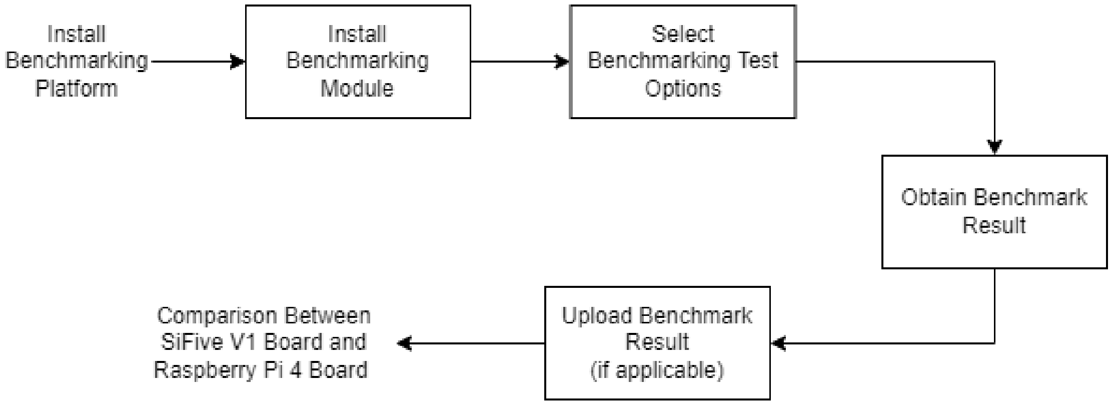

Install Benchmarking Platform
Install Benchmarking Module
Select Benchmarking Test Options
Obtain Benchmark Result
Upload Benchmark Result (if applicable)
Comparison Between SiFive V1 Board and Raspberry Pi 4 Board

Figure 3.1.2.1: General Workflow of project

As the performance of the board is undefined yet, its benchmarking performance will be recorded and compare with the current Raspberry Pi model, RPi 4B. This is because both models have the similar characteristics in terms of physical board designation (GPIO pins) and most of the organization had built trust on Raspberry Pi product. This will be a chance for

VisionFive V1 board to take place in the market or even replace the application of Raspberry Pi single board computer if and only if its performance benchmark is encouraging.

To apply the methodology, previous studies on single board computer benchmarking will be reviewed and investigate the methods proposed in the past time. After the method used had been defined, it will be used to apply it on the VisionFive V1 board and obtained the result of performance benchmarking. Then, the result between both VisionFive V1 Board and RPi 4 Board will be compared. There are lots of benchmarking references available in [13], and most of them will be used to compare with the result on VisionFive V1 board.

For the workflow of this project, benchmarking platform such as Phoronix Test Suite will first need to be installed. After the platform is installed, the following benchmarking module will also be installed. Different benchmark module provides different benchmark test for the single board computer. For example, different module will be chosen to examine the performance for CPU speed and RAM speed. Some of the modules will provide different options or parameter to complete the testing. Therefore, an appropriate option or parameter need to be chosen to ensure that the testing is ran completely. The single board computer needs time to process the test and the benchmark result can be obtained after a period. Also, the benchmark result will be uploaded onto a website to let the other user have a clearer picture on the result if the platform allows me to do so. Finally, a comparison of the benchmarking result between VisionFive V1 Board and Raspberry Pi 4 Board will be done to examine the difference between them.

CHAPTER 4 – Experiment / Simulation

This chapter will show the work done and current result for my project.

4.1 Hardware Setup

The hardware involved in this project is laptop, VisionFive V1 board and RPi 4B Board. A laptop is used to connect to the board through SSH and do configuration to the board. Other than that, RPi 4B will also be used to complete any benchmarking that is cannot be found online. VisionFive V1 board is used to examine the performance of its processor and system.

Description	Specifications
Model	Illegear Onyx V series
Processor	AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
Operating System	Windows 11 64-bit
Graphic	NVIDIA GeForce GTX 1650
Memory	Samsung 16GB DDR4 3200MHz
Storage	2x 512 GB SSD SAMSUNG MZVLB5 12HBJQ-00007

Table 4.1: Specifications of laptop

Description	Specifications
Model	VisionFive V1 board
Processor	RISC-V SiFive U74 Dual-Core 64-bit RV64GC ISA SoC with 2MB L2 cache @ 1.0 GHz
Operating System	Fedora Linux
Video Processing	Video Decoder (H264/H265) up to 4K@60fps; Support Dual stream decoding for 2K@30 fps each
Memory	8GB LPDDR4
Storage	32GB of Micro-SD card

Table 4.2: Specifications of VisionFive V1 board

Description	Specifications
Model	Raspberry Pi 4 Model B Board
Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Operating System	Raspberry Pi OS (known as Raspbian)
Video Processing	H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode), OpenGL ES 3.1, Vulkan 1.0
Memory	8GB LPDDR4
Storage	32GB of Micro-SD card

Table 4.3: Specifications of RPi 4 Board

4.2 Software Setup

The software involved in this project is PuTTY and WinSCP.

PuTTY

PuTTY is a free software that allow users to connect other devices through SSH or telnet services [14]. It is used to connect the VisonFive V1 board in this project without the needs of displaying monitor and keyboard. All command can be easily done on terminal on my laptop once the software is successfully connected to the board.

WinSCP

WinSCP is a popular software for users to transfer the files between local computer and remote devices [15]. Once both devices are connected through SSH, the local computer can easily create, delete, and even copy the file to the remote devices. This had saved user's time to upload the file to cloud services and download it at the wanted device. In most of the time, file on VisionFive V1 board will be accessed through my laptop using WinSCP software.

Programming Language

The programming language used in this project is C language, C++ language and Python.

C Programming Language

C programming language is a middle-level of programming language which combines features of both low-level and high-level languages. It can be used for low-level programming such as kernel or system level of scripting and high-level programming such as software application [16].

C++ Programming Language

C++ programming language is an better version of C programming language. In most of the case, C++ programming language can compile over 99% of C programming without changing a single line of code. It is a lot safer than C programming language due to its object-oriented property [17].

Python

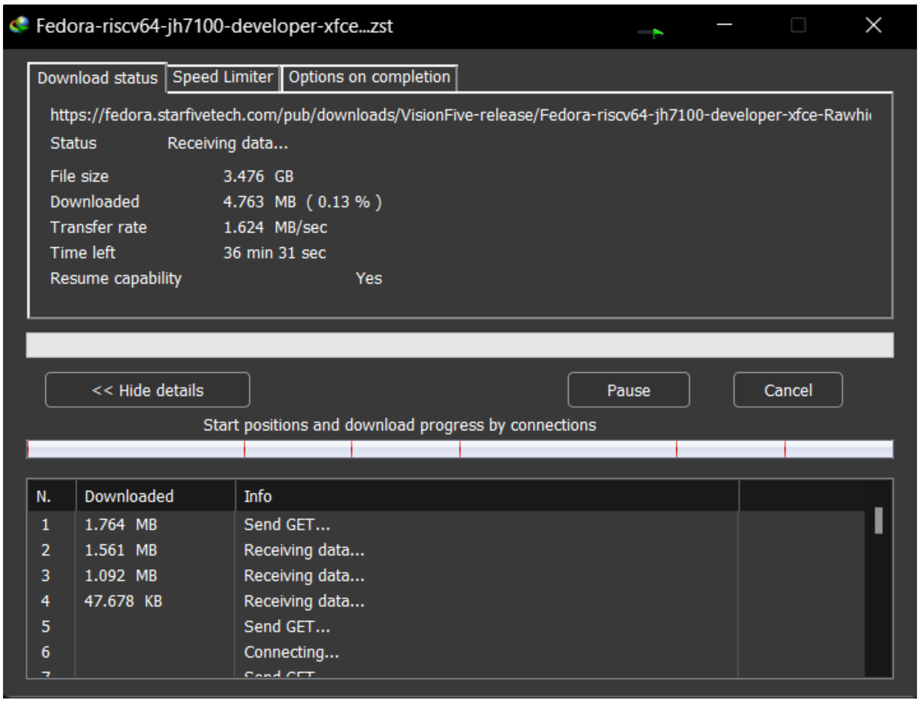
Python is a programming language that works on multiple different platform such as Windows, Linux, and Raspberry Pi. Due to its simplified syntax that emphasize on natural language, it is easy to be understand especially for those who are not developing the project. There is also lots of framework and libraries available in Python such as NumPy for scientific calculation and matplotlib for plotting graphs [18].

4.3 Setting and Configuration

4.3.1 Fedora OS

A micro-SD card will be used as a storage for VisionFive V1 Board and Fedora OS has to be installed into it. The following steps are the steps to install Fedora OS onto micro-SD card:

1) Download the Fedora OS image that made for RISCV64 architecture



Fedora-riscv64-jh7100-developer-xfce...zst

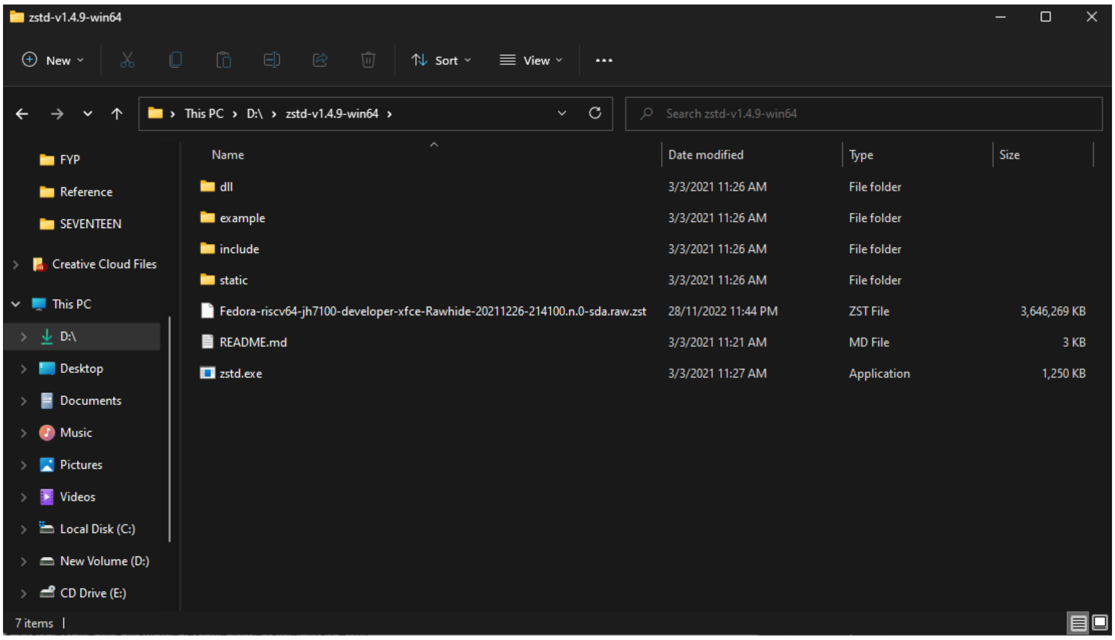
https://fedora.starfivetech.com/pub/downloads/VisionFive-release/Fedora-riscv64-jh7100-developer-xfce-Rawhi

Status	Receiving data...
File size	3.476 GB
Downloaded	4.763 MB (0.13 %)
Transfer rate	1.624 MB/sec
Time left	36 min 31 sec
Resume capability	Yes

Start positions and download progress by connections

N.	Downloaded	Info
1	1.764 MB	Send GET...
2	1.561 MB	Receiving data...
3	1.092 MB	Receiving data...
4	47.678 KB	Receiving data...
5		Send GET...
6		Connecting...
7		Send GET

Figure 4.3.1.1: Downloading Fedora OS Image

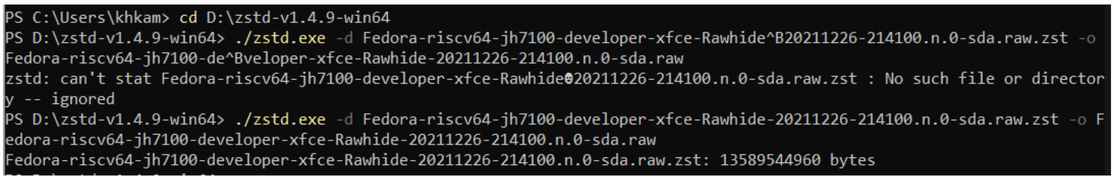


zstd-v1.4.9-win64

Name	Date modified	Type	Size
dll	3/3/2021 11:26 AM	File folder	
example	3/3/2021 11:26 AM	File folder	
include	3/3/2021 11:26 AM	File folder	
static	3/3/2021 11:26 AM	File folder	
Fedora-riscv64-jh7100-developer-xfce-Rawhide-20211226-214100.n.0-sda.raw.zst	28/11/2022 11:44 PM	ZST File	3,646,269 KB
README.md	3/3/2021 11:21 AM	MD File	3 KB
zstd.exe	3/3/2021 11:27 AM	Application	1,250 KB

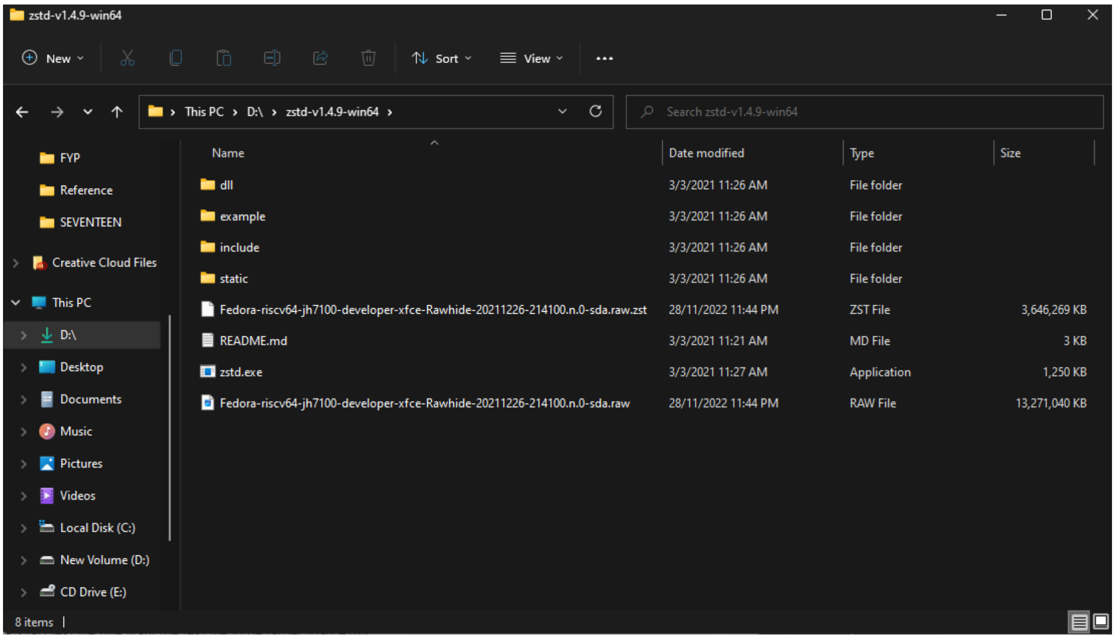
Figure 4.3.1.2: Fedora OS image had successfully downloaded

2) Extract the Fedora OS image that had been downloaded



```
PS C:\Users\khkam> cd D:\zstd-v1.4.9-win64
PS D:\zstd-v1.4.9-win64> ./zstd.exe -d Fedora-riscv64-jh7100-developer-xfce-Rawhide^B20211226-214100.n.0-sda.raw.zst -o
Fedora-riscv64-jh7100-de^Bveloper-xfce-Rawhide-20211226-214100.n.0-sda.raw
zstd: can't stat Fedora-riscv64-jh7100-developer-xfce-Rawhide@20211226-214100.n.0-sda.raw.zst : No such file or director
y -- ignored
PS D:\zstd-v1.4.9-win64> ./zstd.exe -d Fedora-riscv64-jh7100-developer-xfce-Rawhide-20211226-214100.n.0-sda.raw.zst -o F
edora-riscv64-jh7100-developer-xfce-Rawhide-20211226-214100.n.0-sda.raw
Fedora-riscv64-jh7100-developer-xfce-Rawhide-20211226-214100.n.0-sda.raw.zst: 13589544960 bytes
```

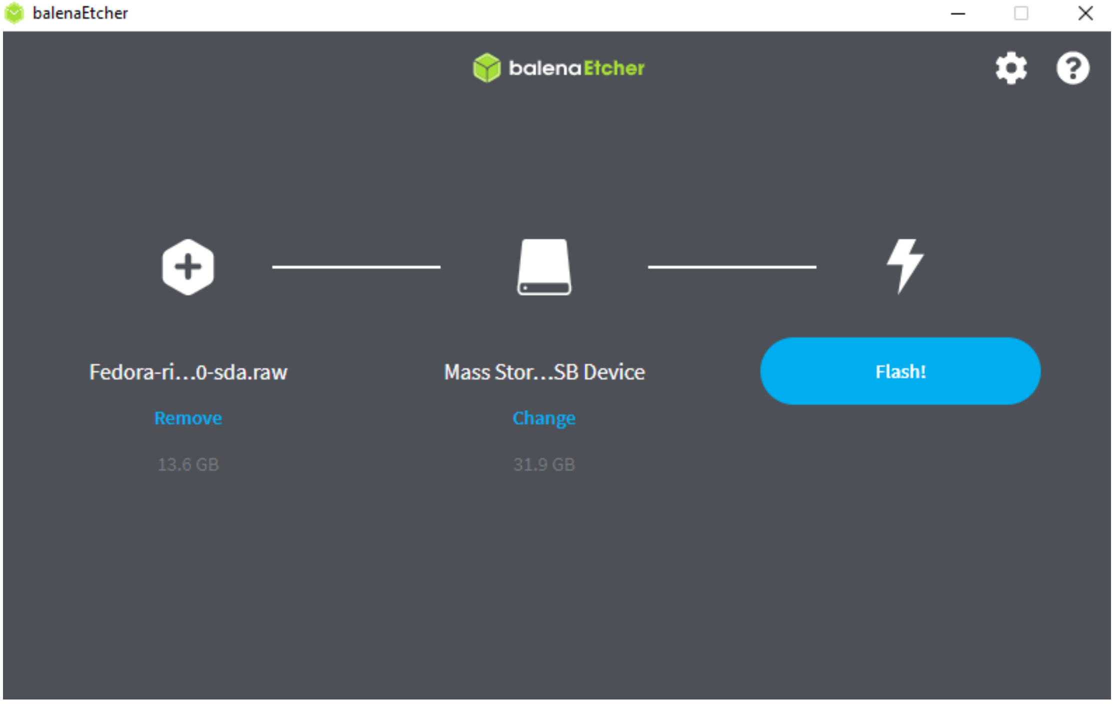
Figure 4.3.1.3: Extracting the fedora OS image.



Name	Date modified	Type	Size
dll	3/3/2021 11:26 AM	File folder	
example	3/3/2021 11:26 AM	File folder	
include	3/3/2021 11:26 AM	File folder	
static	3/3/2021 11:26 AM	File folder	
Fedora-riscv64-jh7100-developer-xfce-Rawhide-20211226-214100.n.0-sda.raw.zst	28/11/2022 11:44 PM	ZST File	3,646,269 KB
README.md	3/3/2021 11:21 AM	MD File	3 KB
zstd.exe	3/3/2021 11:27 AM	Application	1,250 KB
Fedora-riscv64-jh7100-developer-xfce-Rawhide-20211226-214100.n.0-sda.raw	28/11/2022 11:44 PM	RAW File	13,271,040 KB

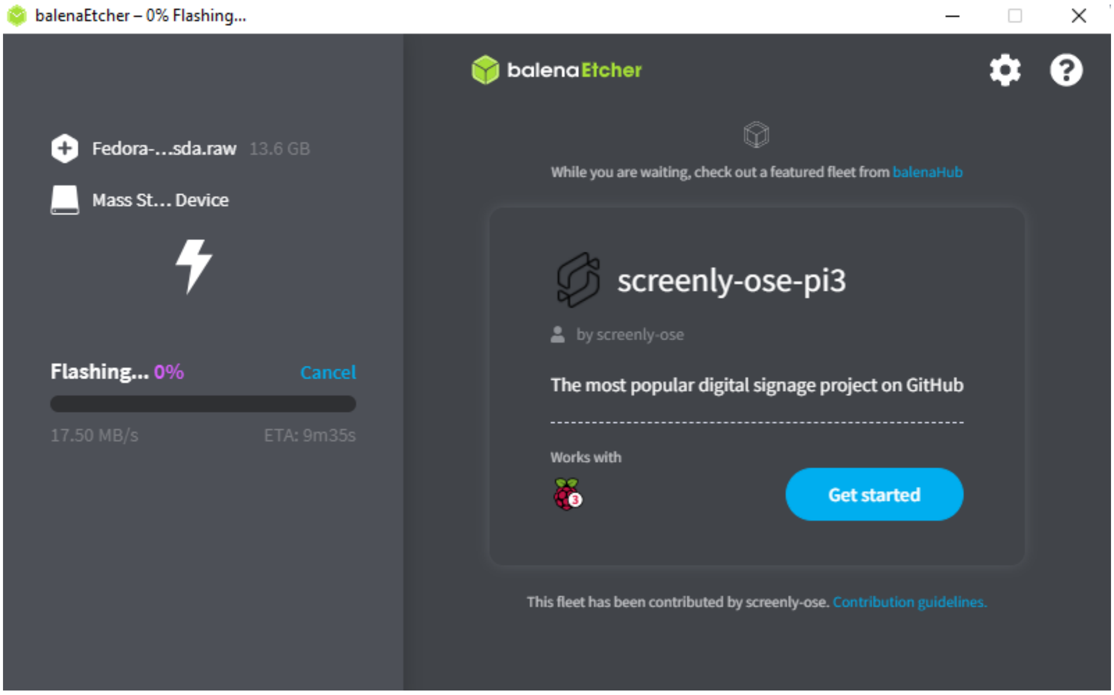
Figure 4.3.1.4: Fedora OS image had successfully installed.

3) Flash the Fedora OS image onto micro-SD card using balenaEtcher.



Fedora-ri…0-sda.raw
Remove
13.6 GB

Mass Stor…SB Device
Change
31.9 GB



balenaEtcher – 0% Flashing...

balenaEtcher

Fedora-...sda.raw 13.6 GB

Mass St... Device

Flashing... 0% Cancel

17.50 MB/s ETA: 9m35s

While you are waiting, check out a featured fleet from balenaHub

screenly-ose-pi3

by screenly-ose

The most popular digital signage project on GitHub

Works with

Get started

This fleet has been contributed by screenly-ose. Contribution guidelines.

Figure 4.3.1.5: Flash the Fedora OS image onto micro-SD

4) Fedora OS image had successfully installed.

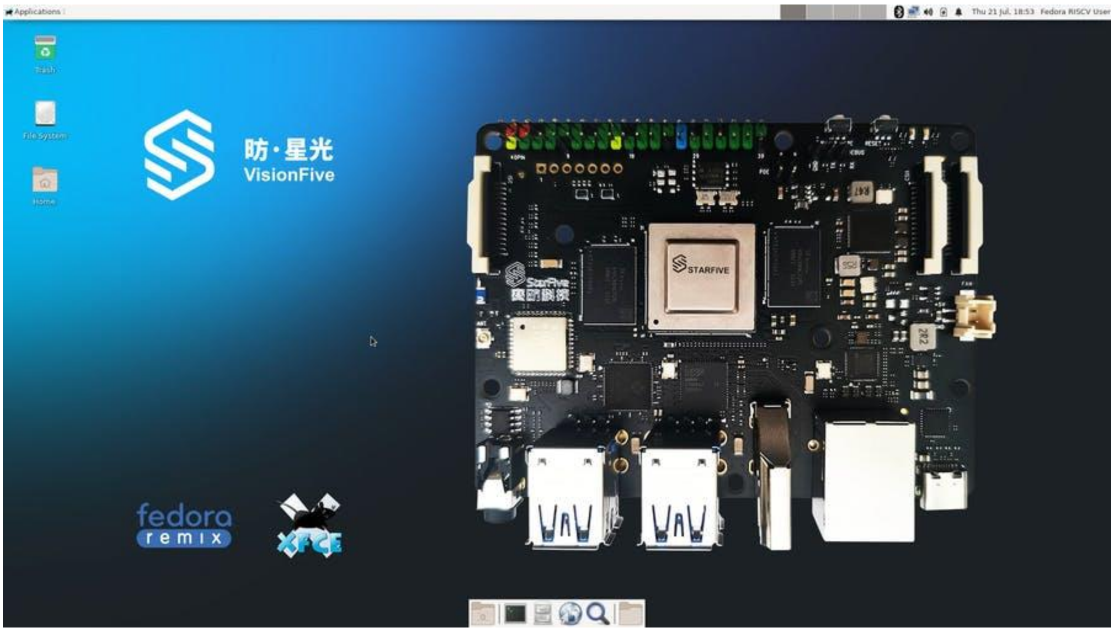
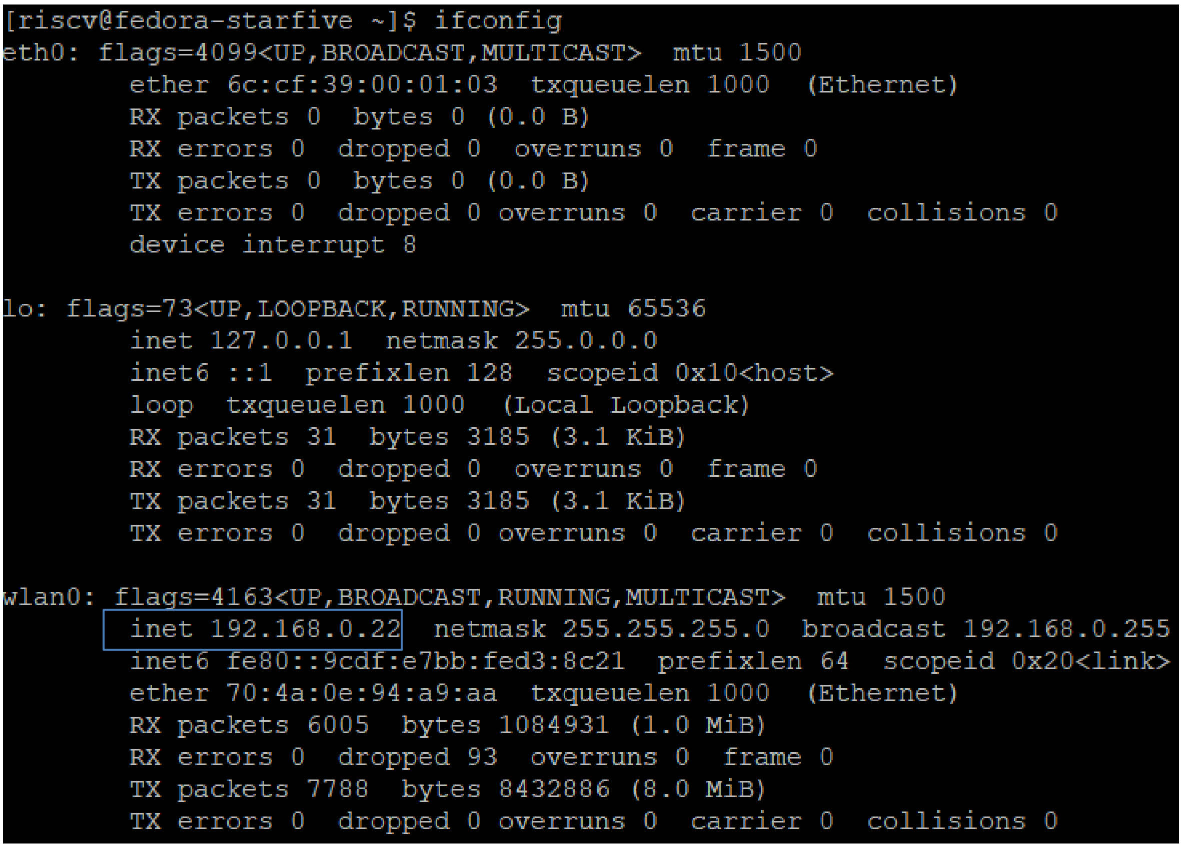


Figure 4.3.1.6: Home screen of VisionFive V1 Board

4.3.2 Connect to the VisionFive V1 Board using PuTTY and WinSCP

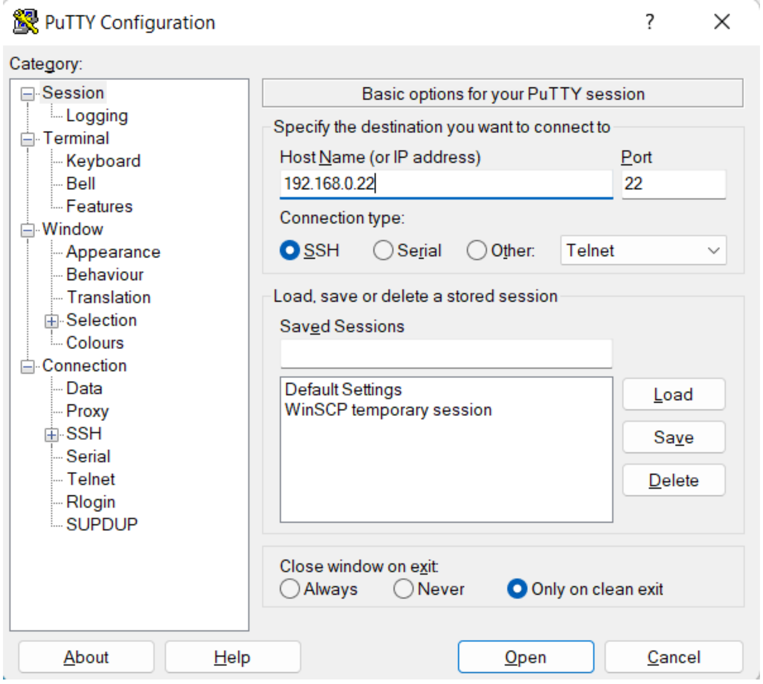
1) Get the IP address of the board that connected to a WiFi.



```
[riscv@fedora-starfive ~]$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 6c:cf:39:00:01:03  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 8

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 31  bytes 3185 (3.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31  bytes 3185 (3.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.22  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::9cdf:e7bb:fed3:8c21  prefixlen 64  scopeid 0x20<link>
        ether 70:4a:0e:94:a9:aa  txqueuelen 1000  (Ethernet)
        RX packets 6005  bytes 1084931 (1.0 MiB)
        RX errors 0  dropped 93  overruns 0  frame 0
        TX packets 7788  bytes 8432886 (8.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 4.3.2.1: IP address of the device

2) Insert the IP address of the board using port 22 which is SSH protocol.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.22

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

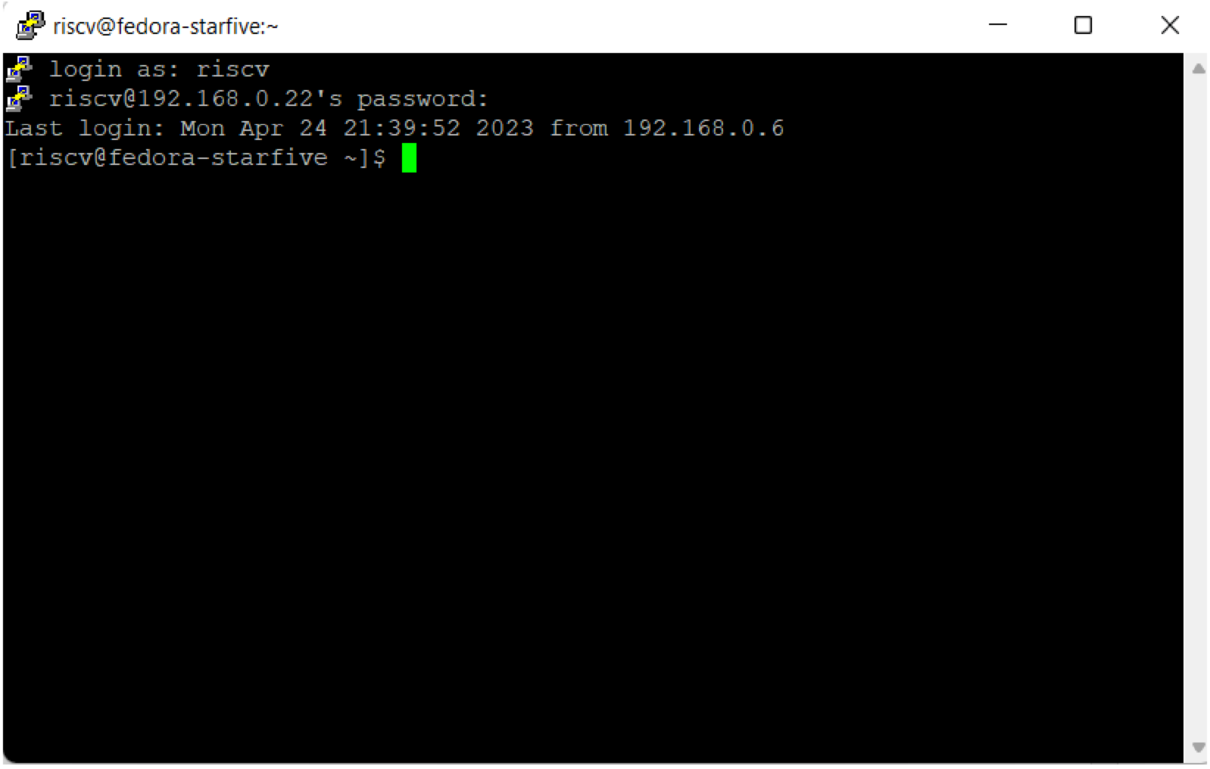
Saved Sessions

Default Settings
WinSCP temporary session

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 4.3.2.2: PuTTY User Interface

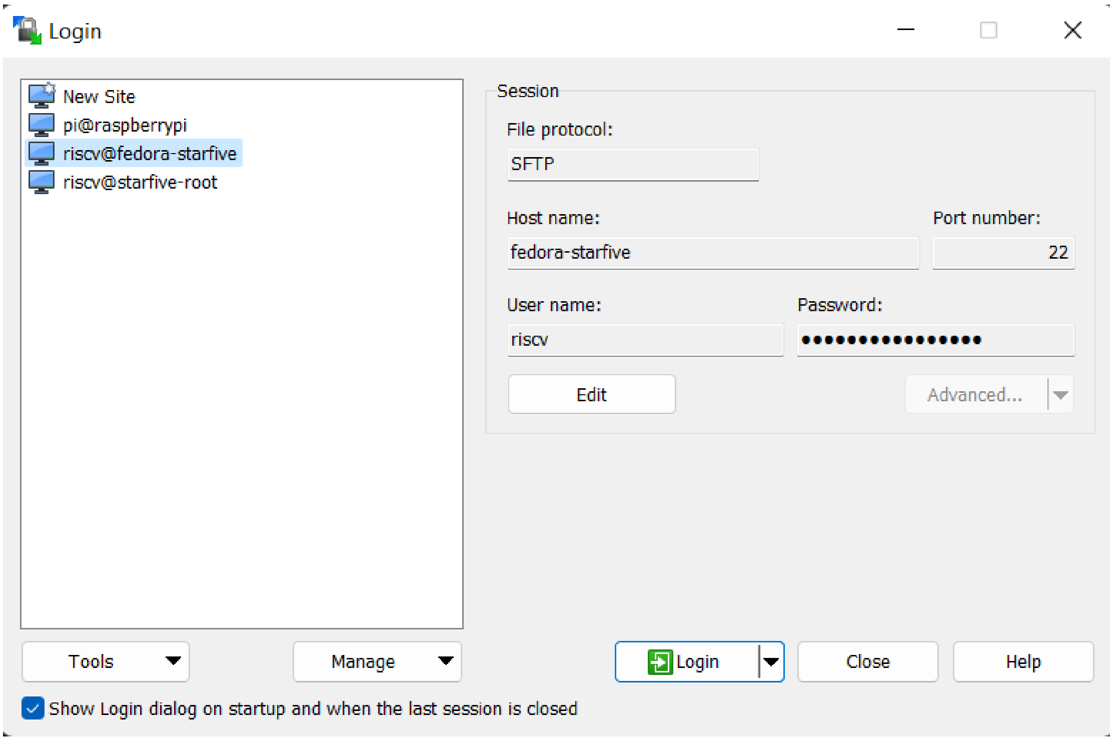
3) Login role and password will be asked upon login.



```
login as: riscv
riscv@192.168.0.22's password:
Last login: Mon Apr 24 21:39:52 2023 from 192.168.0.6
[riscv@fedora-starfive ~]$
```

Figure 4.3.2.3: Key in the password and role on PuTTY interface.

4) For WinSCP, the required information needs to be key in before trying to login to the board.



Session

File protocol: SFTP

Host name: fedora-starfive

Port number: 22

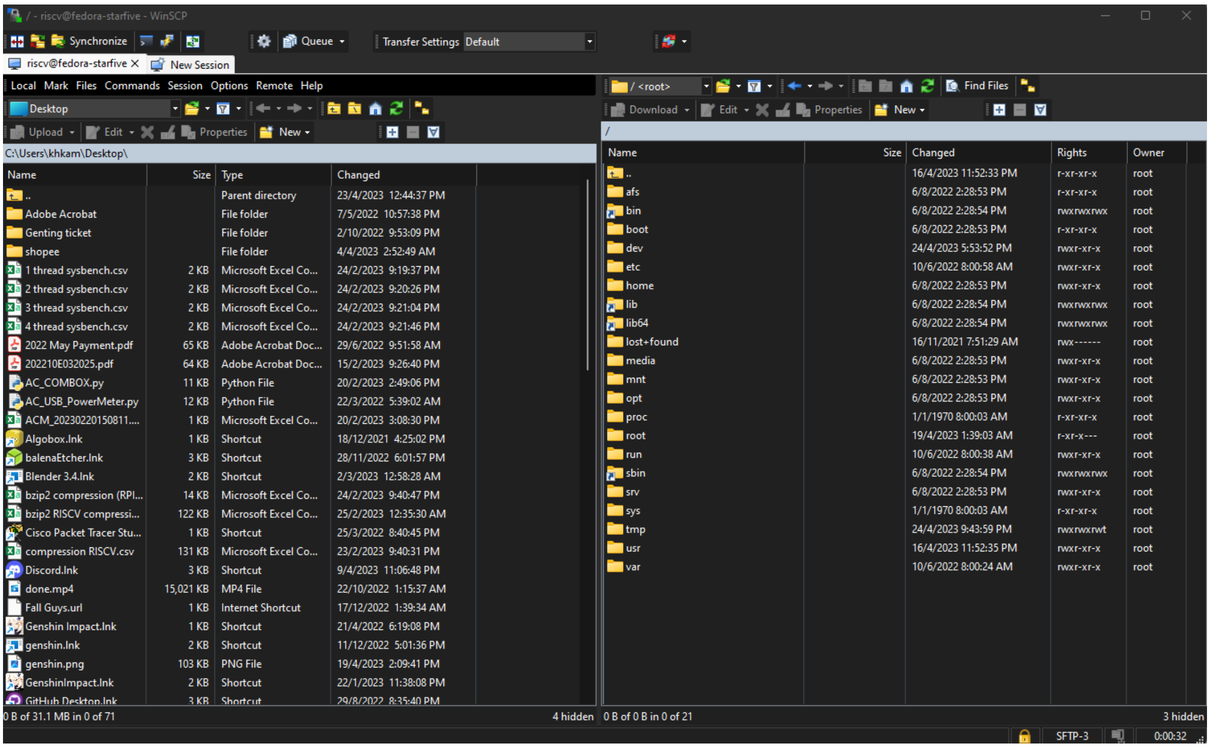
User name: riscv

Password: ••••••••••••••••

☑ Show Login dialog on startup and when the last session is closed

Figure 4.3.2.4: User interface of WinSCP

5) A workspace will be created after logged in. Left side will be local machine and right side is remoted machine.



Transfer Settings Default

C:\Users\khkam\Desktop\

Name	Size	Type	Changed
..		Parent directory	23/4/2023 12:44:37 PM
Adobe Acrobat		File folder	7/5/2022 10:57:38 PM
Genting ticket		File folder	2/10/2022 9:53:09 PM
shopee		File folder	4/4/2023 2:52:49 AM
1 thread sysbench.csv	2 KB	Microsoft Excel Co...	24/2/2023 9:19:37 PM
2 thread sysbench.csv	2 KB	Microsoft Excel Co...	24/2/2023 9:20:26 PM
3 thread sysbench.csv	2 KB	Microsoft Excel Co...	24/2/2023 9:21:04 PM
4 thread sysbench.csv	2 KB	Microsoft Excel Co...	24/2/2023 9:21:46 PM
2022 May Payment.pdf	65 KB	Adobe Acrobat Doc...	29/6/2022 9:51:58 AM
202210E032025.pdf	64 KB	Adobe Acrobat Doc...	15/2/2023 9:26:40 PM
AC_COMBOX.py	11 KB	Python File	20/2/2023 2:49:06 PM
AC_USB_PowerMeter.py	12 KB	Python File	22/3/2022 5:39:02 AM
ACM_20230220150811....	1 KB	Microsoft Excel Co...	20/2/2023 3:08:30 PM
Algobox.lnk	1 KB	Shortcut	18/12/2021 4:25:02 PM
balenaEtcher.lnk	3 KB	Shortcut	28/11/2022 6:01:57 PM
Blender 3.4.lnk	2 KB	Shortcut	2/3/2023 12:58:28 AM
bzip2 compression (RPI...	14 KB	Microsoft Excel Co...	24/2/2023 9:40:47 PM
bzip2 RISCV compressi...	122 KB	Microsoft Excel Co...	25/2/2023 12:35:30 AM
Cisco Packet Tracer Stu...	1 KB	Shortcut	25/3/2022 8:40:45 PM
compression RISCV.csv	131 KB	Microsoft Excel Co...	23/2/2023 9:40:31 PM
Discord.lnk	3 KB	Shortcut	9/4/2023 11:06:48 PM
done.mp4	15,021 KB	MP4 File	22/10/2022 1:15:37 AM
Fall Guys.url	1 KB	Internet Shortcut	17/12/2022 1:39:34 AM
Genshin Impact.lnk	1 KB	Shortcut	21/4/2022 6:19:08 PM
genshin.lnk	2 KB	Shortcut	11/12/2022 5:01:36 PM
genshin.png	103 KB	PNG File	19/4/2023 2:09:41 PM
GenshinImpact.lnk	2 KB	Shortcut	22/1/2023 11:38:08 PM
GitHub Desktop.lnk	3 KB	Shortcut	29/8/2022 8:35:40 PM

0 B of 31.1 MB in 0 of 71 4 hidden

/

Name	Size	Changed	Rights	Owner
..		16/4/2023 11:52:33 PM	r-xr-xr-x	root
afs		6/8/2022 2:28:53 PM	r-xr-xr-x	root
bin		6/8/2022 2:28:54 PM	rwxrwxrwx	root
boot		6/8/2022 2:28:53 PM	r-xr-xr-x	root
dev		24/4/2023 5:53:52 PM	rwxr-xr-x	root
etc		10/6/2022 8:00:58 AM	rwxr-xr-x	root
home		6/8/2022 2:28:53 PM	rwxr-xr-x	root
lib		6/8/2022 2:28:54 PM	rwxrwxrwx	root
lib64		6/8/2022 2:28:54 PM	rwxrwxrwx	root
lost+found		16/11/2021 7:51:29 AM	rwx------	root
media		6/8/2022 2:28:53 PM	rwxr-xr-x	root
mnt		6/8/2022 2:28:53 PM	rwxr-xr-x	root
opt		6/8/2022 2:28:53 PM	rwxr-xr-x	root
proc		1/1/1970 8:00:03 AM	r-xr-xr-x	root
root		19/4/2023 1:39:03 AM	r-xr-x---	root
run		10/6/2022 8:00:38 AM	rwxr-xr-x	root
sbin		6/8/2022 2:28:54 PM	rwxrwxrwx	root
srv		6/8/2022 2:28:53 PM	rwxr-xr-x	root
sys		1/1/1970 8:00:03 AM	r-xr-xr-x	root
tmp		24/4/2023 9:43:59 PM	rwxrwxrwt	root
usr		16/4/2023 11:52:35 PM	rwxr-xr-x	root
var		10/6/2022 8:00:24 AM	rwxr-xr-x	root

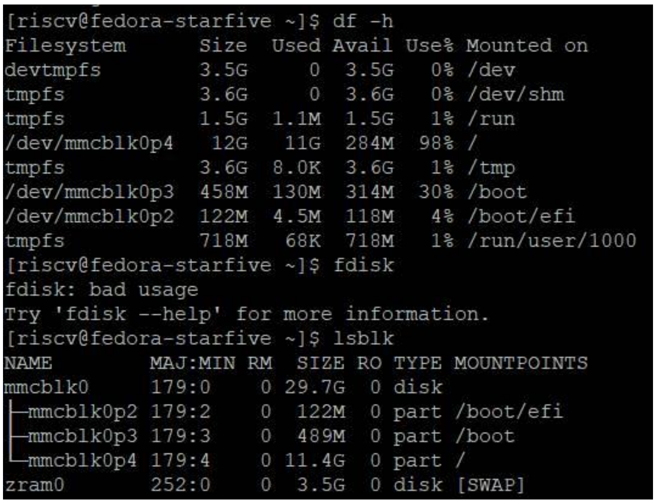
0 B of 0 B in 0 of 21 3 hidden

SFTP-3 0:00:32

Figure 4.3.2.5: Workspace of WinSCP after successfully login.

4.3.3 Resizing the partition of VisionFive V1 Board

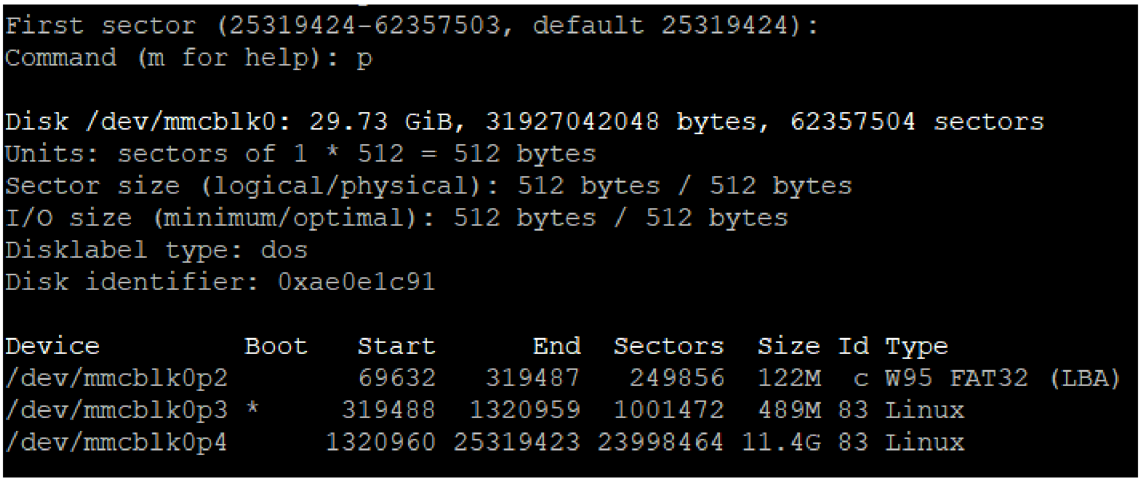
1) The partition will be resized since the operating system did not fully utilize the disk space. The SD card used is 32GB, but it only used up to 12 GB in mmcblk0 parent partition which the main partition, mmcblk0p4 only takes 11.4 GB. It means that there is still a lot of free spaces did not utilize.



```
[riscv@fedora-starfive ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.5G     0  3.5G   0% /dev
tmpfs           3.6G     0  3.6G   0% /dev/shm
tmpfs           1.5G  1.1M  1.5G   1% /run
/dev/mmcblk0p4   12G   11G  284M  98% /
tmpfs           3.6G  8.0K  3.6G   1% /tmp
/dev/mmcblk0p3  458M  130M  314M  30% /boot
/dev/mmcblk0p2  122M  4.5M  118M   4% /boot/efi
tmpfs           718M   68K  718M   1% /run/user/1000
[riscv@fedora-starfive ~]$ fdisk
fdisk: bad usage
Try 'fdisk --help' for more information.
[riscv@fedora-starfive ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
mmcblk0     179:0    0 29.7G  0 disk
├─mmcblk0p2 179:2    0  122M  0 part /boot/efi
├─mmcblk0p3 179:3    0  489M  0 part /boot
└─mmcblk0p4 179:4    0 11.4G  0 part /
zram0       252:0    0  3.5G  0 disk [SWAP]
```

Figure 4.3.3.1: Information of disk space before resizing.

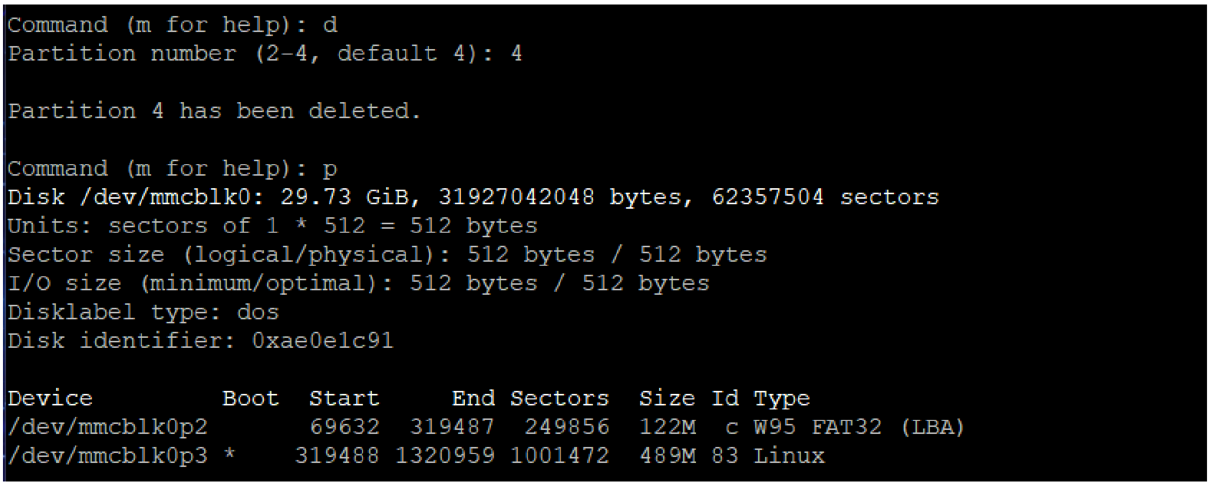
2) Use command “fdisk /dev/mmcblk0” to switch to partition edit mode.



```
First sector (25319424-62357503, default 25319424):
Command (m for help): p

Disk /dev/mmcblk0: 29.73 GiB, 31927042048 bytes, 62357504 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xae0e1c91

Device          Boot   Start      End  Sectors  Size Id Type
/dev/mmcblk0p2         69632   319487   249856  122M  c W95 FAT32 (LBA)
/dev/mmcblk0p3 *      319488  1320959  1001472  489M 83 Linux
/dev/mmcblk0p4       1320960 25319423 23998464 11.4G 83 Linux
```

Figure 4.3.3.2: Partition edit mode.

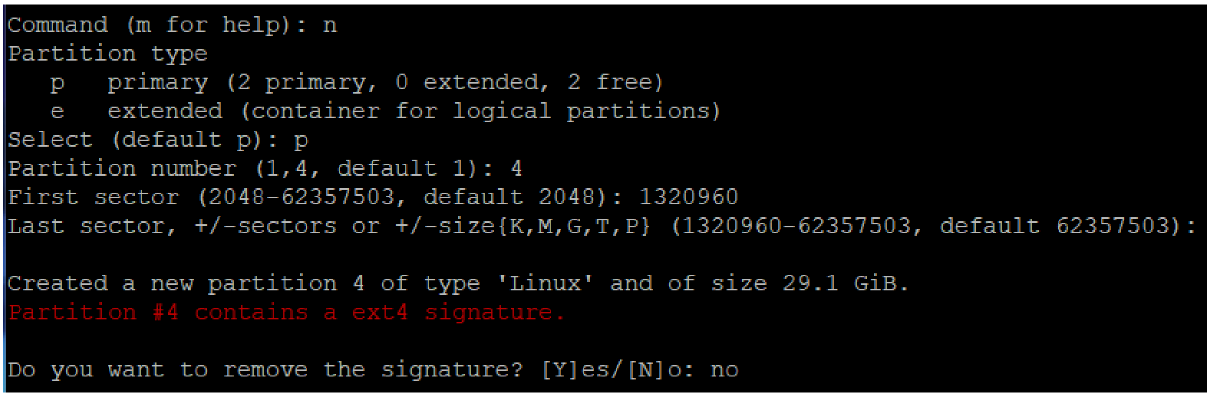
3) Use command “d” which indicate delete feature to delete the partition 4.



```
Command (m for help): d
Partition number (2-4, default 4): 4

Partition 4 has been deleted.

Command (m for help): p
Disk /dev/mmcblk0: 29.73 GiB, 31927042048 bytes, 62357504 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xae0e1c91

Device          Boot  Start     End Sectors  Size Id Type
/dev/mmcblk0p2        69632  319487  249856  122M  c W95 FAT32 (LBA)
/dev/mmcblk0p3 *     319488 1320959 1001472  489M 83 Linux
```

Figure 4.3.3.3: Delete the main partition which does not fully utilize the disk space.

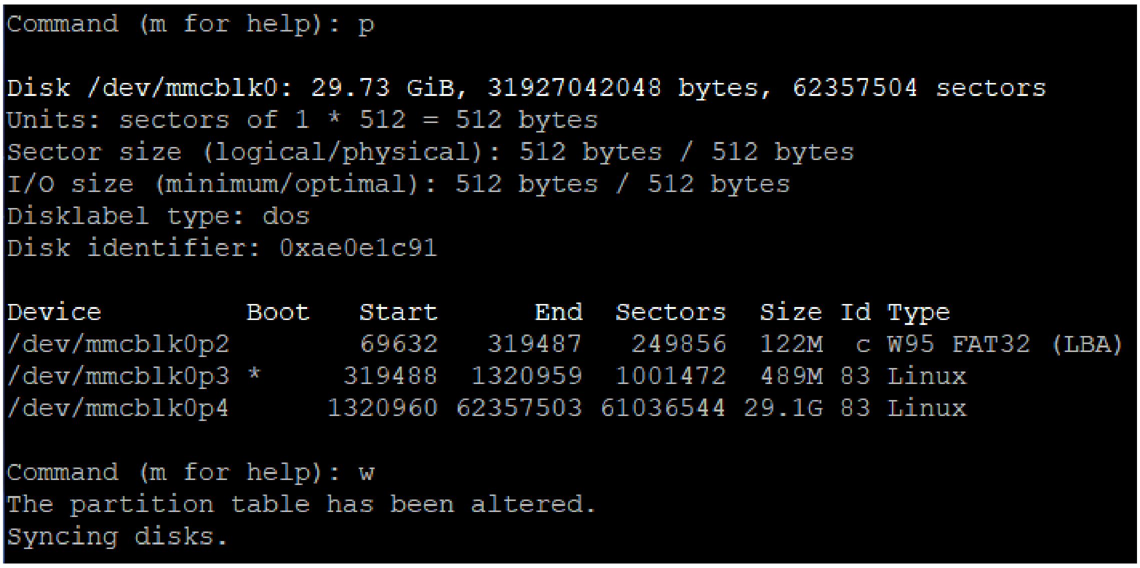
4) Use command “n” which indicates new partition feature to create a new partition. Select the first sector as the start point of partition mmcblk0p4 in step 2.



```
Command (m for help): n
Partition type
   p   primary (2 primary, 0 extended, 2 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1,4, default 1): 4
First sector (2048-62357503, default 2048): 1320960
Last sector, +/-sectors or +/-size{K,M,G,T,P} (1320960-62357503, default 62357503):

Created a new partition 4 of type 'Linux' and of size 29.1 GiB.
Partition #4 contains a ext4 signature.

Do you want to remove the signature? [Y]es/[N]o: no
```

Figure 4.3.3.4: Create a new partition.

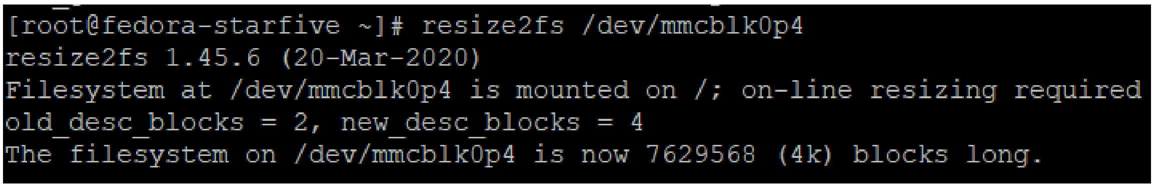
5) Sync the partition size that had configured.



```
Command (m for help): p

Disk /dev/mmcblk0: 29.73 GiB, 31927042048 bytes, 62357504 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xae0e1c91

Device         Boot   Start      End  Sectors  Size Id Type
/dev/mmcblk0p2        69632   319487   249856  122M  c W95 FAT32 (LBA)
/dev/mmcblk0p3 *     319488  1320959  1001472  489M 83 Linux
/dev/mmcblk0p4      1320960 62357503 61036544 29.1G 83 Linux

Command (m for help): w
The partition table has been altered.
Syncing disks.
```

Figure 4.3.3.5: Syncing the changes.

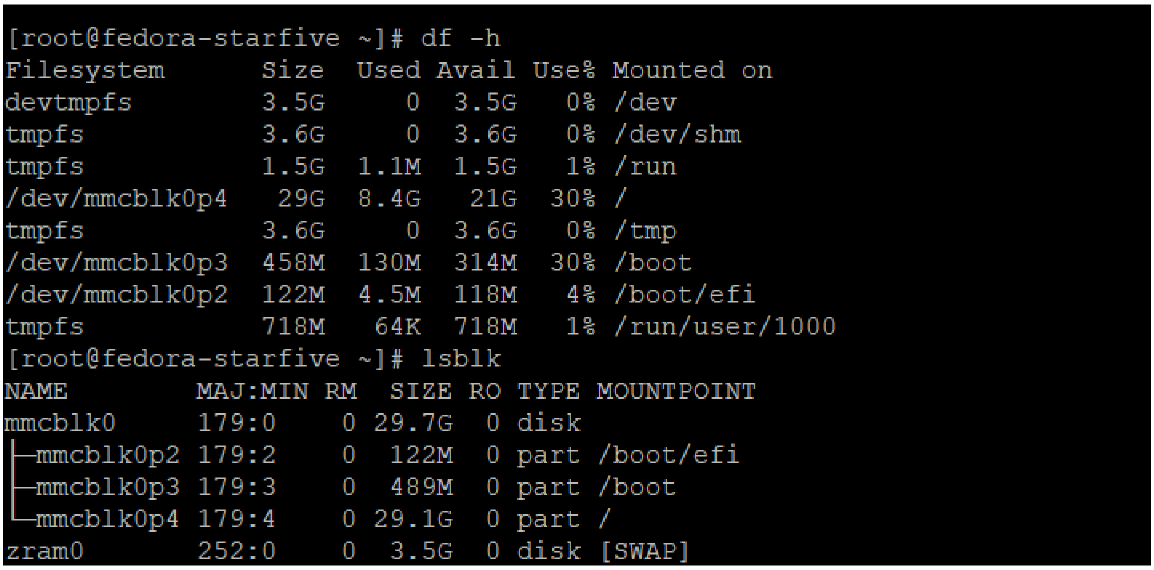
6) command “resize2fs” is used to enlarge or shrink an unmounted file system located on the device.



```
[root@fedora-starfive ~]# resize2fs /dev/mmcblk0p4
resize2fs 1.45.6 (20-Mar-2020)
Filesystem at /dev/mmcblk0p4 is mounted on /; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 4
The filesystem on /dev/mmcblk0p4 is now 7629568 (4k) blocks long.
```

Figure 4.3.3.6: Perform enlarging on the partition.

7) As a result, VisionFive V1 board had fully utilized the free space available on the SD card.



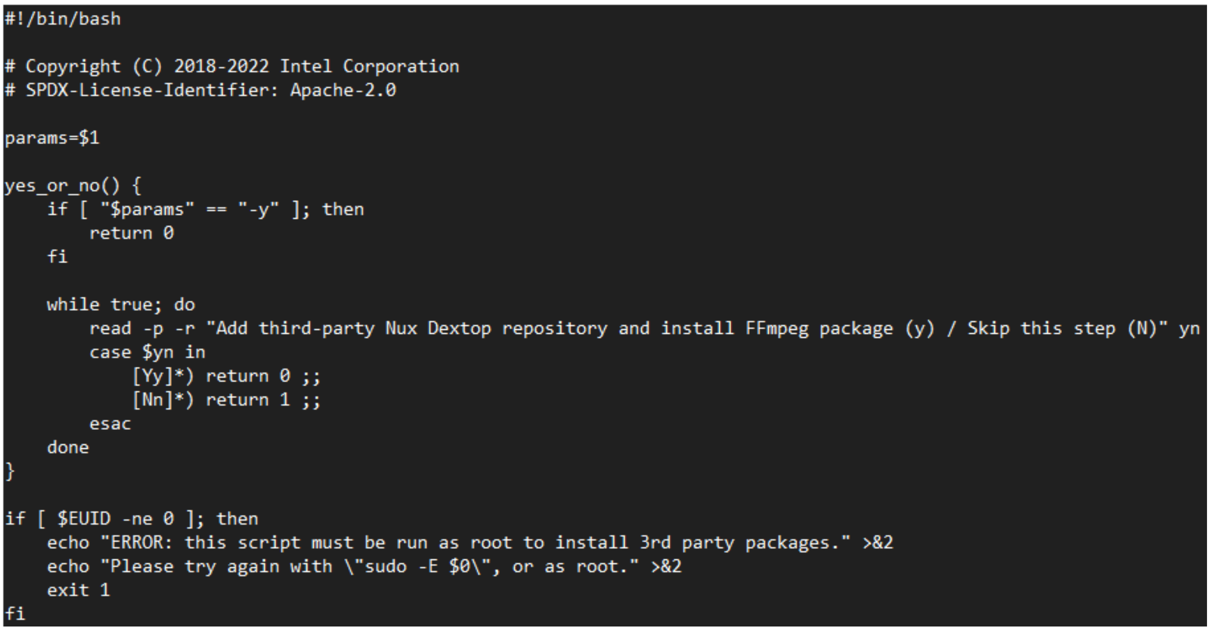
```
[root@fedora-starfive ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.5G     0  3.5G   0% /dev
tmpfs           3.6G     0  3.6G   0% /dev/shm
tmpfs           1.5G  1.1M  1.5G   1% /run
/dev/mmcblk0p4   29G  8.4G   21G  30% /
tmpfs           3.6G     0  3.6G   0% /tmp
/dev/mmcblk0p3  458M  130M  314M  30% /boot
/dev/mmcblk0p2  122M  4.5M  118M   4% /boot/efi
tmpfs           718M   64K  718M   1% /run/user/1000
[root@fedora-starfive ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0     179:0    0 29.7G  0 disk
├─mmcblk0p2 179:2    0  122M  0 part /boot/efi
├─mmcblk0p3 179:3    0  489M  0 part /boot
└─mmcblk0p4 179:4    0 29.1G  0 part /
zram0       252:0    0  3.5G  0 disk [SWAP]
```

Figure 4.3.3.7: Partition size after resizing.

4.4 System Operation

Before using PTS on the device, it was essential to ensure that the environment of the device was suitable for installing the platform. The first step was to use the command "sudo apt-get update" to obtain the latest updates of the operating system. Next, a PHP 5 command interpreter was required, and the command "sudo apt-get install php5-cli" was used to install the interpreter on the device. This step completed the basic installation process, and the next step was to install the PTS platform. The command "git clone https://github.com/phoronix-test-suite/phoronix-test-suite/" was used to install the latest version of PTS, which is 10.8.4. Alternatively, any other version of PTS could also be installed. After cloning the repository, the command "cd phoronix-test-suite" was used to navigate to the cloned directory. The command "git checkout v10.8.4" was used to switch to the latest version. Finally, the command "./install-sh" was used to install PTS.

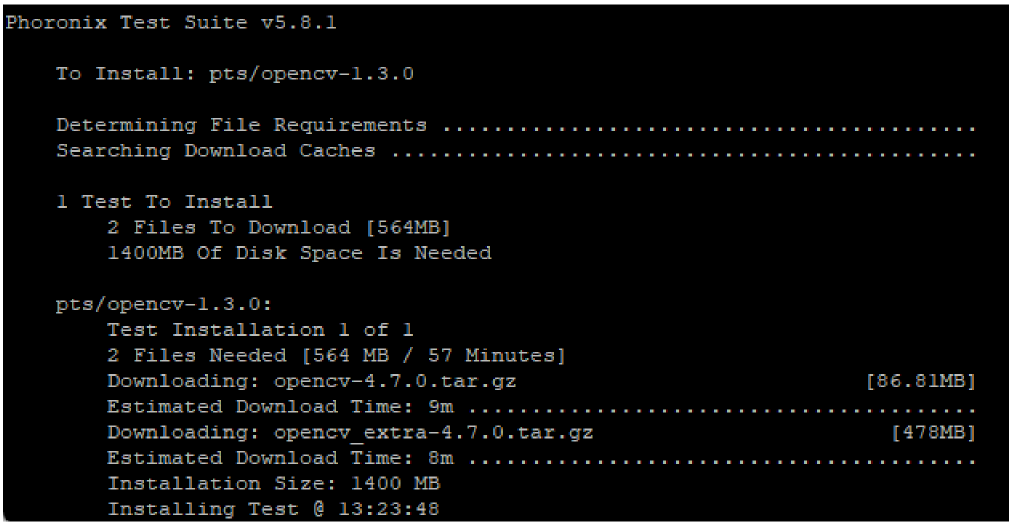
Next, a folder will be created for the test that is chosen to be performed. The installation file is available on [13]. The test can now be installed using the shell script. The platform will also help to install any extra libraries needed for the test.



```
#!/bin/bash

# Copyright (C) 2018-2022 Intel Corporation
# SPDX-License-Identifier: Apache-2.0

params=$1

yes_or_no() {
    if [ "$params" == "-y" ]; then
        return 0
    fi

    while true; do
        read -p -r "Add third-party Nux Dextop repository and install FFmpeg package (y) / Skip this step (N)" yn
        case $yn in
            [Yy]*) return 0 ;;
            [Nn]*) return 1 ;;
        esac
    done
}

if [ $EUID -ne 0 ]; then
    echo "ERROR: this script must be run as root to install 3rd party packages." >&2
    echo "Please try again with \"sudo -E $0\", or as root." >&2
    exit 1
fi
```

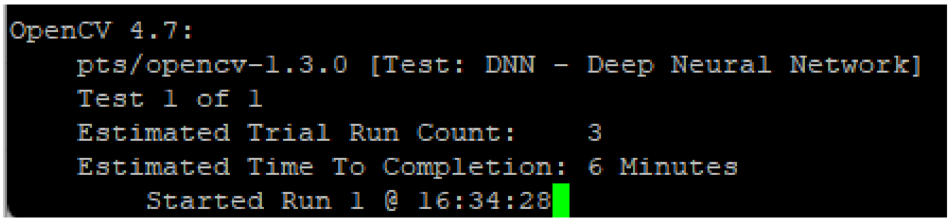
Figure 4.4.1: Sample shell script for installing the test.

After successfully installing test available on PTS, the user could start running tests using the command "phoronix-test-suite benchmark pts/<TEST NAME>". This command is used to benchmark the test when the test had successfully installed. The figure below shows an example of the installation and benchmarking process using PTS.



```
Phoronix Test Suite v5.8.1

    To Install: pts/opencv-1.3.0

    Determining File Requirements ......................................
    Searching Download Caches ..........................................

    1 Test To Install
        2 Files To Download [564MB]
        1400MB Of Disk Space Is Needed

    pts/opencv-1.3.0:
        Test Installation 1 of 1
        2 Files Needed [564 MB / 57 Minutes]
        Downloading: opencv-4.7.0.tar.gz                           [86.81MB]
        Estimated Download Time: 9m ....................................
        Downloading: opencv_extra-4.7.0.tar.gz                       [478MB]
        Estimated Download Time: 8m ....................................
        Installation Size: 1400 MB
        Installing Test @ 13:23:48
```

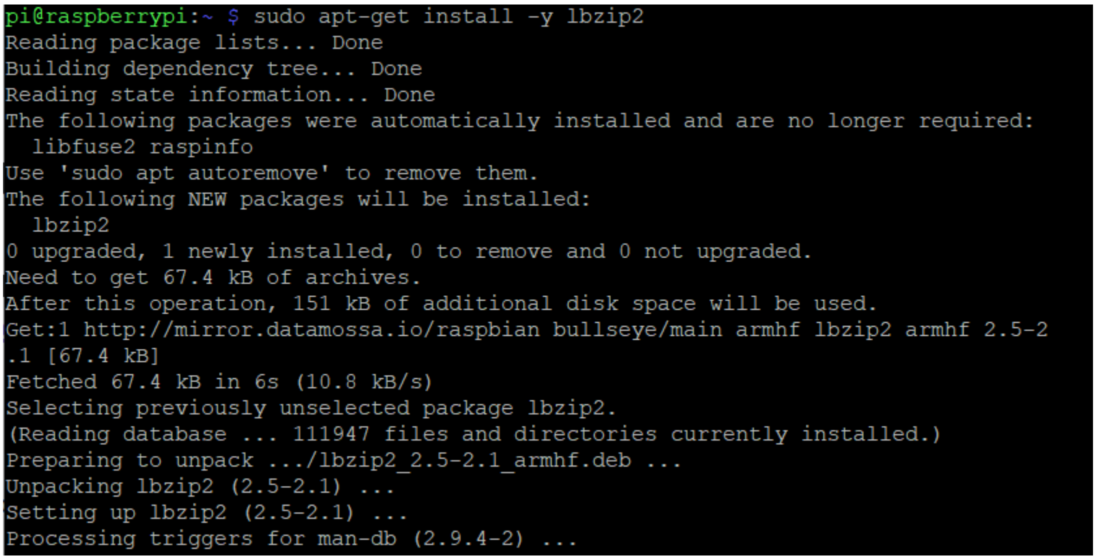
Figure 4.4.2: Example of installing a test on PTS platform.



```
OpenCV 4.7:
    pts/opencv-1.3.0 [Test: DNN - Deep Neural Network]
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 6 Minutes
        Started Run 1 @ 16:34:28
```

Figure 4.4.3: Example of benchmarking a test on PTS platform.

Apart from determining device performance using PTS, the thread performance of the device was also evaluated using dataset compression methods. Bzip2 and lbzip2 were chosen as the compression software for this project because they were faster tools by default and had both single-threaded and multi-threaded options [19]. Before using lbzip2, it had to be installed on the device. The figure below shows the installation process of lbzip2 compression software.

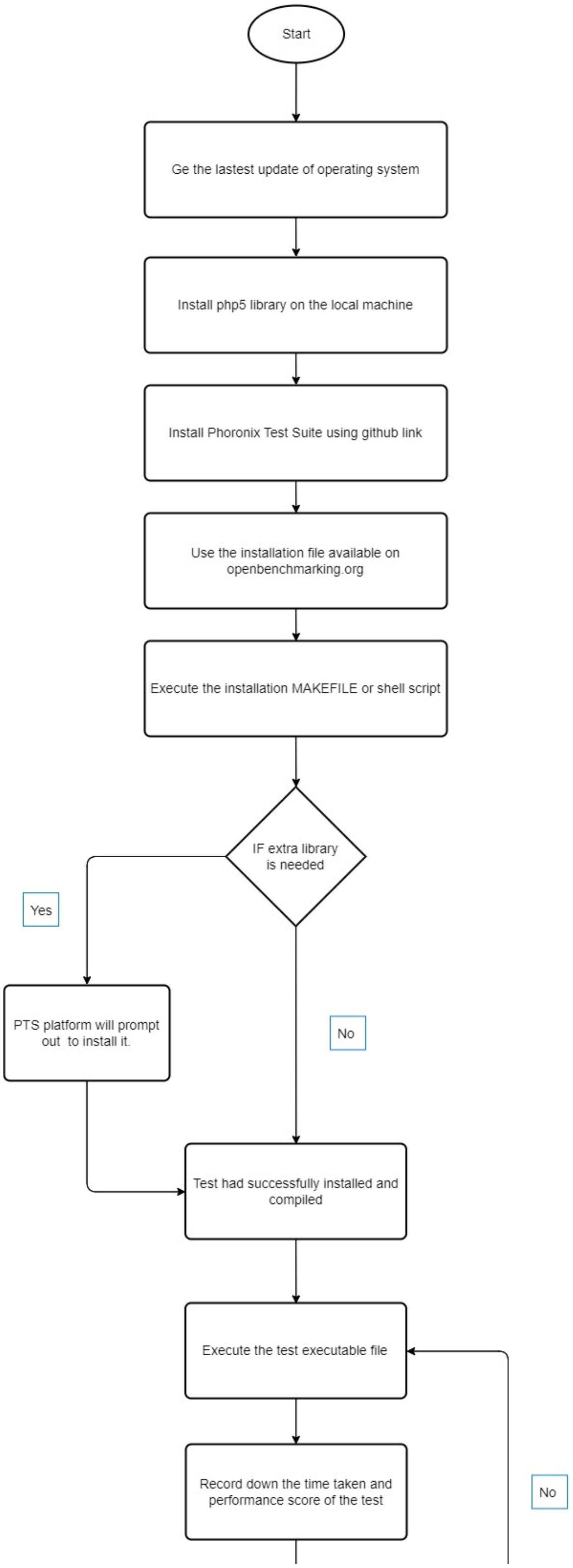


```
pi@raspberrypi:~ $ sudo apt-get install -y lbzip2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfuse2 raspinfo
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  lbzip2
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 67.4 kB of archives.
After this operation, 151 kB of additional disk space will be used.
Get:1 http://mirror.datamossa.io/raspbian bullseye/main armhf lbzip2 armhf 2.5-2
.1 [67.4 kB]
Fetched 67.4 kB in 6s (10.8 kB/s)
Selecting previously unselected package lbzip2.
(Reading database ... 111947 files and directories currently installed.)
Preparing to unpack .../lbzip2_2.5-2.1_armhf.deb ...
Unpacking lbzip2 (2.5-2.1) ...
Setting up lbzip2 (2.5-2.1) ...
Processing triggers for man-db (2.9.4-2) ...
```

Figure 4.4.4: Installation of lbzip2 compression software.

After successfully installing the compression tool, a dataset was generated using Python code that sized 1.19 GB in total. The dataset was then compressed using the number of threads ranging from 1 to 4 to determine thread performance and speed. Besides using dataset compression method, a customized micro-benchmark is also used to measure the performance of multiple thread by calculating Pi value. Finally, the power consumption when compressing the dataset will also be recorded down and analyzed in the later stage [20].

4.4.1 Phoronix Test Suite General Workflow


Start
Ge the lastest update of operating system
Install php5 library on the local machine
Install Phoronix Test Suite using github link
Use the installation file available on openbenchmarking.org
Execute the installation MAKEFILE or shell script
IF extra library is needed
Yes
PTS platform will prompt out to install it.
No
Test had successfully installed and compiled
Execute the test executable file
Record down the time taken and performance score of the test
No

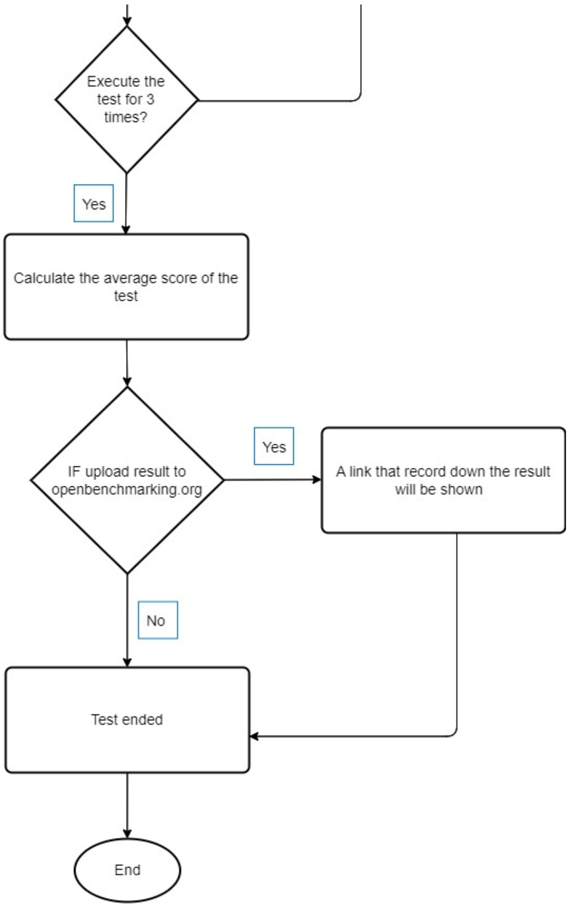
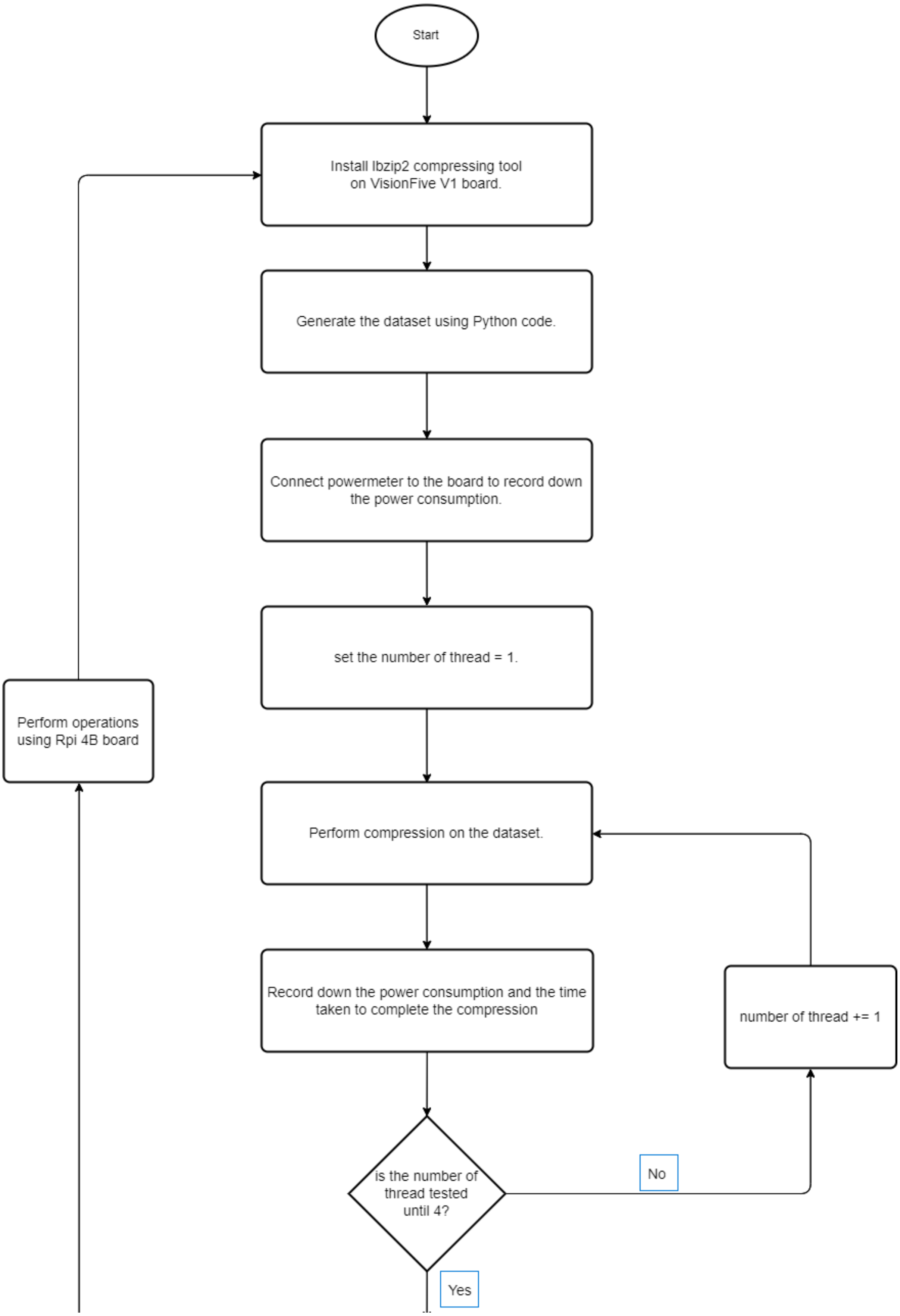

Execute the test for 3 times?
Yes
Calculate the average score of the test
IF upload result to openbenchmarking.org
Yes
A link that record down the result will be shown
No
Test ended
End

Figure 4.4.1.1: General Workflow of Phoronix Test Suite

4.4.2 Dataset Compression General Workflow


Start
Install lbzip2 compressing tool on VisionFive V1 board.
Generate the dataset using Python code.
Connect powermeter to the board to record down the power consumption.
set the number of thread = 1.
Perform compression on the dataset.
Record down the power consumption and the time taken to complete the compression
is the number of thread tested until 4?
No
number of thread += 1
Yes
Perform operations using Rpi 4B board

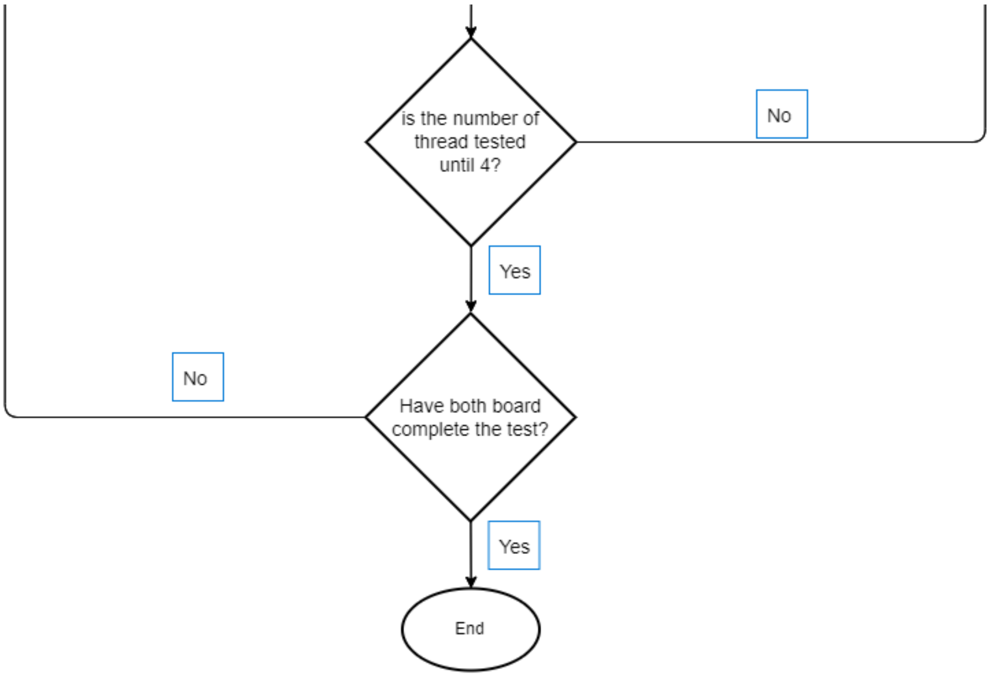

is the number of thread tested until 4?
No
Yes
No
Have both board complete the test?
Yes
End

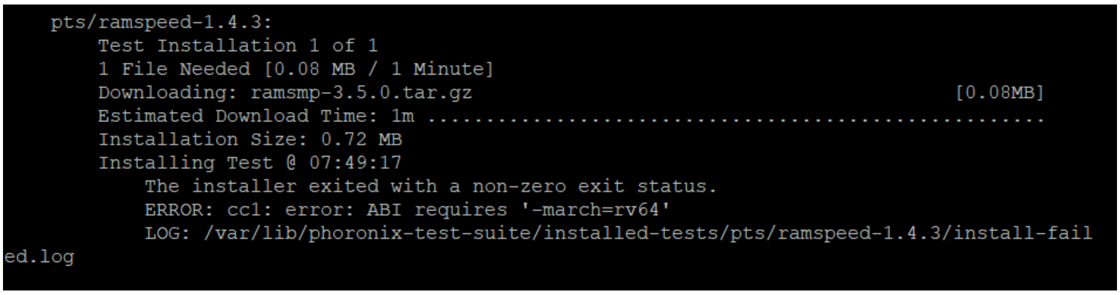
Figure 4.4.2.1: General Workflow of Dataset Compression

4.5 Verification Plan

The verification plan is by comparing the performance benchmarking result between VisionFive V1 Board and Raspberry Pi 4 board. User should be able to know the differences between them and make their choices to verify the best single board computer for their project.

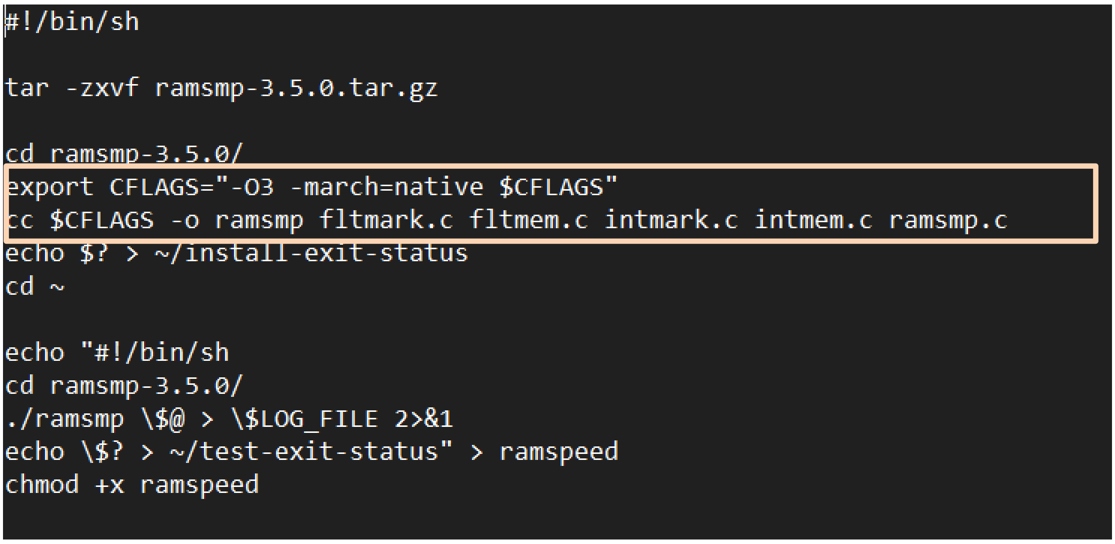
4.6 Implementation Issues and Challenges

When a project is running, it is normal that project member will face some implementation issues and challenges. One of the implementation issues and challenges that I faced during the project is some of the module does not suit the system perfectly which means that some of the configuration on the library file need to be done. For example, the default installation file for the module is *-march=native*. Native option is used for the user who does not know what setting to choose, and it makes the user to save time. However, an installation fail message occurred when I use this setting to install the library. Thus, I have to specify it as *rv64* in order to make the installation run completely.



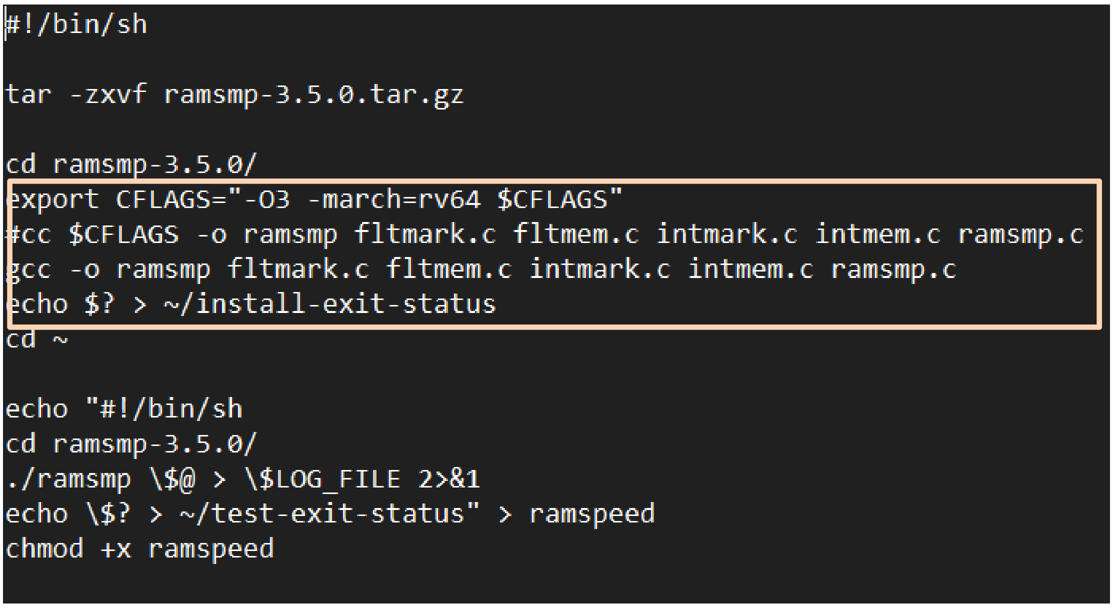
```
    pts/ramspeed-1.4.3:
        Test Installation 1 of 1
        1 File Needed [0.08 MB / 1 Minute]
        Downloading: ramsmp-3.5.0.tar.gz                                         [0.08MB]
        Estimated Download Time: 1m ............................................
        Installation Size: 0.72 MB
        Installing Test @ 07:49:17
            The installer exited with a non-zero exit status.
            ERROR: cc1: error: ABI requires '-march=rv64'
            LOG: /var/lib/phoronix-test-suite/installed-tests/pts/ramspeed-1.4.3/install-fail
ed.log
```

Figure 4.6.1: Test Installation Fail



```
#!/bin/sh

tar -zxvf ramsmp-3.5.0.tar.gz

cd ramsmp-3.5.0/
export CFLAGS="-O3 -march=native $CFLAGS"
cc $CFLAGS -o ramsmp fltmark.c fltmem.c intmark.c intmem.c ramsmp.c
echo $? > ~/install-exit-status
cd ~

echo "#!/bin/sh
cd ramsmp-3.5.0/
./ramsmp \$@ > \$LOG_FILE 2>&1
echo \$? > ~/test-exit-status" > ramspeed
chmod +x ramspeed
```

Figure 4.6.2: Installation before configuration



```
#!/bin/sh

tar -zxvf ramsmp-3.5.0.tar.gz

cd ramsmp-3.5.0/
export CFLAGS="-O3 -march=rv64 $CFLAGS"
#cc $CFLAGS -o ramsmp fltmark.c fltmem.c intmark.c intmem.c ramsmp.c
gcc -o ramsmp fltmark.c fltmem.c intmark.c intmem.c ramsmp.c
echo $? > ~/install-exit-status
cd ~

echo "#!/bin/sh
cd ramsmp-3.5.0/
./ramsmp \$@ > \$LOG_FILE 2>&1
echo \$? > ~/test-exit-status" > ramspeed
chmod +x ramspeed
```

Figure 4.6.3: Installation file after configuration

Other than that, I had also found that some of the flags is missing in the library installation file which will makes the installation cannot be done. Therefore, I have to add those flags myself so that the installation can run smoothly. Not only flags, the build type for installation file sometimes can also be missing and I found out that I have to specify the build type based on the architecture used. As an example, if it is an ARM architecture, the build type needs to be specified as arm-linux. For this project's case, the build type will be set as riscv-linux.

Moreover, some of the benchmark test done for RPi 4 is not recorded in the existing documentation. The performance benchmarking test will be performed on both VisionFive V1 Board and Raspberry Pi 4 Board so that the reader can compare the score obtained by both boards directly.

CHAPTER 5 – System Evaluation and Discussion

5.1 System Testing and Performance Metrics

The testing process consists of several testing, including CPU testing, RAM performance testing, thread performance testing, power consumption testing and machine learning test. Each tests involved different aspects of the system and helps to compare the performance of both different architecture device. In this section, we will discuss each phase of the testing process in more detail. Next, the performance metrics used to evaluate the performance of both device will be the throughput, CPU Usage, time to complete a task and the power needed to complete a task.

5.1.1 CPU Testing

Central Processing Unit (CPU) is one of the most important components of a computer system. It is responsible for executing instructions and performing arithmetic and logic operations. The performance of the CPU can have a significant impact on the overall system performance. However, the different type of CPU will consist of different number of cores which will also bring greatly impact to the system performance.

5.1.2 RAM Performance Testing

RAM is an essential component of a computer system. It provides temporary storage for data that the CPU needs to access quickly. The performance of the RAM can have a significant impact on the overall system performance. Therefore, it is important to test the RAM to ensure that it is performing optimally. The RAM performance will be determined by using the read and write operation to the OS.

5.1.3 Thread Performance Testing

Thread performance testing is an important aspect of system testing, particularly for multi-core processors, as it measures the performance of the system when multiple threads are executing simultaneously. Thread performance testing is particularly relevant for applications that are designed to run on multi-core processors, as they can take advantage of the additional processing power to improve their performance. The number of threads will be controlled during the test to let them have a fair environment to produce the result.

5.1.4 Power Consumption Testing

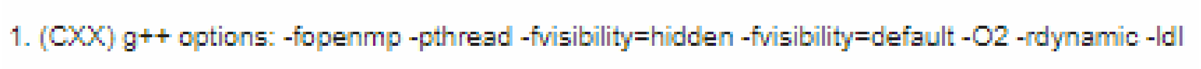
Power consumption testing is also important in system testing, particularly for portable devices such as laptops and smartphones, as it measures the amount of power consumed by the system during different activities. Power consumption testing is particularly relevant for battery life testing, as it can help determine the consumption of energy after certain amount of time.

5.1.5 Machine Learning Testing

Machine learning algorithms are designed to learn from data and improve their performance over time. A machine learning benchmarking is essential for single board computer as most of the IoT product will be invented using single board computer as its core.

5.2 Testing Setup and Result

The approach taken in this project to test the performance of the VisionFive V1 Board and the RPi 4B Board is to compare their computational times using the same workload. To ensure a fair environment for both boards, the tests that run on the PTS platform use the default g++ flag option. This means that both boards are subject to the same settings, and their performance results can be accurately compared. The g++ flag option used in this project is a set of compiler flags that provide a balance between optimization and code size. This option is known to produce good results in most cases and is suitable for the tasks performed in this project.



```
1. (CXX) g++ options: -fopenmp -pthread -fvisibility=hidden -fvisibility=default -O2 -rdynamic -ldl
```

Figure 5.2.1: Default g++ flag option for the test on PTS.

In addition to using the same workload for both boards, the same dataset is also used for the dataset compression stage. This is to ensure that the performance comparison is not affected by differences in the data being used. During the testing process, a power supply is connected to a power meter to record the power consumption of both boards. This data is used to compare the power efficiency of the two boards.

Finally, a customized micro-benchmark introduced in this project is used to compute Pi with the same workload for both boards. The computational time required for each board to complete this task is used to determine its performance.

Overall, the testing setup for this project is designed to create a fair and controlled environment for testing the performance of the VisionFive V1 Board and the RPi 4B Board. By using the same workload, dataset, and power supply, the results obtained will be accurate and unbiased. This will help to determine which board is more efficient and suitable for specific tasks.

5.2.1 Phoronix Test Suite Test Details and Result

Table 5.2.1.1: Tests that had been carried out for both boards.

	Test	Test Details
1)	FLAC Audio Encoding	This test examines how long the system takes to encode a simple WAV file become FLAC format for five time. This is to examine the performance of single core since FLAC encoder only uses one thread to complete the task.
2)	Himeno Benchmark	This test is used to examine the performance of multi-core by using linear solver of pressure Poisson.
3)	Tinymembench	This is a test that examine the system memory performance.
4)	t-test1 - 1 Thread - 2 Threads	This is a basic test to evaluate the memory allocator performance. Both 1 thread and 2 threads options are chosen to evaluate its performance from all perspective.
5)	CacheBench - Read Cache Test - Write Cache Test	This is another test that benchmark the memory and cache bandwidth performance. Read and write cache test is performed in this test.
6)	RAMspeed SMP	This test examines the system memory performance speed by copying a bunch of integers to and from RAM.
7)	PostMark	This test simulates small file that similar to the tasks endured by web and mail servers. Then, the speed of performing 25,000 transactions with 500 files with file sizes between 5 and 512 kilobytes will be taken down.

8)	Dolfyn	Dolfyn is a test that measures the execution time the bundled computational fluid dynamics demos.
9)	FFTE	This is a test that compute and count the discrete fourier transform of 1 to 3 dimensional sequences of length.
10)	WebP Image Encode - Default option - Quality 100 - Quality 100, Lossless - Quality 100, Highest Compression - Quality 100, Lossless, Highest Compression	Webp Image Encode is a test that examine the system image encoding ability by using a sample 6000x4000 pixel JPEG image. Different encode options are set to provide a better view of different scenario for user.
11)	GraphicsMagick - Swirl - Rotate - Sharpen - Enhanced - Resizing - Noise-Gaussian - HWB Color Space	This is another test that make use of 6000x4000 pixel JPEG sample image to examine the system performance. However, the difference is this test performs different imaging operations on sample image and get the duration of completely performing the operations.
12)	Coremark	This test provides the processor benchmark.
13)	AOBench	AOBench is a simple test that examine the speed of renderer written in C language. The test profile file used is 2048x2048.
14)	Perl Benchmarks - Pod2html - Interpreter	This test will be used to examine the speed between different Perl version which is by using Pod2html and interpreter.
15)	RNNoise	RNNoise is another single-threaded test which measuring the time of denoise a 26 minutes long and 16-bit sample audio.

16)	OpenSSL	This is a benchmark test that examine the performance by using SSL and TLS transaction speed.
17)	SQLite Speedtest	Benchmark test that uses SQLite to increase the problem size up to 1000. The time required to complete the task will be taken down.
18)	PyPerformance - go - 2to3 - chaos - float - nbody - pathlib - raytrace - json-loads - crypto_pyaes - regex_compile - python_startup - django_template - pickle_pure_python	As the name implies, this is a test that by referencing python performance benchmark. All the options for Python are available in this suite.
19)	PHPBench	It is suite that evaluate PHP performance. It runs a large number of simple tests to evaluate the PHP interpreter. It is mostly used to compare hardware, operating system and etc.
20)	OpenCV - Deep Neural Network (DNN) - Image Processing	It is a benchmarking test that examine the performance of OpenCV (Computer Vision) library's built in function.
21)	Numpy Benchmark	Numpy is a package that particularly for scientific computing in Python. It can be used in various type of operations such as fast operations on arrays, including mathematical, logical, shape manipulation and etc.

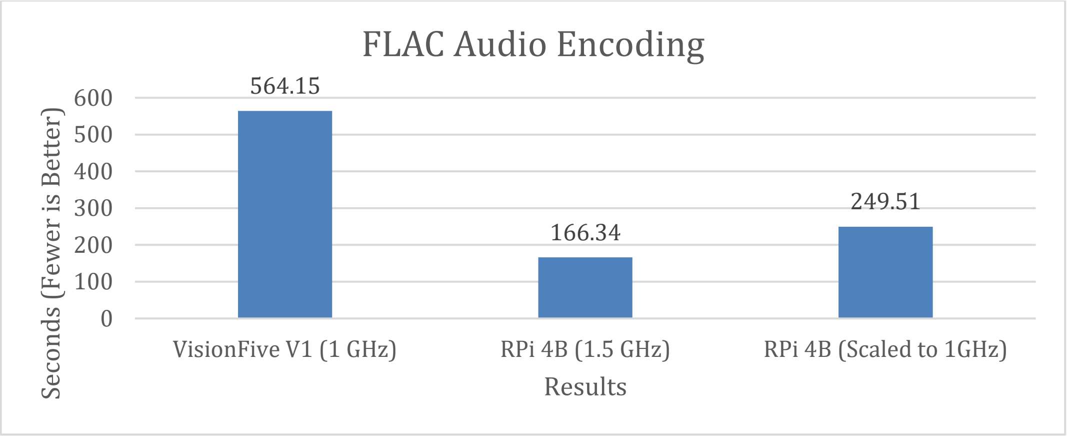


Figure 5.2.1.1: Bar chart of FLAC Audio Encoding performance comparison for both boards.

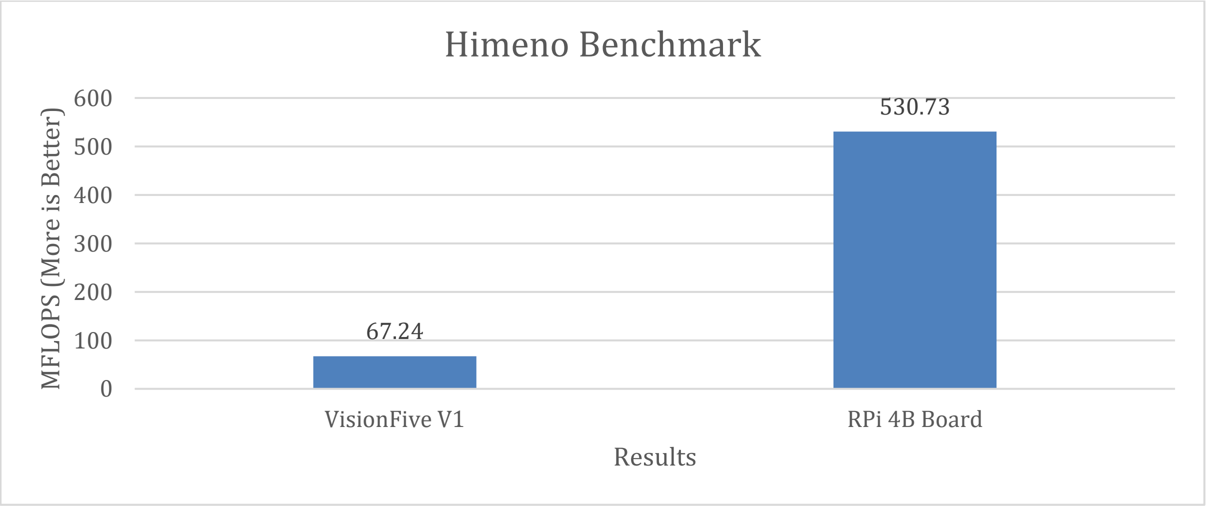


Figure 5.2.1.2: Bar chart of Himeno Benchmark performance comparison for both boards.

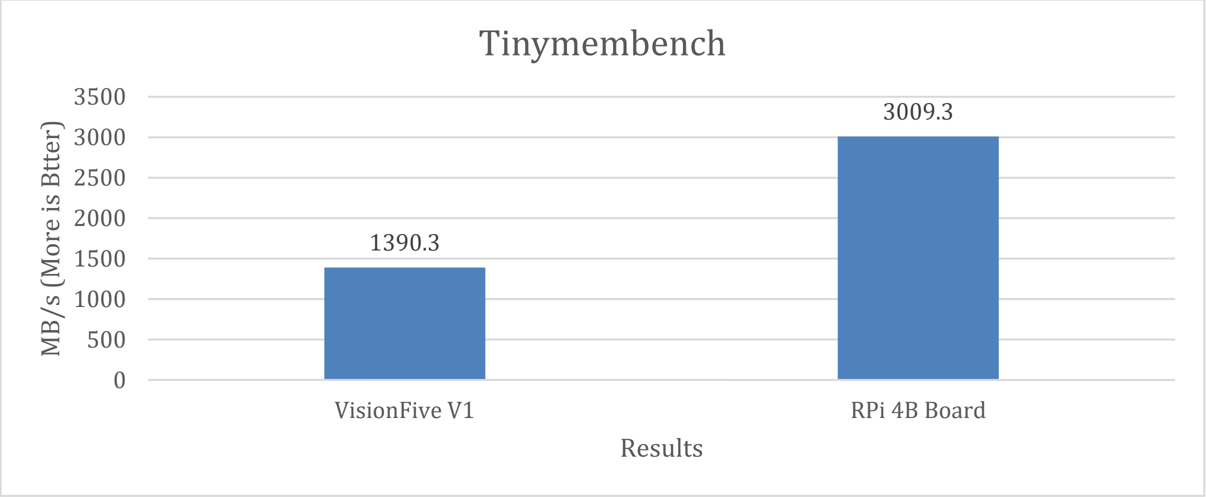


Figure 5.2.1.3: Bar chart of Tinymembench performance comparison for both boards

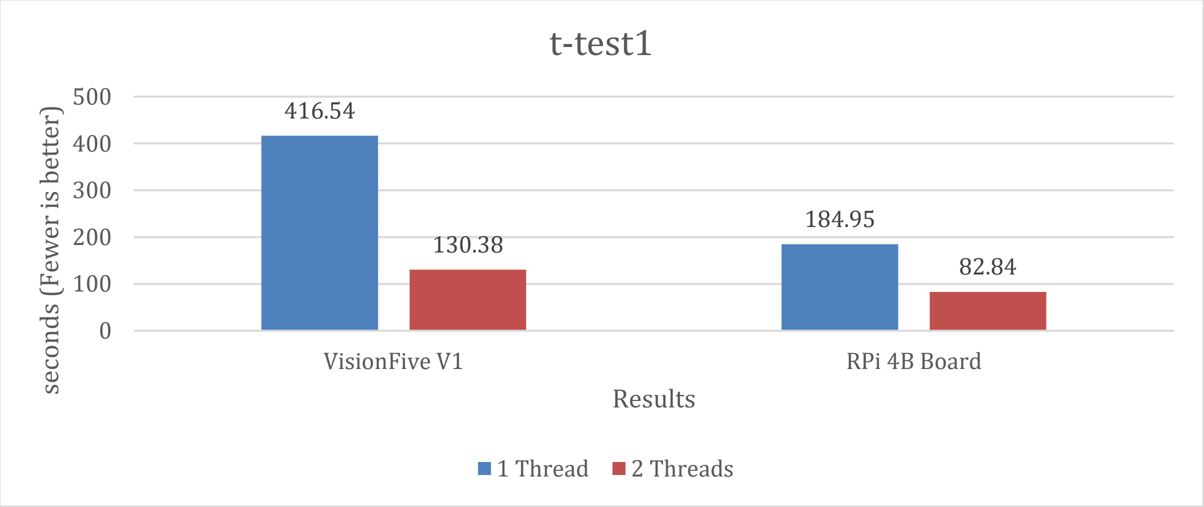


Figure 5.2.1.4: Bar chart of t-test1 performance comparison for both boards

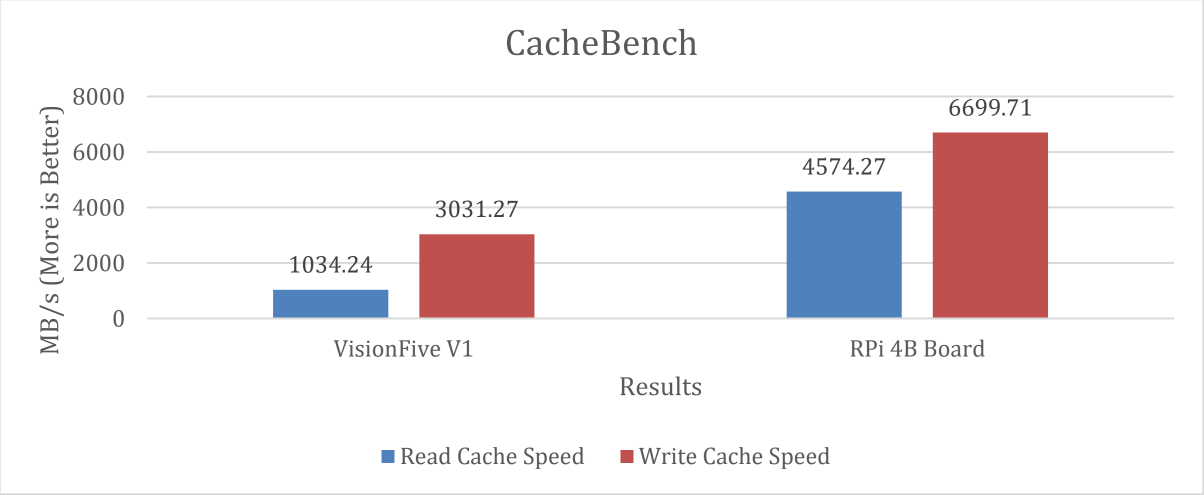


Figure 5.2.1.5: Bar chart of CacheBench performance comparison for both boards.

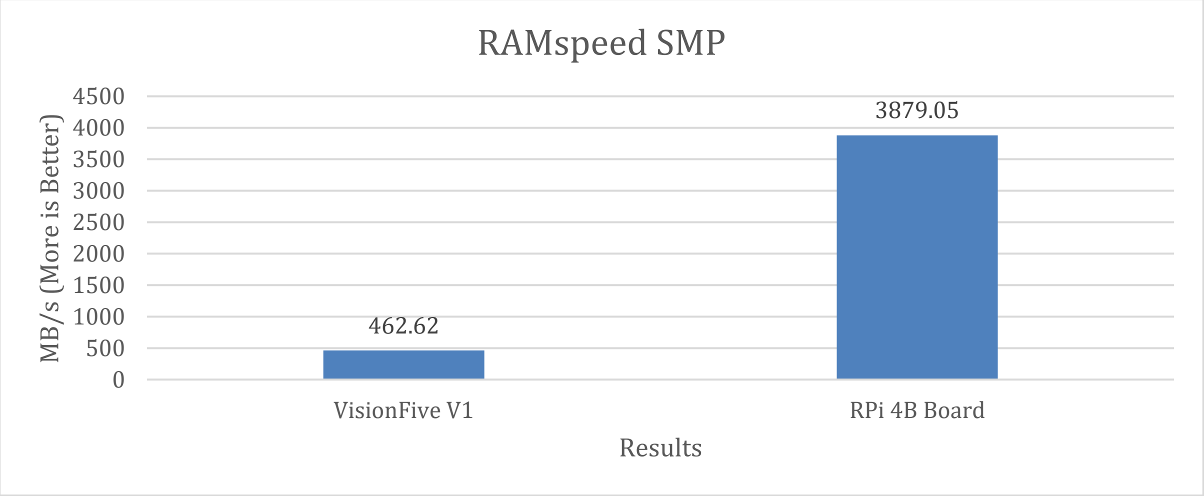


Figure 5.2.1.6: Bar chart of RAMspeed SMP performance comparison for both boards.

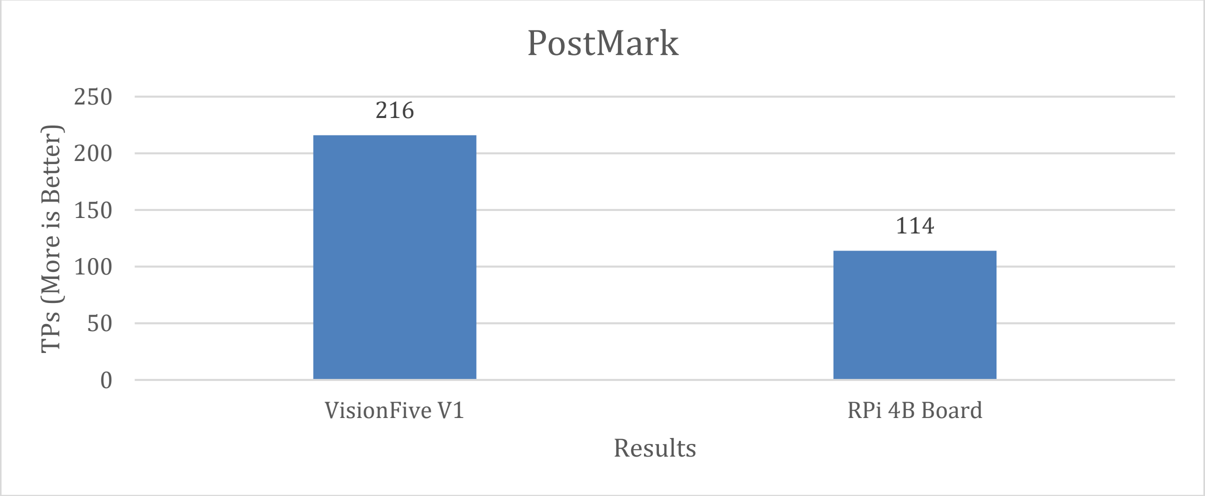


Figure 5.2.1.7: Bar chart of PostMark performance comparison for both boards.

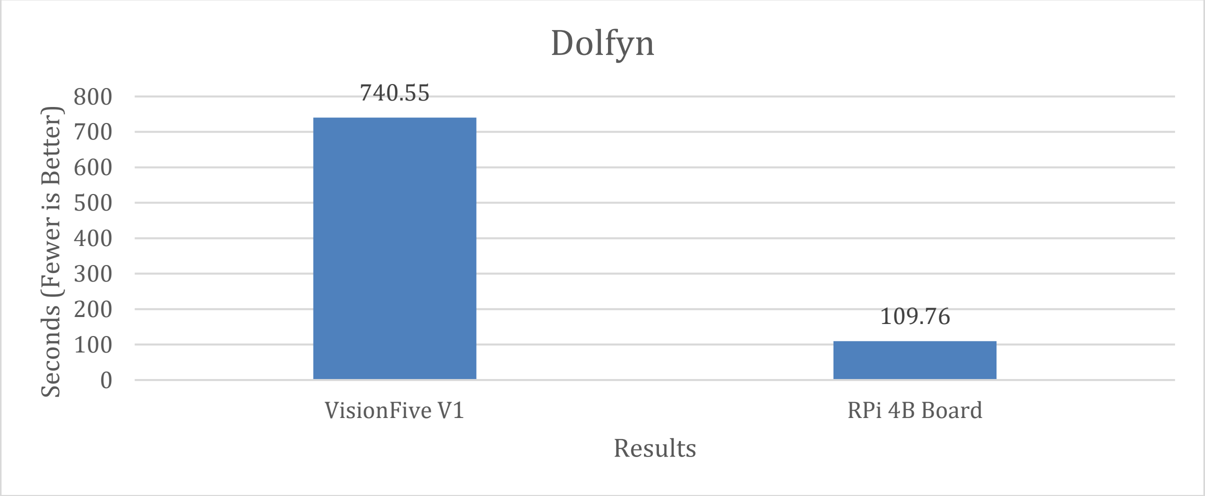


Figure 5.2.1.8: Bar chart of Dolfyn performance comparison for both boards.

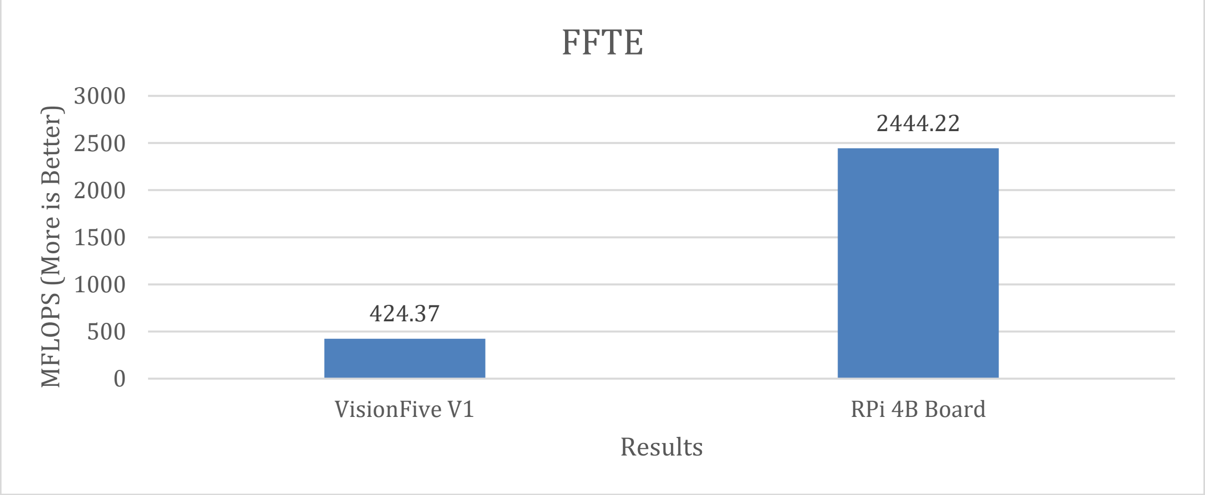


Figure 5.2.1.9: Bar chart of FFTE performance comparison for both boards.

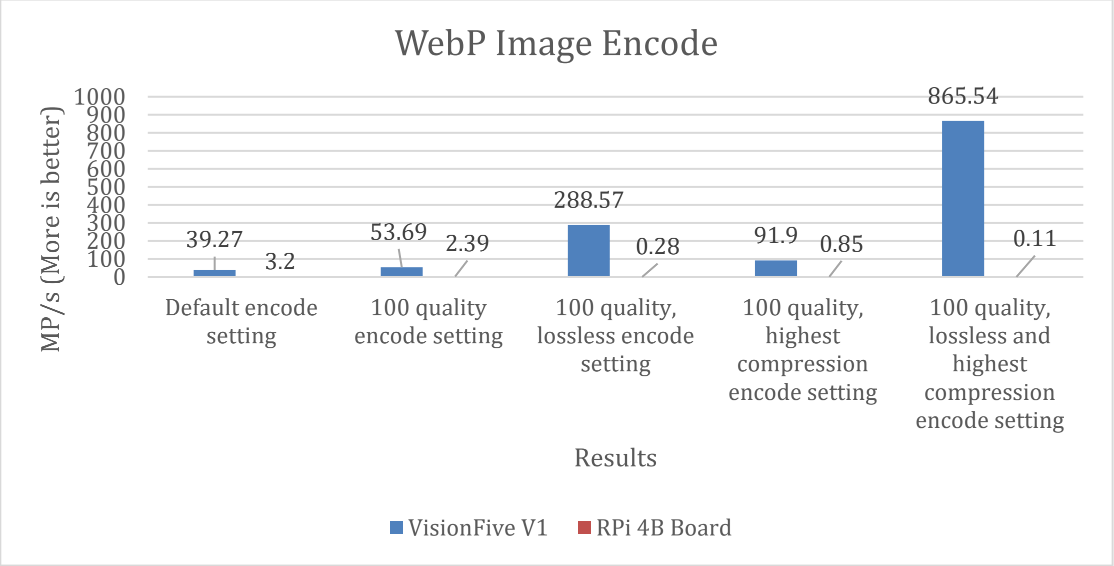


Figure 5.2.1.10: Bar chart of WebP Image Encode performance comparison for both boards.

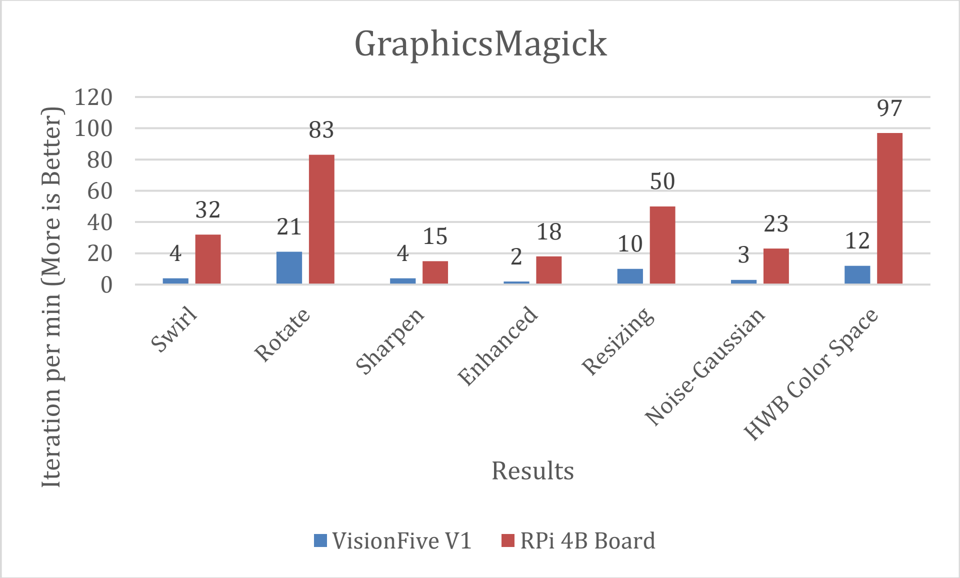


Figure 5.2.1.11: Bar chart of GraphicsMagick performance comparison for both boards.

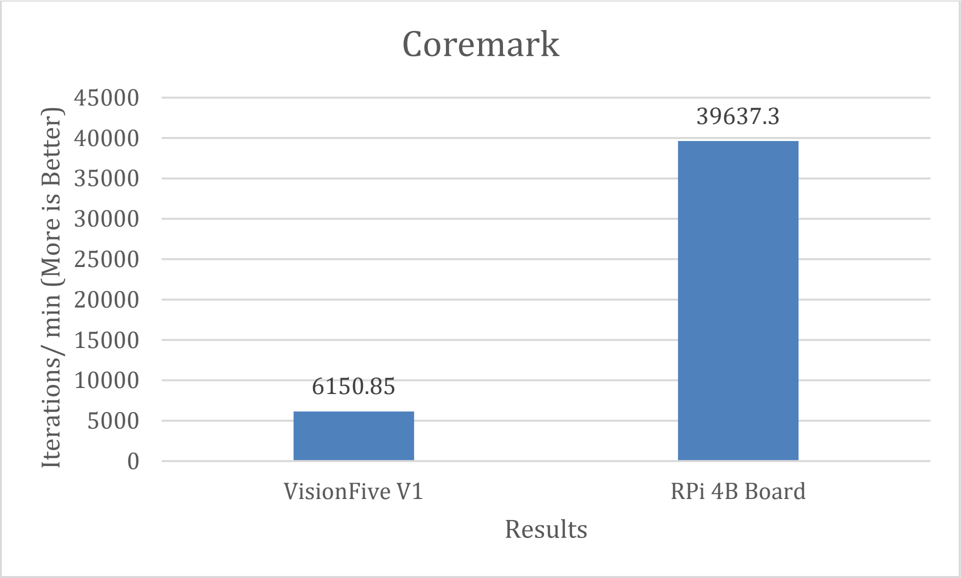


Figure 5.2.1.12: Bar chart of Coremark performance comparison for both boards.

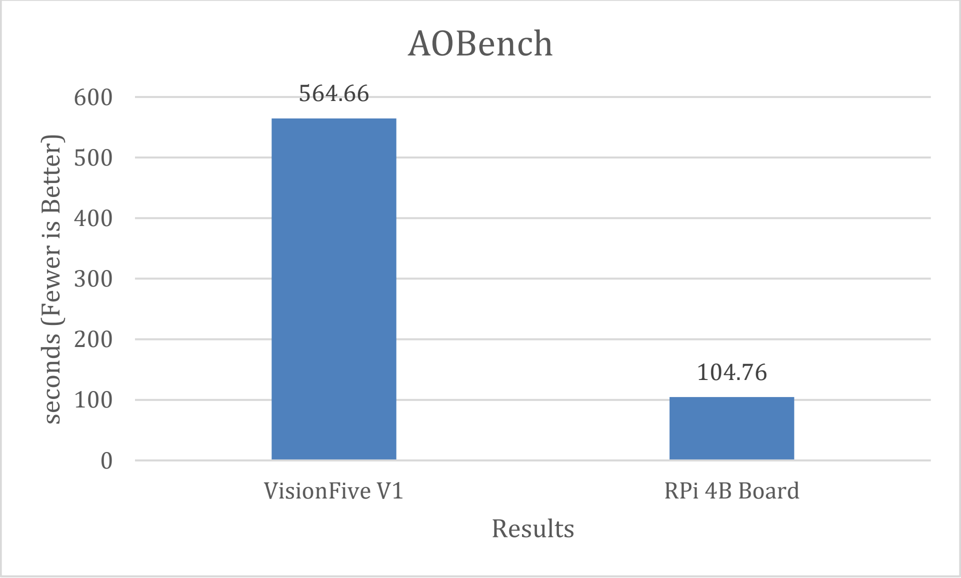


Figure 5.2.1. 13: Bar chart of AOBench performance comparison for both boards.

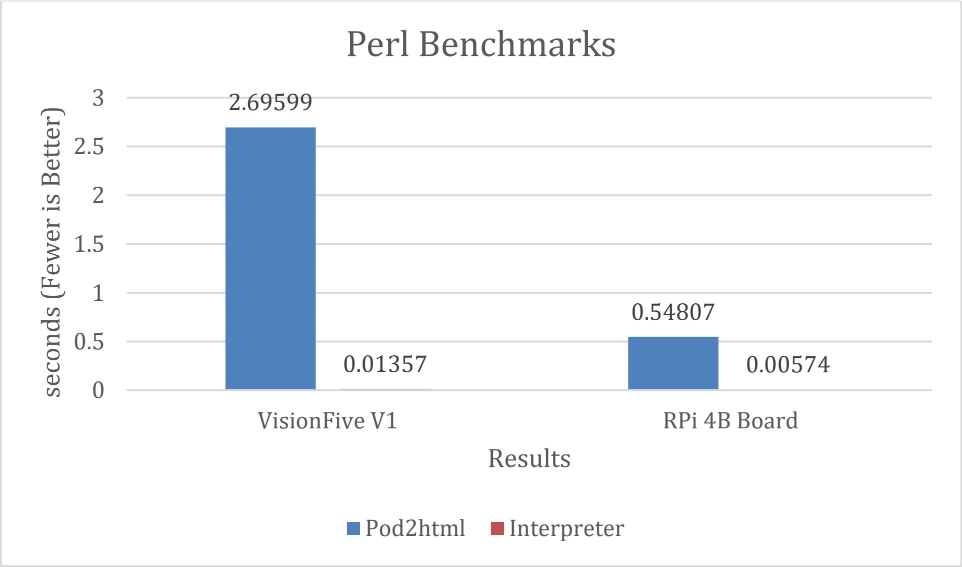


Figure 5.2.1.14: Bar chart of Perl Benchmarks performance comparison for both boards.

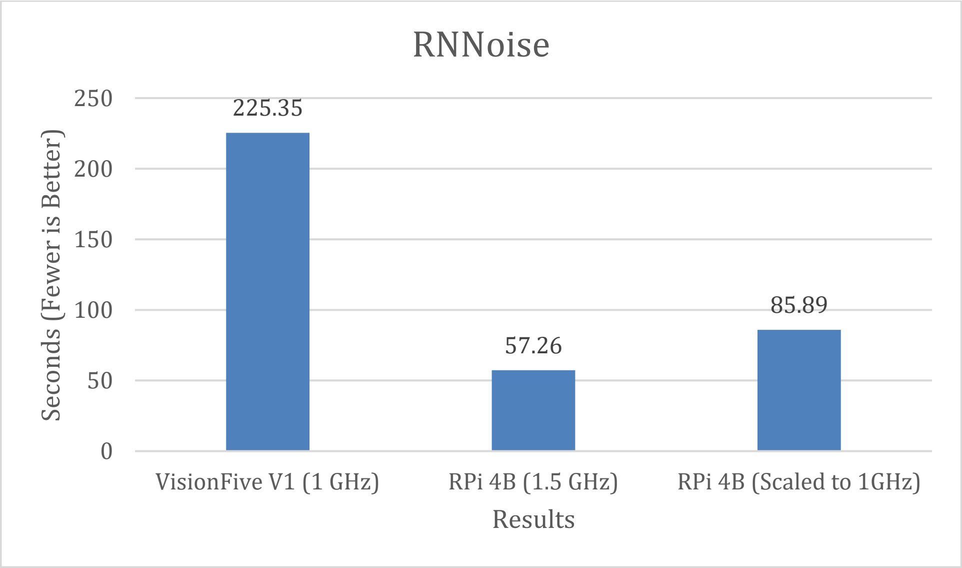


Figure 5.2.1.15: Bar chart of RNNoise performance comparison for both boards.

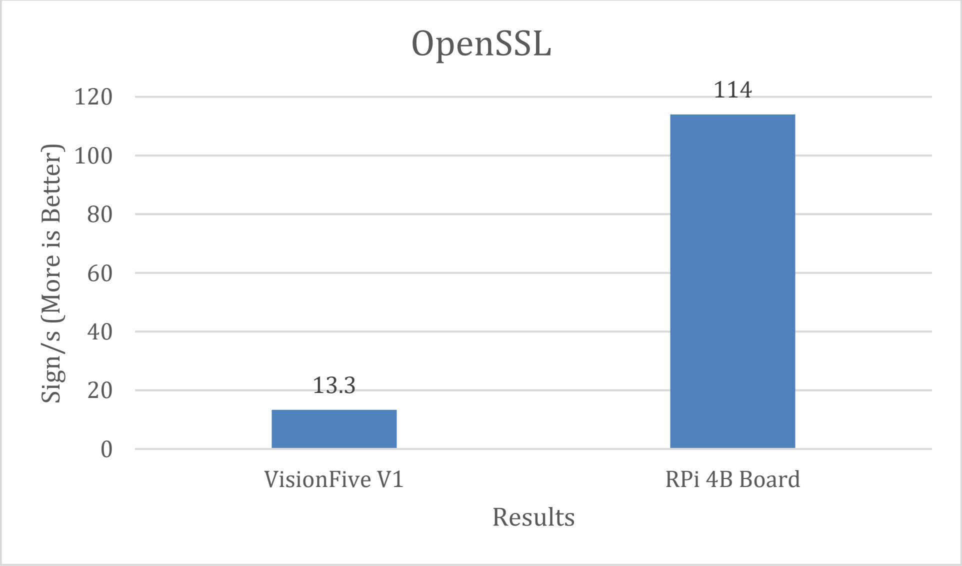


Figure 5.2.1.16: Bar chart of OpenSSL performance comparison for both boards.

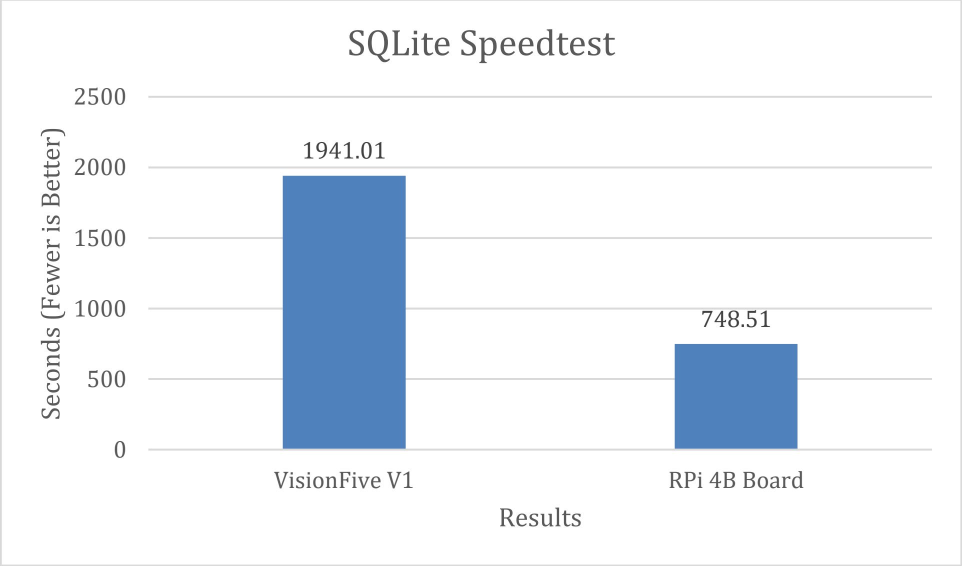


Figure 5.2.1.17: Bar chart of SQLite Speedtest performance comparison for both boards.

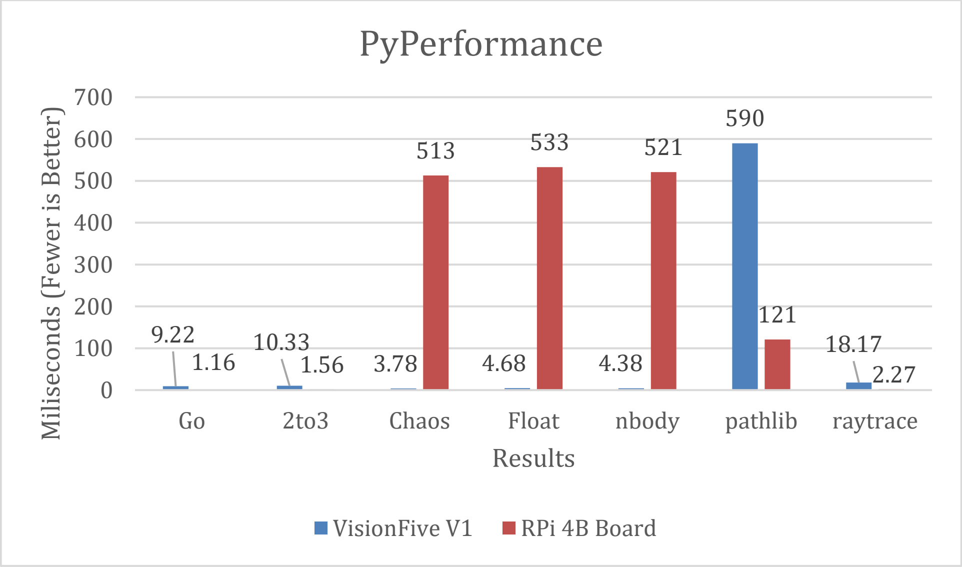


Figure 5.2.1.18: Bar chart of PyPerformance performance comparison for both boards.

(Part 1/2)

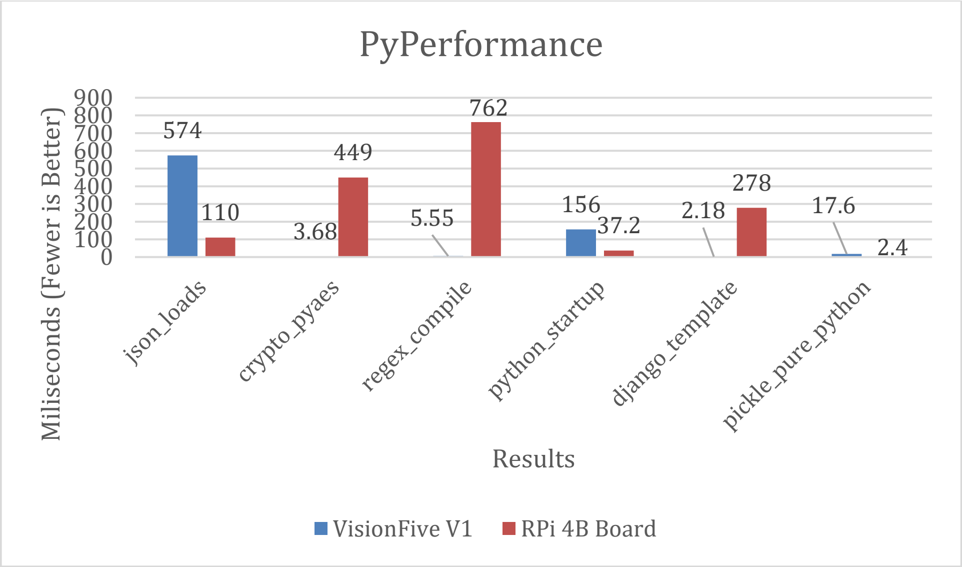


Figure 5.2.1.19: Bar chart of PyPerformance performance comparison for both boards. (Part 2/2)

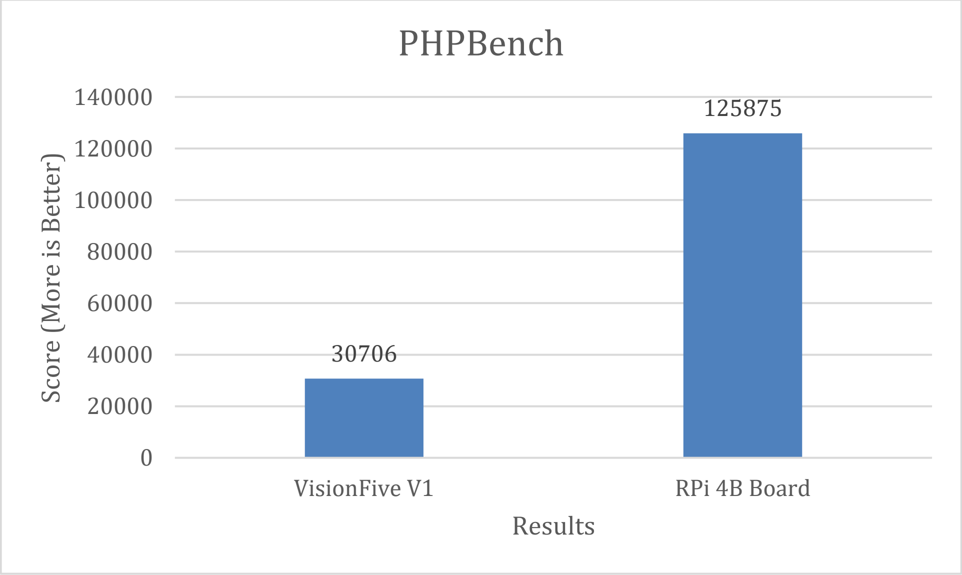


Figure 5.2.1.20: Bar chart of PHPBench performance comparison for both boards.

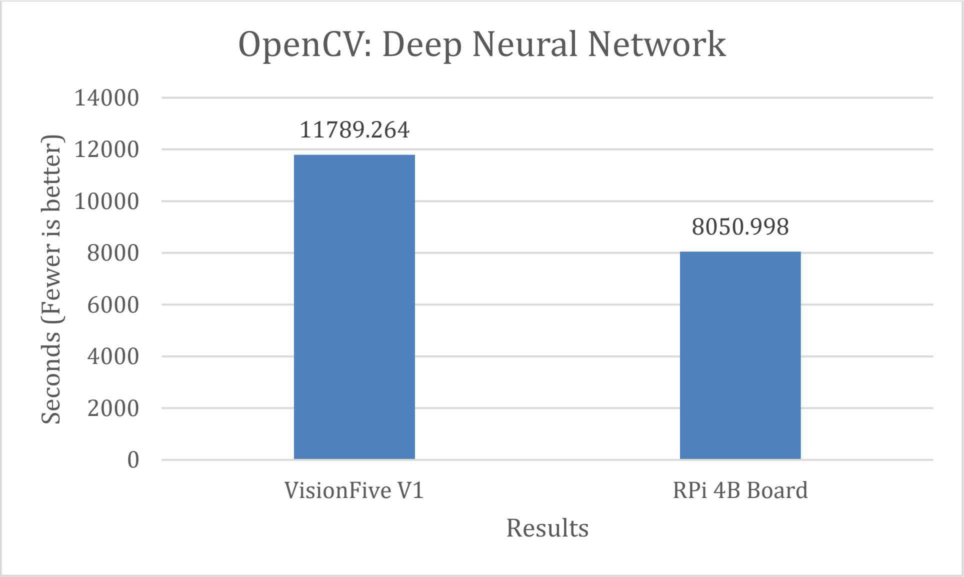


Figure 5.2.1.21: Bar chart of Deep Neural Network performance comparison for both boards.

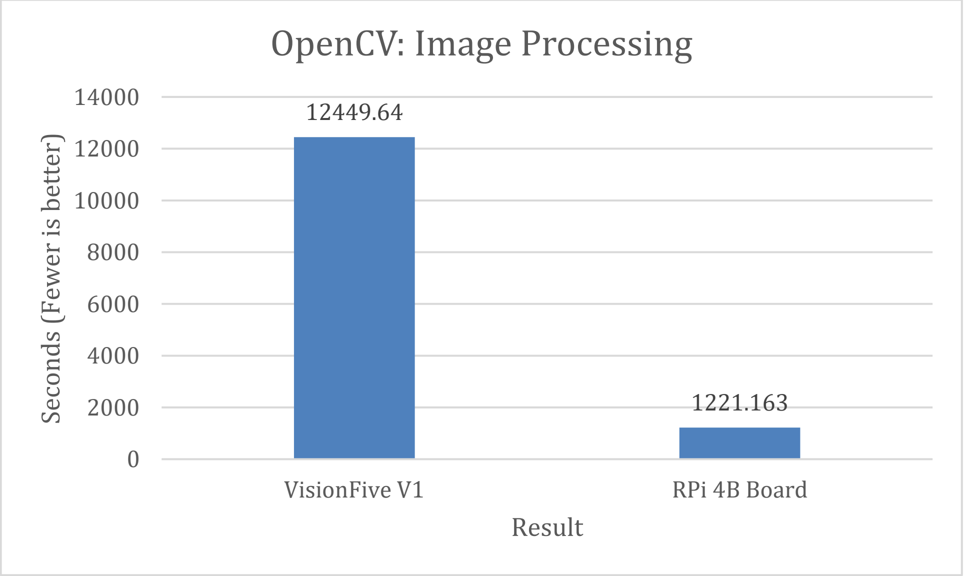


Figure 5.2.1.22: Bar chart of image processing performance comparison for both boards.

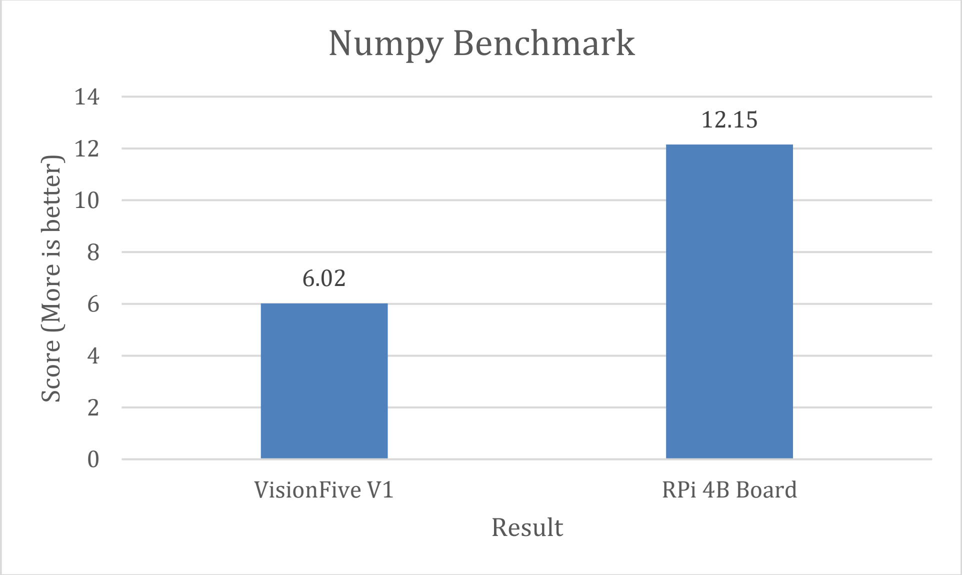


Figure 5.2.1.23: Bar chart of NumPy Benchmark performance comparison for both boards.

All tested result had been uploaded on openbenchmarking.org and it will be convenient for other developer to have a direct comparison from the website.

For VisionFive V1 tests result of Test 1 until Test 6, it can be seen from [21] . Tests result of Test 7 until Test 19, it can be obtained from [22] while test result of machine learning test which is Test 20 until Test 21 can be obtained from [23].

For RPi 4B model tests result of Test 1 until Test 19 can be retrieved from [24]. Lastly, the test result of Test 20 until Test 21 can be obtained from [25].

Other than that, all the proofs for tested result will be included in *Appendix A* section.

5.2.2 Thread Performance Test Details and Result

Table 5.2.2.1: Tests that had been carried out for both boards.

	Test	Test Details
1)	Lbzip2 compression performance	Dataset compression will be performed using lbzip2 by controlling the different number of threads.
2)	Pi calculation program	It is a customized micro-benchmark is used to measure the compute abilities by controlling the different number of threads.
3)	Power Consumption Test	Power consumed to complete a test will be recorded down. Besides, the power consumption IDLE and fully utilized processor will also be compared between VisionFive V1 Board and RPi 4B.

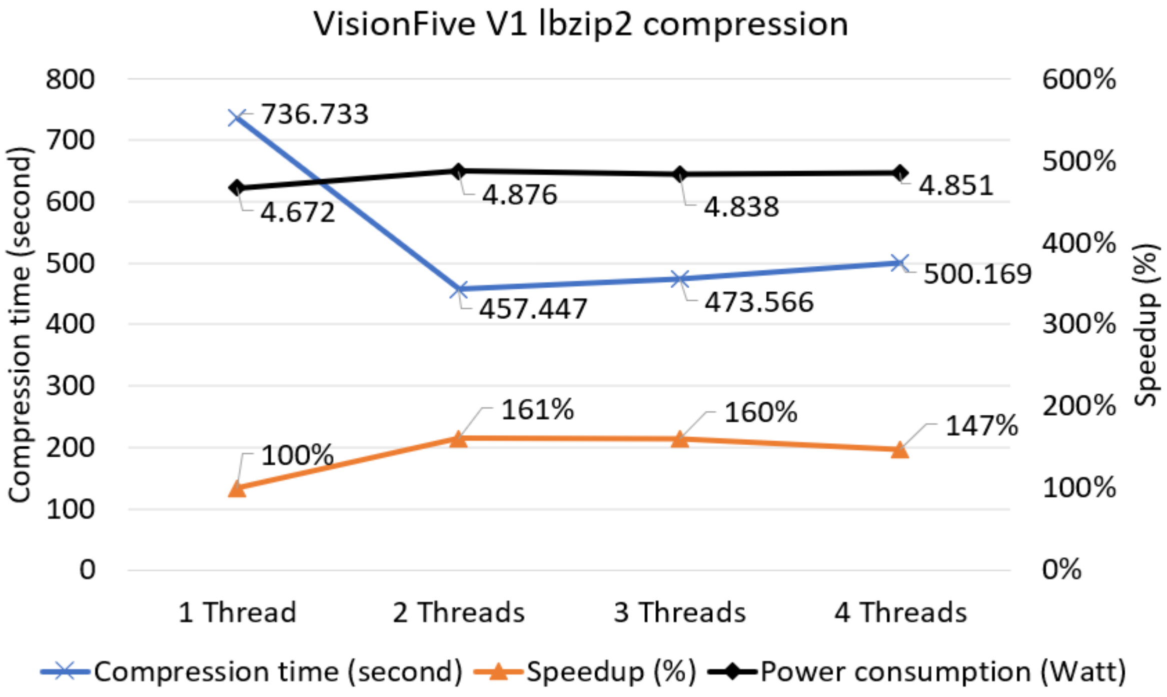


Figure 5.2.2.1: Results of dataset compression time, threads scalability in terms of speedup and power consumption by VisionFive V1 Board

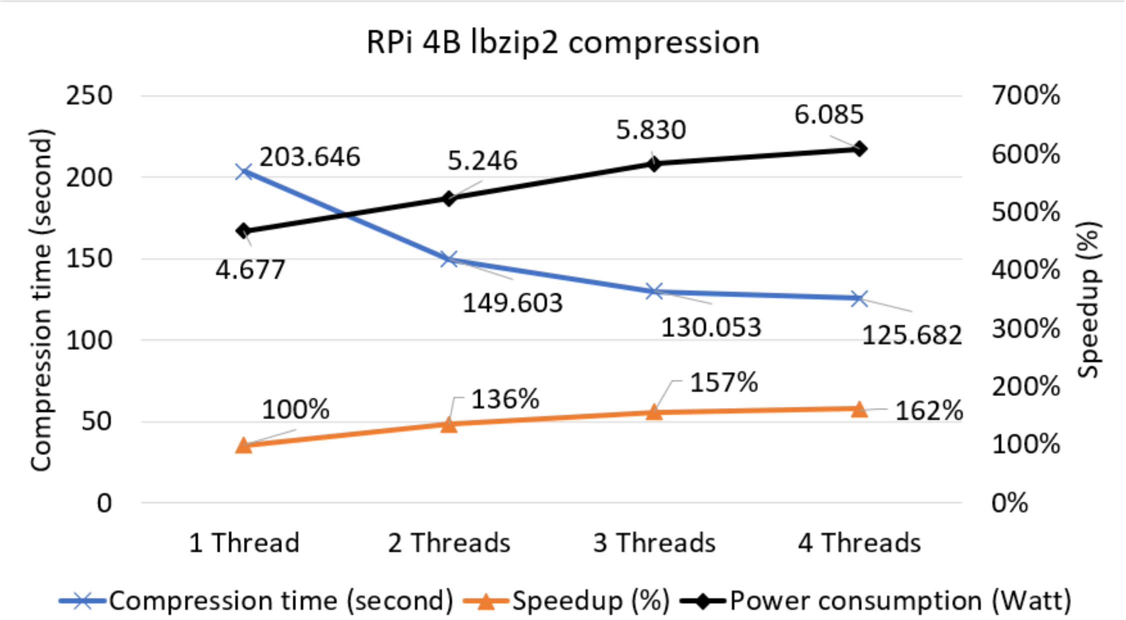


Figure 5.2.2.2: Results of dataset compression time, threads scalability in terms of speedup and power consumption by RPi 4B model.

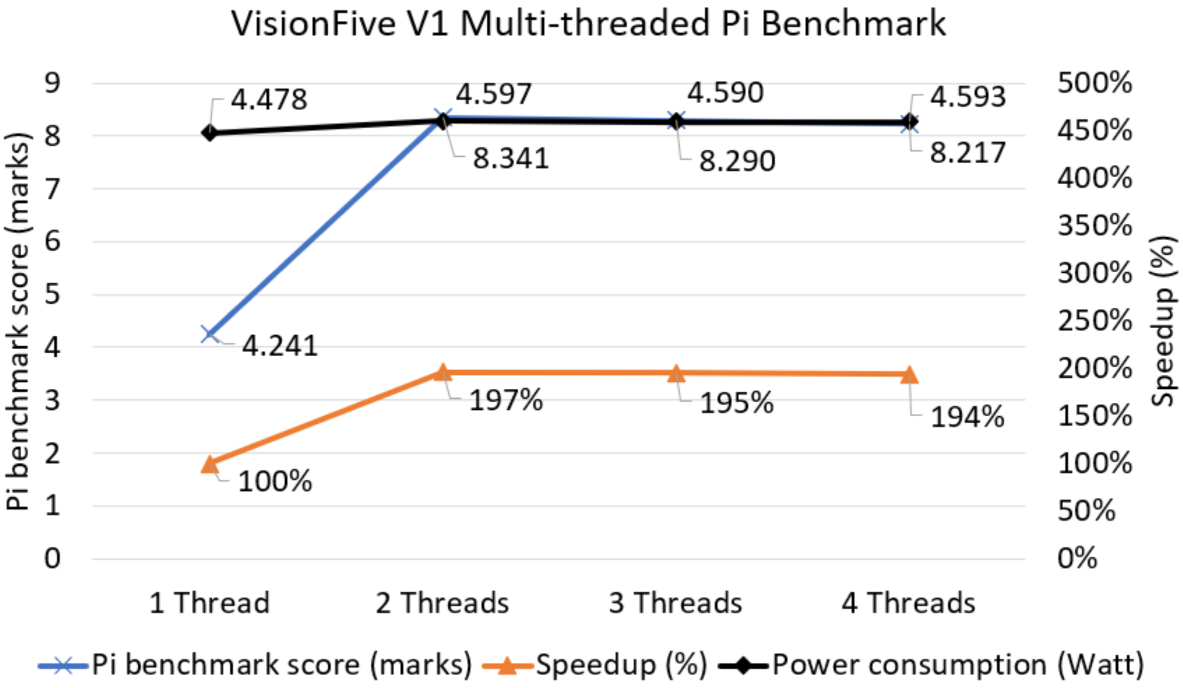


Figure 5.2.2.3: Results of Pi benchmark score, threads scalability in terms of speedup and power consumption by VisionFive V1 Board.

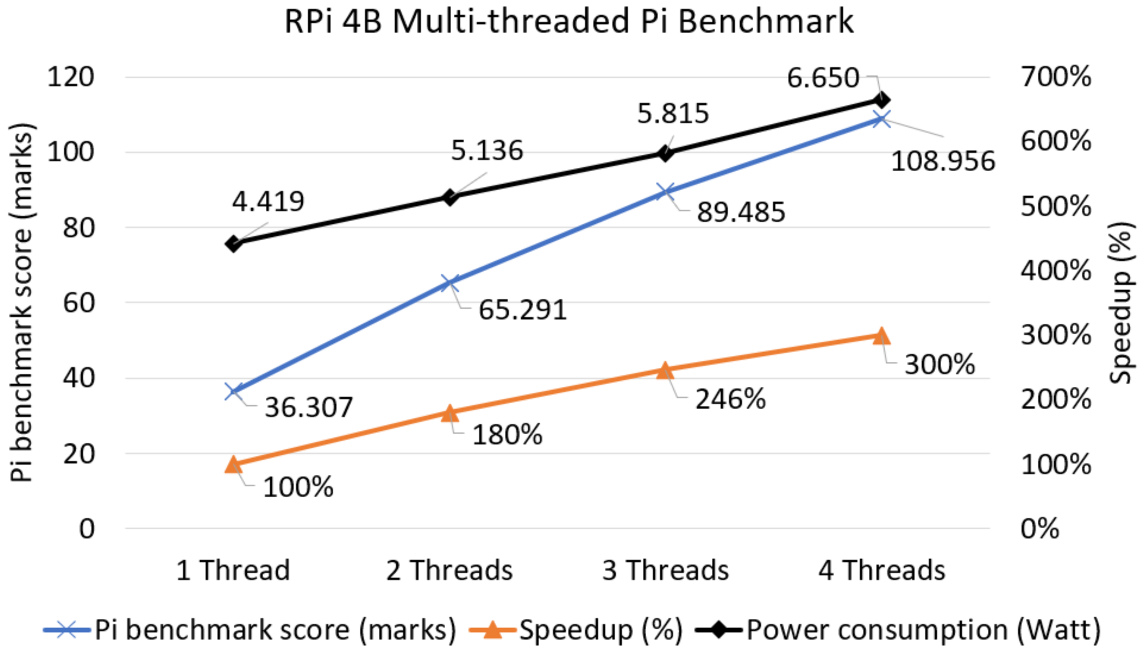


Figure 5.2.2.4: Results of Pi benchmark score, threads scalability in terms of speedup and power consumption by RPi 4B model.

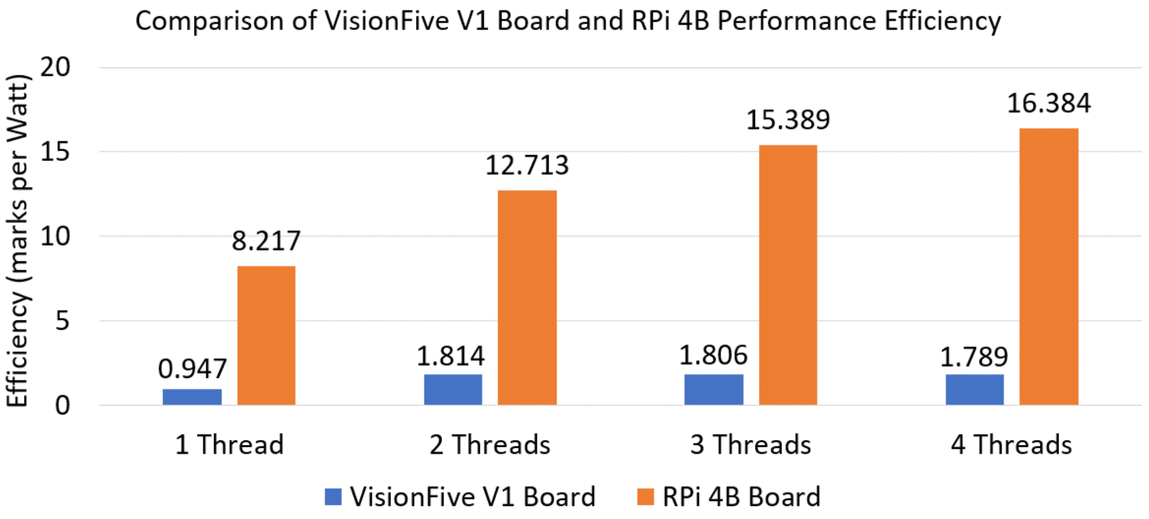


Figure 5.2.2.5: Bar chart of VisionFive V1 Board and RPi 4B performance efficiency in Pi benchmark score per Watt.

5.3 Performance Result Analysis

According to section 5.2.1, appropriate scaling and normalization were done for tests that involved CPU benchmarking. For single-threaded tests, such as FLAC Audio Encoding and RNNoise, a scaling of 1.5 times was used to decrease the RPi 4 Board benchmark score. This is because the VisionFive V1 Board only has a CPU frequency that runs at 1.0 GHz with 2 cores, while the RPi 4 has a CPU frequency that runs at a maximum speed of 1.50 GHz with 4 cores. Therefore, to simulate a fair environment for the VisionFive V1 Board, scaling and normalization were applied.

Additionally, the scalability of thread performance was also determined during the dataset compression test. According to Figure 5.2.2.1, the VisionFive V1 Board experienced a speedup of 161% when compressing the dataset using one thread to two threads. For the RPi 4B, it only experienced a 136% 2-thread speedup compared to the performance of 1 thread, 157% 3-thread speedup compared to the performance of 1 thread, and 162% 4-thread speedup compared to the performance of 1 thread. In short, the scalability of the RPi 4B is still lower than that of the VisionFive V1 Board. However, the time for the VisionFive V1 Board to complete the dataset compression is 3 times longer than the RPi 4B model. Therefore, the RPi 4B model generally has better performance, although it has a lesser speedup when changing to 2 threads compared to 1 thread.

Regarding multi-threaded Pi benchmark in Figure 5.2.2.3, the VisionFive V1 Board also produced a speedup of 197% when experiencing the number of threads from 1 to 2. For RPi 4B thread performance in Pi benchmark, it had a speedup of 180% when experiencing the number of threads from 1 to 2. However, due to the number of cores available in RPi 4B, it allowed the RPi 4B model to experience a speedup of 300% which is 4 thread performance compared to 1 thread. As the number of cores available for the VisionFive V1 Board is only 2 cores, this stopped the VisionFive V1 Board from experiencing further speedup produced by 3 threads and 4 threads.

In terms of power consumption, based on the figures in section 5.2.2, the VisionFive V1 Board consumes less power during full load, which is another advantage compared to RPi 4B. The energy consumed by the VisionFive V1 Board with 1 thread and 2 threads does not have a huge difference, which means that even though the VisionFive V1 Board is experiencing a significant speedup, it does not consume a lot more energy when the performance increases. On the other hand, the RPi 4B experiences a lesser speedup compared to the VisionFive V1 Board, but it requires higher power consumption when using 2 threads compared to 1 thread.

This further increases the cost of the RPi 4B having such good performance, and in return, the power consumption will be a little bit higher than the VisionFive V1 Board.

In conclusion, the RPi 4B outperformed the VisionFive V1 Board in compression and multi-threaded Pi benchmarks in terms of absolute performance and efficiency. Intriguingly, the VisionFive V1 SiFive U74 dual-core processor has better scalability in compute-bound Pi calculation. The VisionFive V1 Board also consumes less power during full load.

5.4 Project Challenges

One of the project found in this project are limited hardware resources available on the single board computer. These type of motherboard often have limited processing power, memory, and storage which affects them to produce a highly accurate benchmark results. In addition, this will also make it difficult to perform complex benchmarks such as machine learning tests that require a lot of resources.

Other than that, the software compatibility will also be one of the project challenges when benchmarking a single board computer. This is because some of the operating system or architecture is not compatible with the benchmarking tools or software. As an example, sysbench is not available to benchmark VisionFive V1 board while it is available for RPi 4B model. This is due to the instruction set architecture of VisionFive V1 board is not compatible with sysbench software.

Besides, there may be an environmental factor that can affect the benchmarking results such as temperature, power supply stability and many else. A single board computer may be shutdown due to the overheating especially when running a complex benchmark such as machine learning test. The processor will be fully utilized by the software or tests which will cause the board to have an overheating problem and they will have a higher risk to experience system shutdown during the test.

Lastly, the lack of a clear or well-documented hardware implementation manual for the VisionFive V1 board can pose challenges. This can limit the optimization opportunities and utilization of available resources, such as the vision DSP chip for machine learning. Without proper documentation, developers may miss out on leveraging the board's capabilities for their IoT product, resulting in wasted potential.

5.5 Objective Evaluation

1) To provide developers with a performance reference point when selecting a suitable single board computer for their project.
    - All the benchmark result had uploaded onto openbenchmarking.org website which enable any public to access it. This is helpful for developers to compare the performance score of VisionFive V1 board with other single board computer.

2) To compare and contrast the VisionFive V1 Board's performance with the current market leader, the Raspberry Pi model.
    - The benchmark result will be compared with RPi 4B model within the same environment and same setting.

3) To contribute to the documentation and knowledge base in the field of single board computers.
    - The result of benchmarking had documented into a report in this project. A conference paper about thread performance of VisionFive V1 board had also published for participating in IEEE conference.

CHAPTER 6 – Conclusion and Recommendation

6.1 Conclusion

In conclusion, this project was carried out to evaluate the performance of the VisionFive V1 Board, providing future developers with a reference. The project successfully addressed the issue of developers needing to know how well the VisionFive V1 Board performs. Furthermore, the advantages of the VisionFive V1 Board were identified, allowing developers to choose the most suitable application to develop using the board. Additionally, other designers will become aware of the advantages of the RISC-V architecture due to this project. With this project, there is no longer a need for other developers to spend time exploring the possibilities of the VisionFive V1 Board.

Based on the benchmarking result and comparison done, it can be concluded that the performance of the VisionFive V1 board, when compared to the RPi 4B model, is a mixed bag. In terms of absolute performance and efficiency, the RPi 4B model appears to outperform the VisionFive V1 board. This is evident in the CPU benchmarking tests such as FLAC Audio Encoding and RNNoise, where the RPi 4B's higher CPU frequency (up to 1.50 GHz with 4 cores) results in better performance compared to the VisionFive V1 board's lower CPU frequency (1.0 GHz with 2 cores). Additionally, the RPi 4B model shows better scalability in multi-threaded tests, with higher speedup percentages when increasing the number of threads.

However, it's worth noting that the VisionFive V1 board has shown better scalability in compute-bound Pi calculation, where it experienced a speedup of 161% when increasing the number of threads from 1 to 2, compared to the RPi 4B's speedup of 136%. This indicates that the VisionFive V1 board's dual-core SiFive U74 processor has better scalability in certain compute-intensive tasks, despite its lower clock frequency and limited number of cores.

Another notable advantage of the VisionFive V1 board is its power consumption during full load. According to the data presented in Section 5.2.2, the VisionFive V1 board consumes less energy compared to the RPi 4B model, particularly when comparing the power consumption of 1 thread and 2 threads. This means that even though the VisionFive V1 board experiences significant speedup, its energy consumption remains relatively low, indicating better power efficiency compared to the RPi 4B model, which requires higher power consumption when using multiple threads.

However, it's important to acknowledge the limitations of the VisionFive V1 board. The board's dual-core processor, lower clock frequency, and limited memory and storage may limit its ability to perform complex workload, such as machine learning tests, that require significant computational resources. Additionally, the lack of clear and well-documented hardware implementation manuals and potential environmental factors may further affect the board's performance and benchmarking results.

6.2 Recommendation

In future works, more various flag optimization setting will be tested on VisionFive V1 board as the open-source property is one of the selling point for RISCV architecture single board computer. Due to the time constraint and environmental factor, the machine learning test is less conducted in this project. Therefore, it is recommended to perform more machine learning test in different aspect such as mobile neural network, TensorFlow benchmark, LwHBench benchmark and many else in future. Other than that, High-Performance Linpack (HPL) Benchmark can also be performed on both single board computer as it is a quite standardized benchmark to be performed by any distributed computer.

REFERENCES

[1] MiniTool, "Introduction to Single Board Computer (SBC)," MiniTool, 2020. [Online]. Available: https://www.minitool.com/lib/single-board-computer.html. [Accessed 28 November 2022].

[2] J. Tranter, "What is RISC-V and Why is it Important?," ICS, 12 May 2021. [Online]. Available: https://www.ics.com/blog/what-risc-v-and-why-it-important. [Accessed 28 November 2022].

[3] E. Corpeno, "An Introduction to RISC-V—Understanding RISC's Open ISA," All About Circuits, 12 June 2022. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/introductions-to-risc-v-instruction-set-understanding-this-open-instruction-set-architecture/. [Accessed 20 November 2022].

[4] W. N. K. G. Mehrdad Poorhosseini, "A Compiler Comparison in the RISC-V Ecosystem," *2020 International Conference on Omni-layer Intelligent Systems (COINS),* pp. 1-6, 2020.

[5] "Introduction to Phoronix Test Suite," ORDINATECHNIC, 18 February 2018. [Online]. Available: https://www.ordinatechnic.com/general-guides/1/Software/2/Benchmark/3/Phoronix%20Test%20Suite/introduction-to-phoronix-test-suite. [Accessed 20 November 2022].

[6] J. M. J. V. A. H. C. G. B. G. P. a. G. M. P. Pedro Miguel Sánchez Sánchez, "LwHBench: A low-level hardware component benchmark and dataset for Single Board Computers," *Journal Of Latex Class Files,* vol. 14, no. 8, 2015.

[7] S. H. Eric Gamess, "Performance Evaluation of Different Raspberry Pi Models for a Broad Spectrum of Interests," *International Journal of Advanced Computer Science and Applications,* vol. 13, no. 2, pp. 819-829, 2022.

[8] S. Hymel, "Single Board Computer Benchmarks," SparkFun Electronics, [Online]. Available: https://learn.sparkfun.com/tutorials/single-board-computer-benchmarks. [Accessed 29 August 2022].

[9] J. Shepard, "RISC-V vs. ARM vs. x86 – What's the difference?," WTWH Media LLC, 13 April 2021. [Online]. Available: https://www.microcontrollertips.com/risc-v-vs-arm-vs-x86-whats-the-difference/. [Accessed 29 August 2022].

[10] S. Technology, "StarFive Technology," Shanghai StarFive Technology, 2022. [Online]. Available: https://www.starfivetech.com/en. [Accessed 29 August 2022].

[11] Raspberry Pi, "Raspberry Pi 4 Tech Specs," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/. [Accessed 1 December 2022].

[12] T. George, "Exploratory Research | Definition, Guide, & Examples," Scribbr, 6 December 2021. [Online]. Available: https://www.scribbr.com/methodology/exploratory-research/. [Accessed 7 May 2022].

[13] OpenBenchmarking, "OpenBenchmarking," OpenBenchmarking, [Online]. Available: https://openbenchmarking.org/. [Accessed 1 September 2022].

[14] "PuTTY," Techopedia, [Online]. Available: https://www.techopedia.com/definition/4335/putty. [Accessed 1 September 2022].

[15] "Introducing WinSCP," WinSCP, [Online]. Available: https://winscp.net/eng/docs/introduction#:~:text=WinSCP%20is%20an%20open%20source,and%20basic%20file%20manager%20functionality.. [Accessed 1 September 2022].

[16] Simplilearn, "Use of C Language," Simplilearn, 15 Mar 2022. [Online]. Available: https://www.simplilearn.com/tutorials/c-tutorial/use-of-c-language#:~:text=C%20programming%20language%20is%20a,foundation%20in%20the%20process%20of. [Accessed 1 September 2022].

[17] B. Thompson, "What is C++? Basic Concepts of C++ Programming Language," Guru99, 25 August 2022. [Online]. Available: https://www.guru99.com/cpp-tutorial.html. [Accessed 1 September 2022].

[18] "Top 10 Reasons Why Python is So Popular With Developers in 2022," upGrad, 10 January 2021. [Online]. Available: https://www.upgrad.com/blog/reasons-why-python-popular-with-developers/#:~:text=The%20python%20language%20is%20one,faster%20than%20other%20programming%20languages.. [Accessed 1 September 2022].

[19] S. Emms, "Best Linux Multi-Core Compression Tools," 9 January 2019. [Online]. Available: https://www.linuxlinks.com/best-linux-multi-core-compression-tools/2/. [Accessed 26 April 2023].

[20] I.K.T. Tan, C.S. Wong, J.W. Lam, and R.D. Kumari, "Measuring Operating Systems' Tasks Fairness for CPU Resource Scheduling," in *IASTED International Conference on Advances Computer Science and Technology*, Langkawi, Malaysia, 2008.

[21] "kam," OpenBenchmarking.org, 5 November 2022. [Online]. Available: https://openbenchmarking.org/result/2211059-BE-KAM45144676. [Accessed 1 December 2022].

[22] "kam," OpenBenchmarking.org, 24 November 2022. [Online]. Available: https://openbenchmarking.org/result/2211241-BE-KAM72440396. [Accessed 1 December 2022].

[23] "kam," OpenBenchmarking.org, 20 April 2023. [Online]. Available: https://openbenchmarking.org/result/2304246-NE-KAM01029328. [Accessed 26 April 2023].

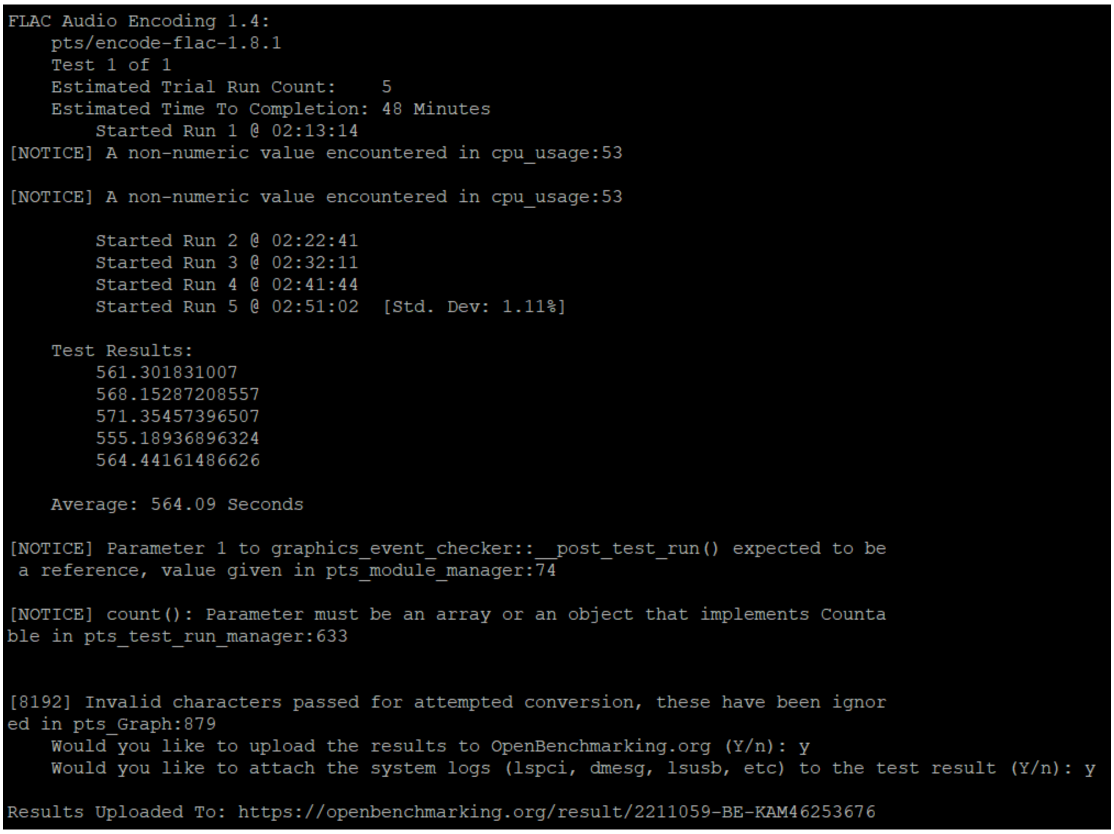
[24] "rpi-4," OpenBenchmarking.org, 19 January 2023. [Online]. Available: https://openbenchmarking.org/result/2212124-NE-RPI41642090. [Accessed 26 April 2023].

[25] "kam," OpenBenchmarking.org, 20 April 2023. [Online]. Available: https://openbenchmarking.org/result/2304241-BE-KAM05475328. [Accessed 26 April 2023].

[26] "Raspberry Pi 4 Benchmarks," OpenBenchmarking.org, 18 October 2020. [Online]. Available: https://openbenchmarking.org/result/2010224-FI-RASPBERRY80. [Accessed 1 December 2022].

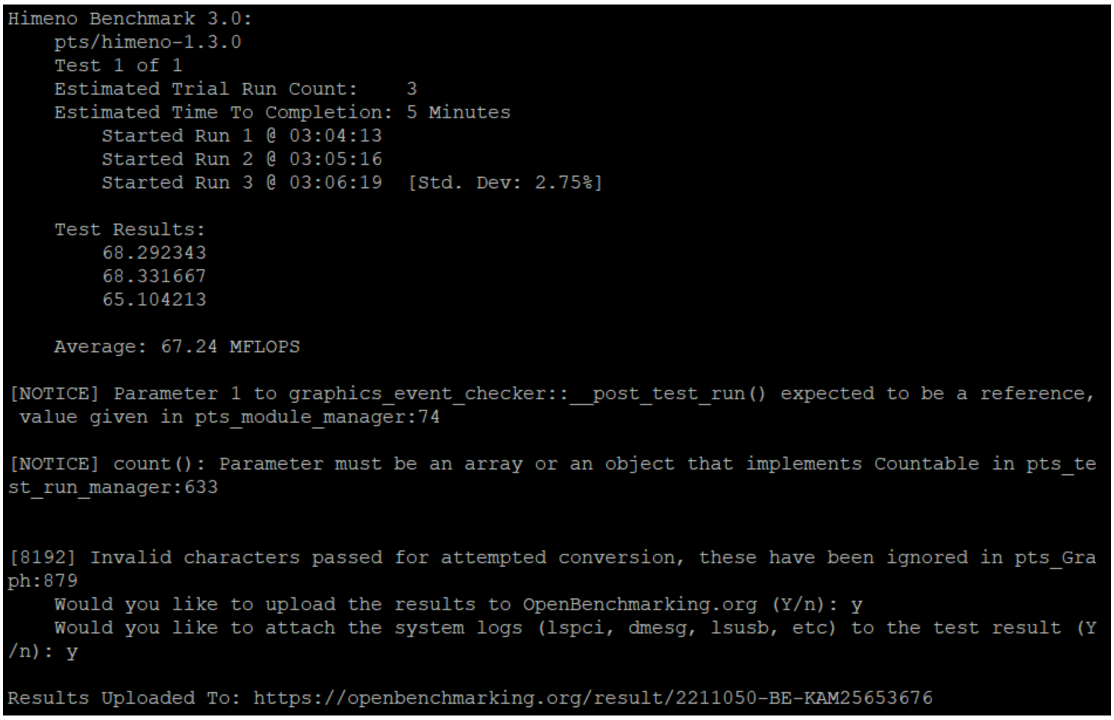
APPENDIX A – RESULT OF PHORONIX TEST SUITE TESTS

A1: FLAC Audio Encoding



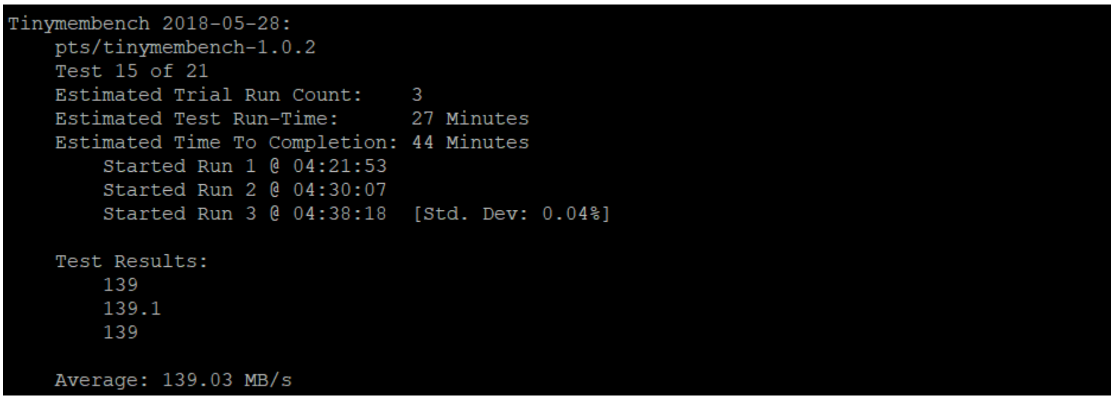
```
FLAC Audio Encoding 1.4:
    pts/encode-flac-1.8.1
    Test 1 of 1
    Estimated Trial Run Count:    5
    Estimated Time To Completion: 48 Minutes
        Started Run 1 @ 02:13:14
[NOTICE] A non-numeric value encountered in cpu_usage:53

[NOTICE] A non-numeric value encountered in cpu_usage:53

        Started Run 2 @ 02:22:41
        Started Run 3 @ 02:32:11
        Started Run 4 @ 02:41:44
        Started Run 5 @ 02:51:02  [Std. Dev: 1.11%]

    Test Results:
        561.301831007
        568.15287208557
        571.35457396507
        555.18936896324
        564.44161486626

    Average: 564.09 Seconds

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be
 a reference, value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Counta
ble in pts_test_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignor
ed in pts_Graph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211059-BE-KAM46253676
```

A2: Himeno Benchmark



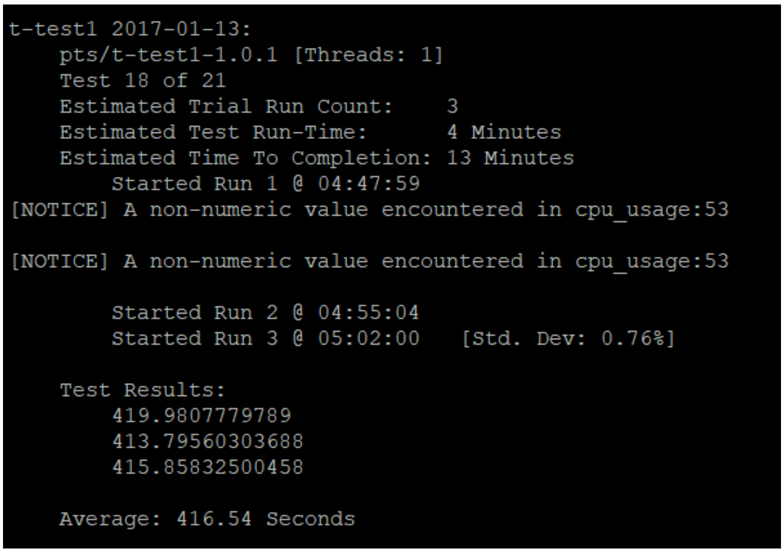
```
Himeno Benchmark 3.0:
    pts/himeno-1.3.0
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 5 Minutes
        Started Run 1 @ 03:04:13
        Started Run 2 @ 03:05:16
        Started Run 3 @ 03:06:19  [Std. Dev: 2.75%]

    Test Results:
        68.292343
        68.331667
        65.104213

    Average: 67.24 MFLOPS

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211050-BE-KAM25653676
```

A3: Tinymembench

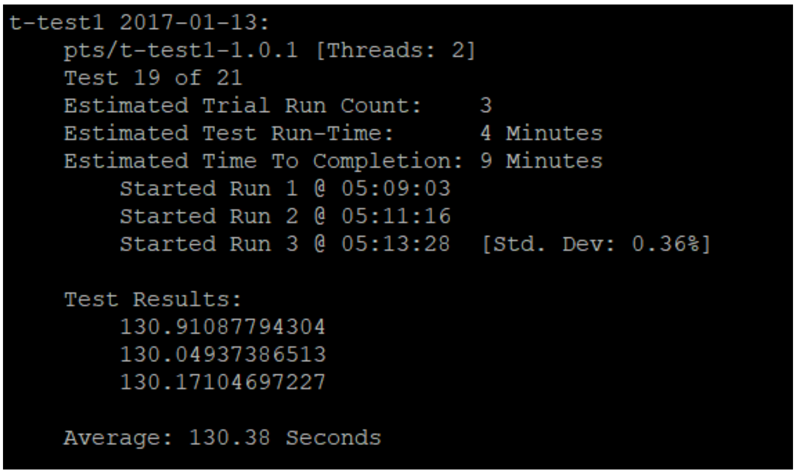


```
Tinymembench 2018-05-28:
    pts/tinymembench-1.0.2
    Test 15 of 21
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      27 Minutes
    Estimated Time To Completion: 44 Minutes
        Started Run 1 @ 04:21:53
        Started Run 2 @ 04:30:07
        Started Run 3 @ 04:38:18  [Std. Dev: 0.04%]

    Test Results:
        139
        139.1
        139

    Average: 139.03 MB/s
```

A4: t-test1

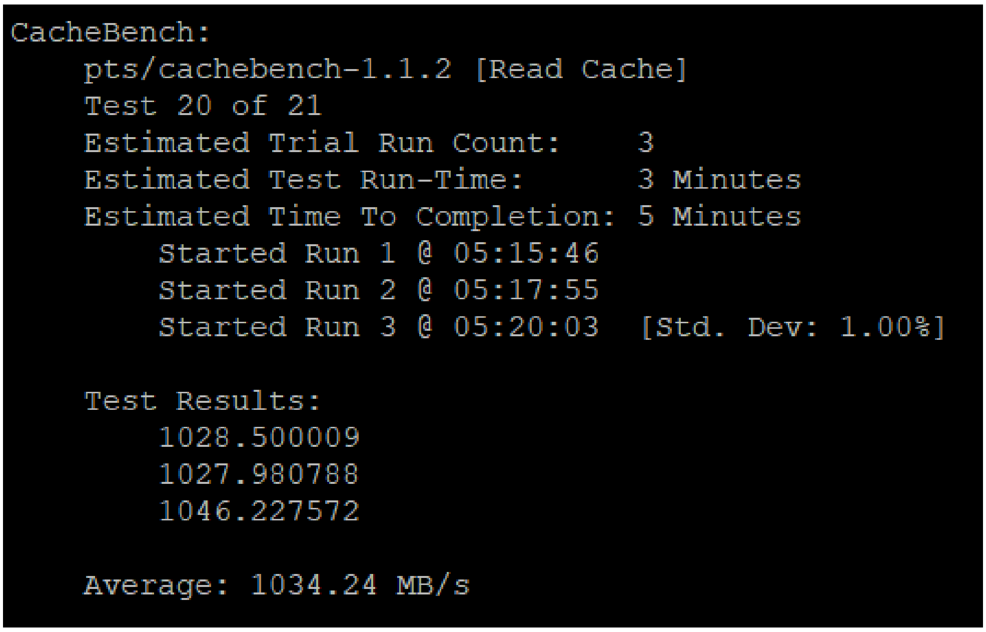


```
t-test1 2017-01-13:
    pts/t-test1-1.0.1 [Threads: 1]
    Test 18 of 21
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 13 Minutes
        Started Run 1 @ 04:47:59
[NOTICE] A non-numeric value encountered in cpu_usage:53

[NOTICE] A non-numeric value encountered in cpu_usage:53

        Started Run 2 @ 04:55:04
        Started Run 3 @ 05:02:00   [Std. Dev: 0.76%]

    Test Results:
        419.9807779789
        413.79560303688
        415.85832500458

    Average: 416.54 Seconds
```

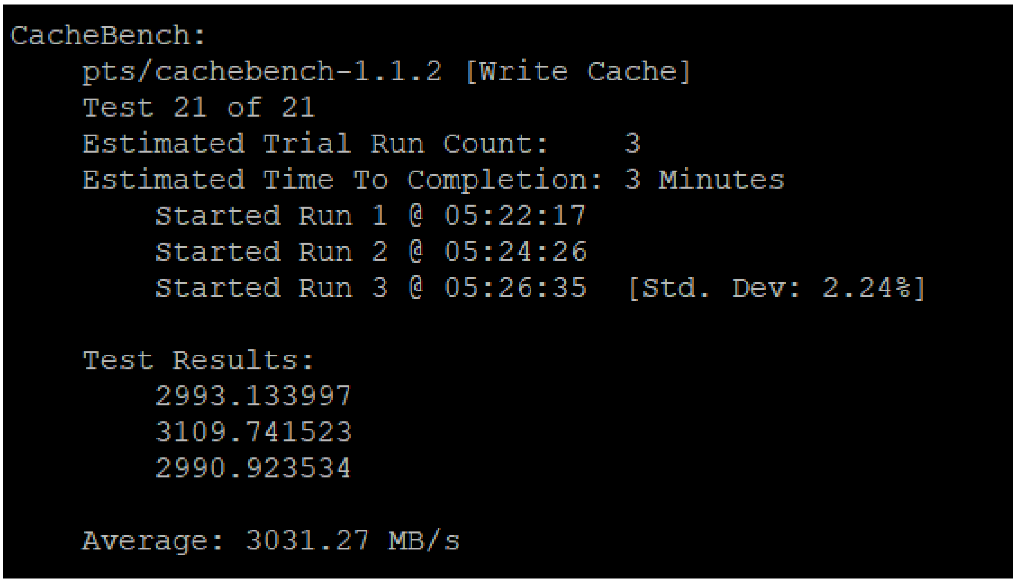


```
t-test1 2017-01-13:
    pts/t-test1-1.0.1 [Threads: 2]
    Test 19 of 21
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 9 Minutes
        Started Run 1 @ 05:09:03
        Started Run 2 @ 05:11:16
        Started Run 3 @ 05:13:28  [Std. Dev: 0.36%]

    Test Results:
        130.91087794304
        130.04937386513
        130.17104697227

    Average: 130.38 Seconds
```

A5: CacheBench

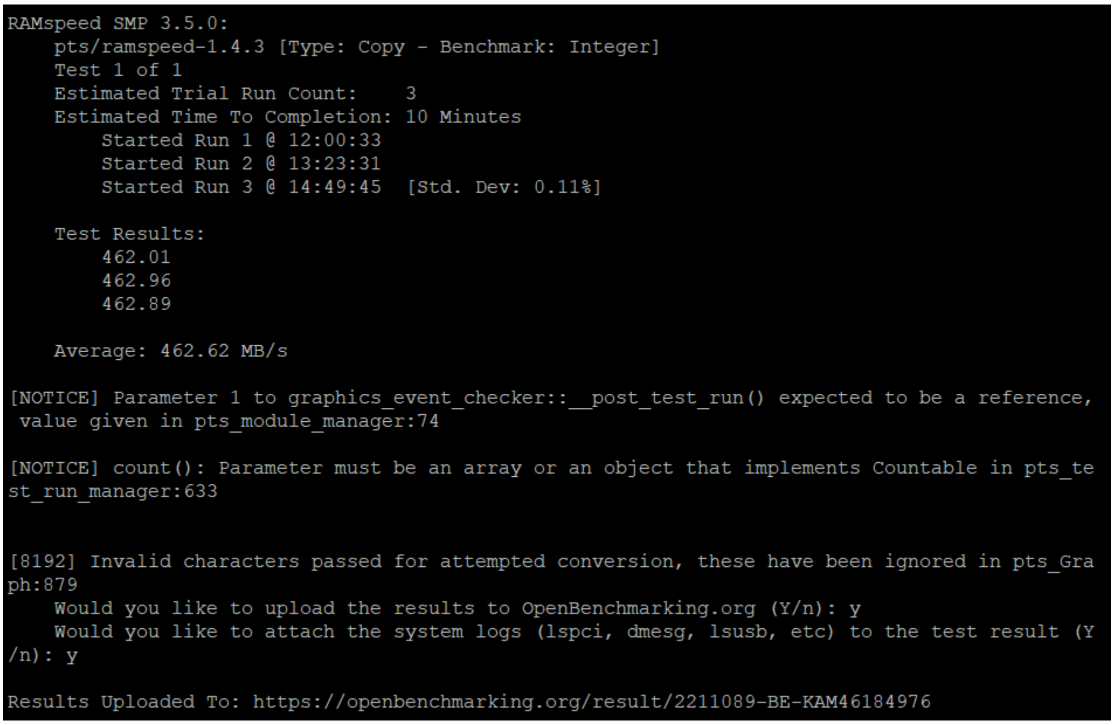


```
CacheBench:
    pts/cachebench-1.1.2 [Read Cache]
    Test 20 of 21
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      3 Minutes
    Estimated Time To Completion: 5 Minutes
        Started Run 1 @ 05:15:46
        Started Run 2 @ 05:17:55
        Started Run 3 @ 05:20:03  [Std. Dev: 1.00%]

    Test Results:
        1028.500009
        1027.980788
        1046.227572

    Average: 1034.24 MB/s
```



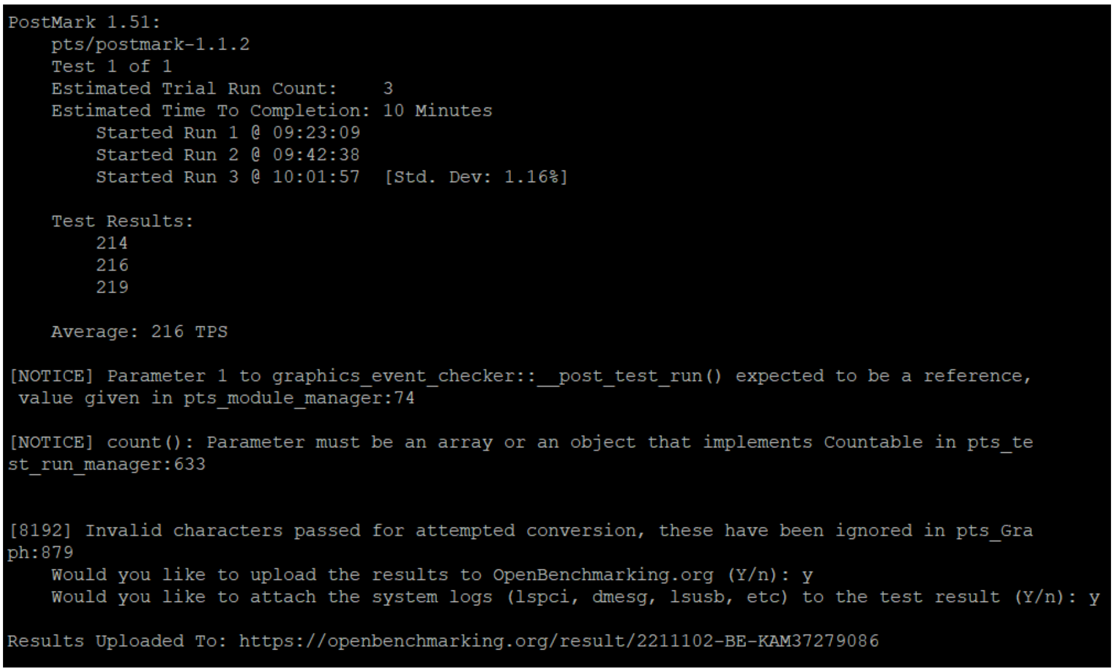
```
CacheBench:
    pts/cachebench-1.1.2 [Write Cache]
    Test 21 of 21
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 3 Minutes
        Started Run 1 @ 05:22:17
        Started Run 2 @ 05:24:26
        Started Run 3 @ 05:26:35  [Std. Dev: 2.24%]

    Test Results:
        2993.133997
        3109.741523
        2990.923534

    Average: 3031.27 MB/s
```

A6: RAMspeed SMP



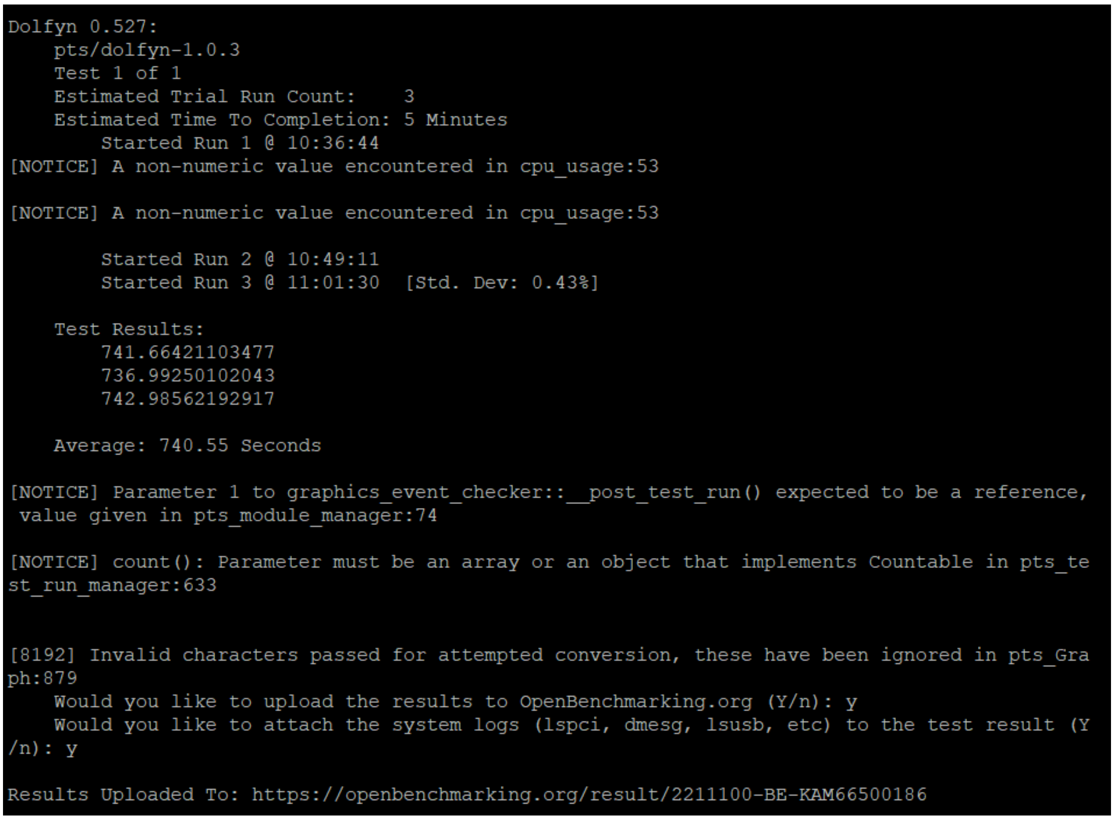
```
RAMspeed SMP 3.5.0:
    pts/ramspeed-1.4.3 [Type: Copy - Benchmark: Integer]
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 10 Minutes
        Started Run 1 @ 12:00:33
        Started Run 2 @ 13:23:31
        Started Run 3 @ 14:49:45  [Std. Dev: 0.11%]

    Test Results:
        462.01
        462.96
        462.89

    Average: 462.62 MB/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211089-BE-KAM46184976
```

A7: PostMark



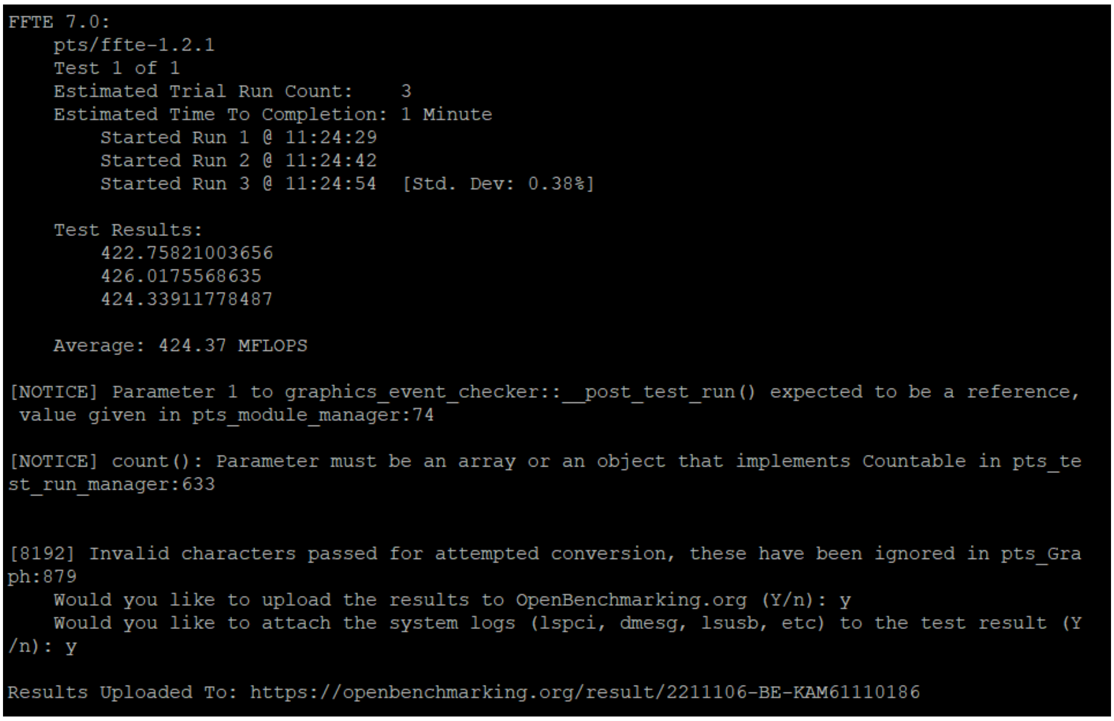
```
PostMark 1.51:
    pts/postmark-1.1.2
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 10 Minutes
        Started Run 1 @ 09:23:09
        Started Run 2 @ 09:42:38
        Started Run 3 @ 10:01:57  [Std. Dev: 1.16%]

    Test Results:
        214
        216
        219

    Average: 216 TPS

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211102-BE-KAM37279086
```

A8: Dolfyn



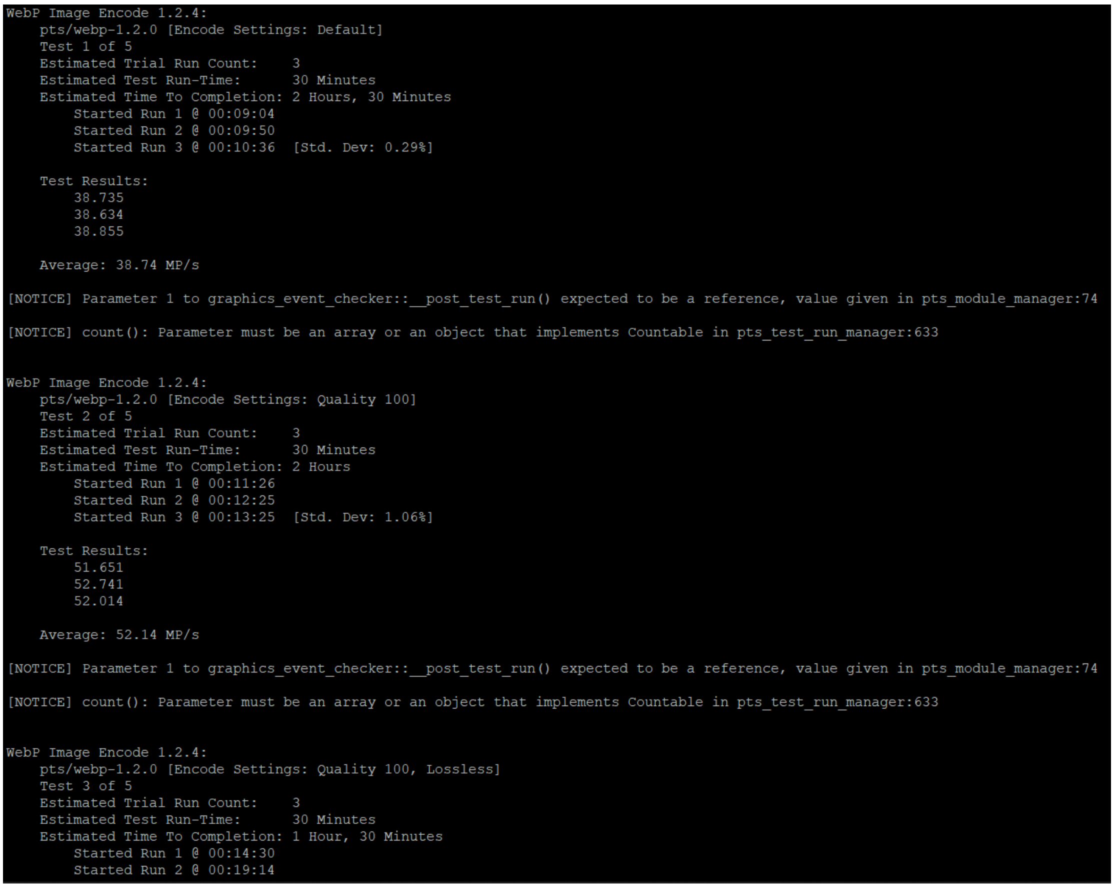
```
Dolfyn 0.527:
    pts/dolfyn-1.0.3
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 5 Minutes
        Started Run 1 @ 10:36:44
[NOTICE] A non-numeric value encountered in cpu_usage:53

[NOTICE] A non-numeric value encountered in cpu_usage:53

        Started Run 2 @ 10:49:11
        Started Run 3 @ 11:01:30  [Std. Dev: 0.43%]

    Test Results:
        741.66421103477
        736.99250102043
        742.98562192917

    Average: 740.55 Seconds

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211100-BE-KAM66500186
```

A9: FFTE

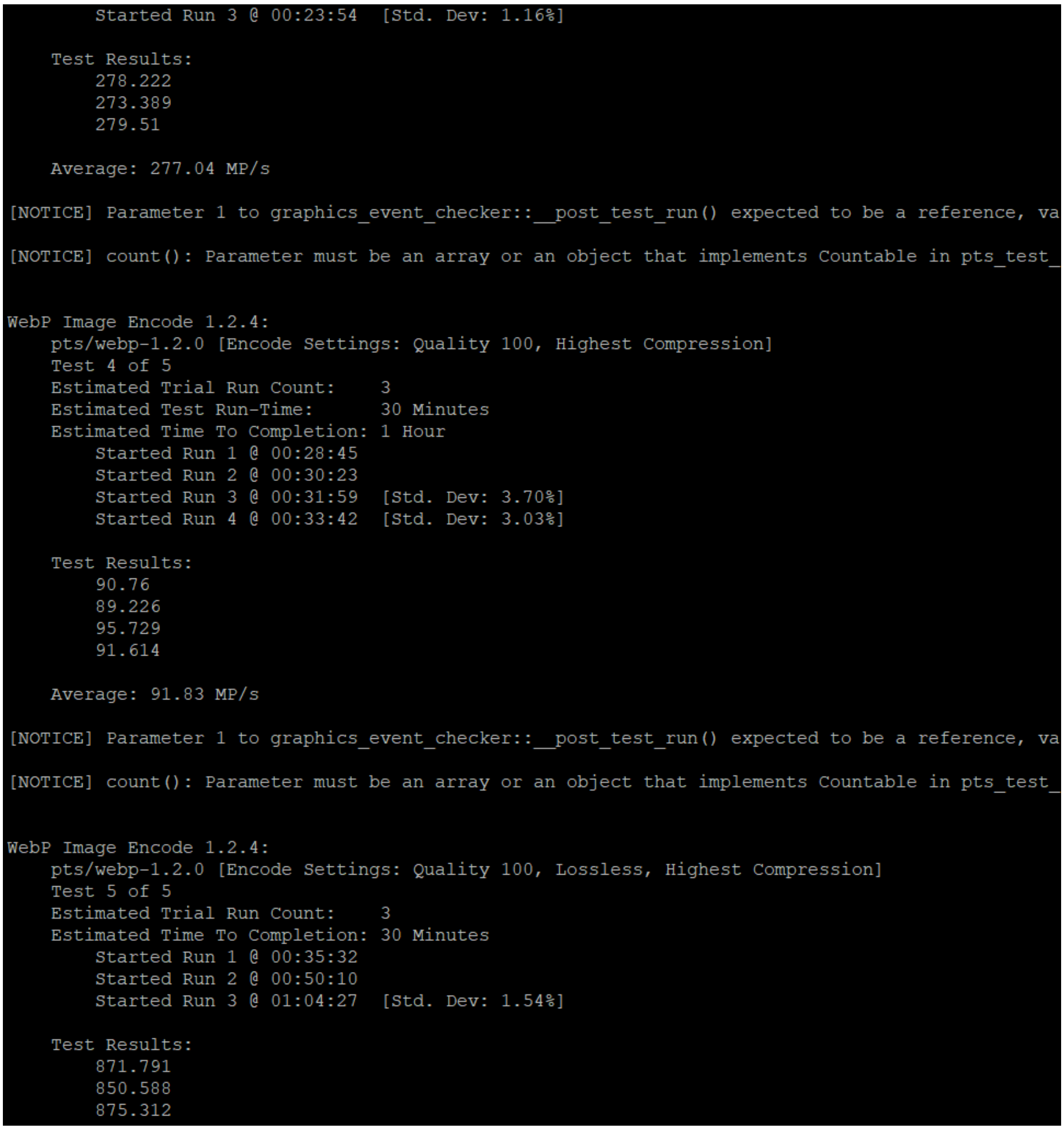


```
FFTE 7.0:
    pts/ffte-1.2.1
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 1 Minute
        Started Run 1 @ 11:24:29
        Started Run 2 @ 11:24:42
        Started Run 3 @ 11:24:54  [Std. Dev: 0.38%]

    Test Results:
        422.75821003656
        426.0175568635
        424.33911778487

    Average: 424.37 MFLOPS

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211106-BE-KAM61110186
```

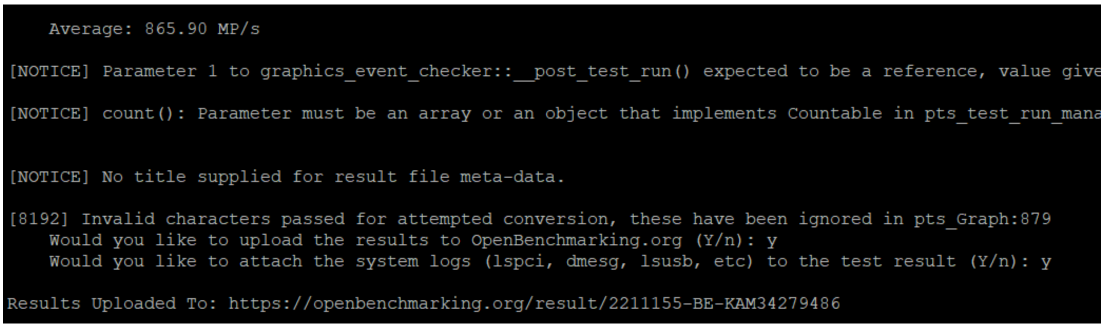
A10: WebP Image Encode



```
WebP Image Encode 1.2.4:
    pts/webp-1.2.0 [Encode Settings: Default]
    Test 1 of 5
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      30 Minutes
    Estimated Time To Completion: 2 Hours, 30 Minutes
        Started Run 1 @ 00:09:04
        Started Run 2 @ 00:09:50
        Started Run 3 @ 00:10:36  [Std. Dev: 0.29%]

    Test Results:
        38.735
        38.634
        38.855

    Average: 38.74 MP/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference, value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_test_run_manager:633


WebP Image Encode 1.2.4:
    pts/webp-1.2.0 [Encode Settings: Quality 100]
    Test 2 of 5
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      30 Minutes
    Estimated Time To Completion: 2 Hours
        Started Run 1 @ 00:11:26
        Started Run 2 @ 00:12:25
        Started Run 3 @ 00:13:25  [Std. Dev: 1.06%]

    Test Results:
        51.651
        52.741
        52.014

    Average: 52.14 MP/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference, value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_test_run_manager:633


WebP Image Encode 1.2.4:
    pts/webp-1.2.0 [Encode Settings: Quality 100, Lossless]
    Test 3 of 5
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      30 Minutes
    Estimated Time To Completion: 1 Hour, 30 Minutes
        Started Run 1 @ 00:14:30
        Started Run 2 @ 00:19:14
```

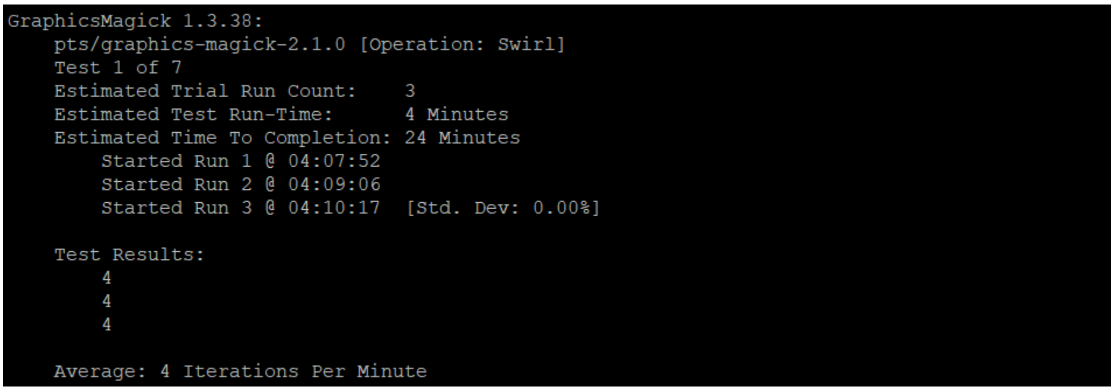


```
        Started Run 3 @ 00:23:54  [Std. Dev: 1.16%]

    Test Results:
        278.222
        273.389
        279.51

    Average: 277.04 MP/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference, va

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_test_


WebP Image Encode 1.2.4:
    pts/webp-1.2.0 [Encode Settings: Quality 100, Highest Compression]
    Test 4 of 5
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      30 Minutes
    Estimated Time To Completion: 1 Hour
        Started Run 1 @ 00:28:45
        Started Run 2 @ 00:30:23
        Started Run 3 @ 00:31:59  [Std. Dev: 3.70%]
        Started Run 4 @ 00:33:42  [Std. Dev: 3.03%]

    Test Results:
        90.76
        89.226
        95.729
        91.614

    Average: 91.83 MP/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference, va

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_test_


WebP Image Encode 1.2.4:
    pts/webp-1.2.0 [Encode Settings: Quality 100, Lossless, Highest Compression]
    Test 5 of 5
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 30 Minutes
        Started Run 1 @ 00:35:32
        Started Run 2 @ 00:50:10
        Started Run 3 @ 01:04:27  [Std. Dev: 1.54%]

    Test Results:
        871.791
        850.588
        875.312
```

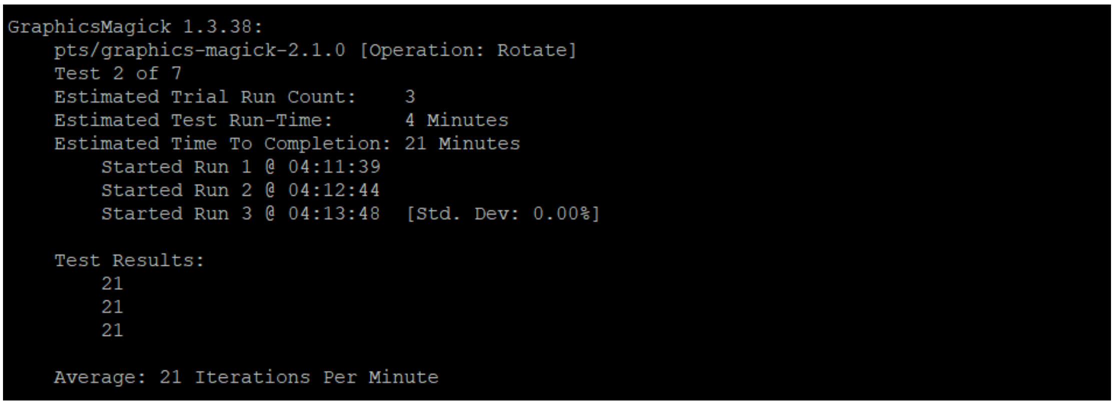


```
    Average: 865.90 MP/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference, value give

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_test_run_mana


[NOTICE] No title supplied for result file meta-data.

[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Graph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211155-BE-KAM34279486
```

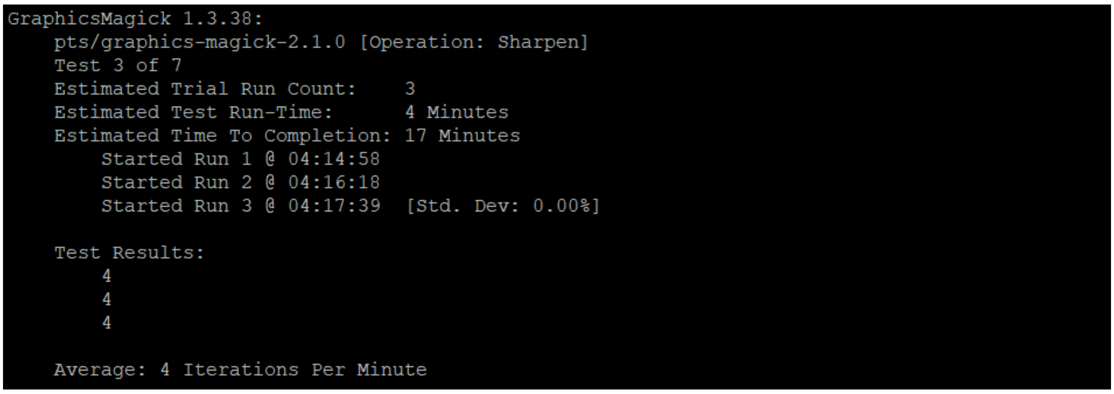
A11: GraphicsMagick



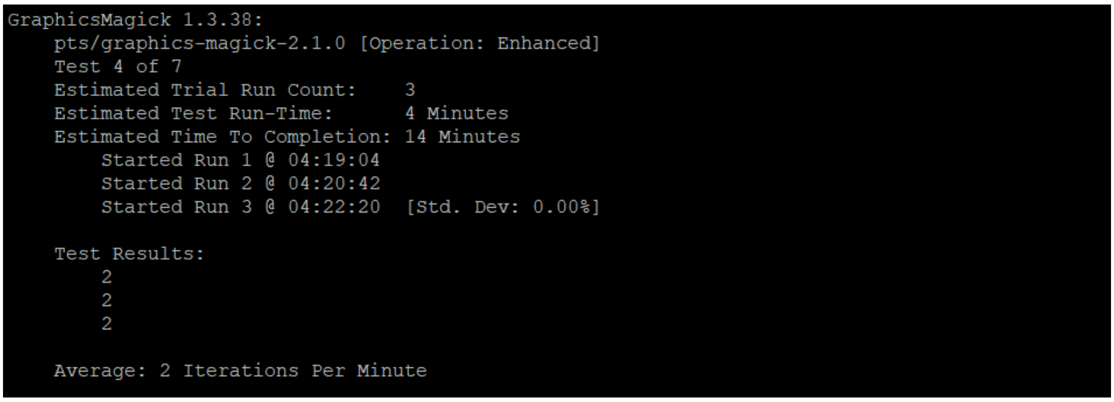
```
GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: Swirl]
    Test 1 of 7
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 24 Minutes
        Started Run 1 @ 04:07:52
        Started Run 2 @ 04:09:06
        Started Run 3 @ 04:10:17  [Std. Dev: 0.00%]

    Test Results:
        4
        4
        4

    Average: 4 Iterations Per Minute
```



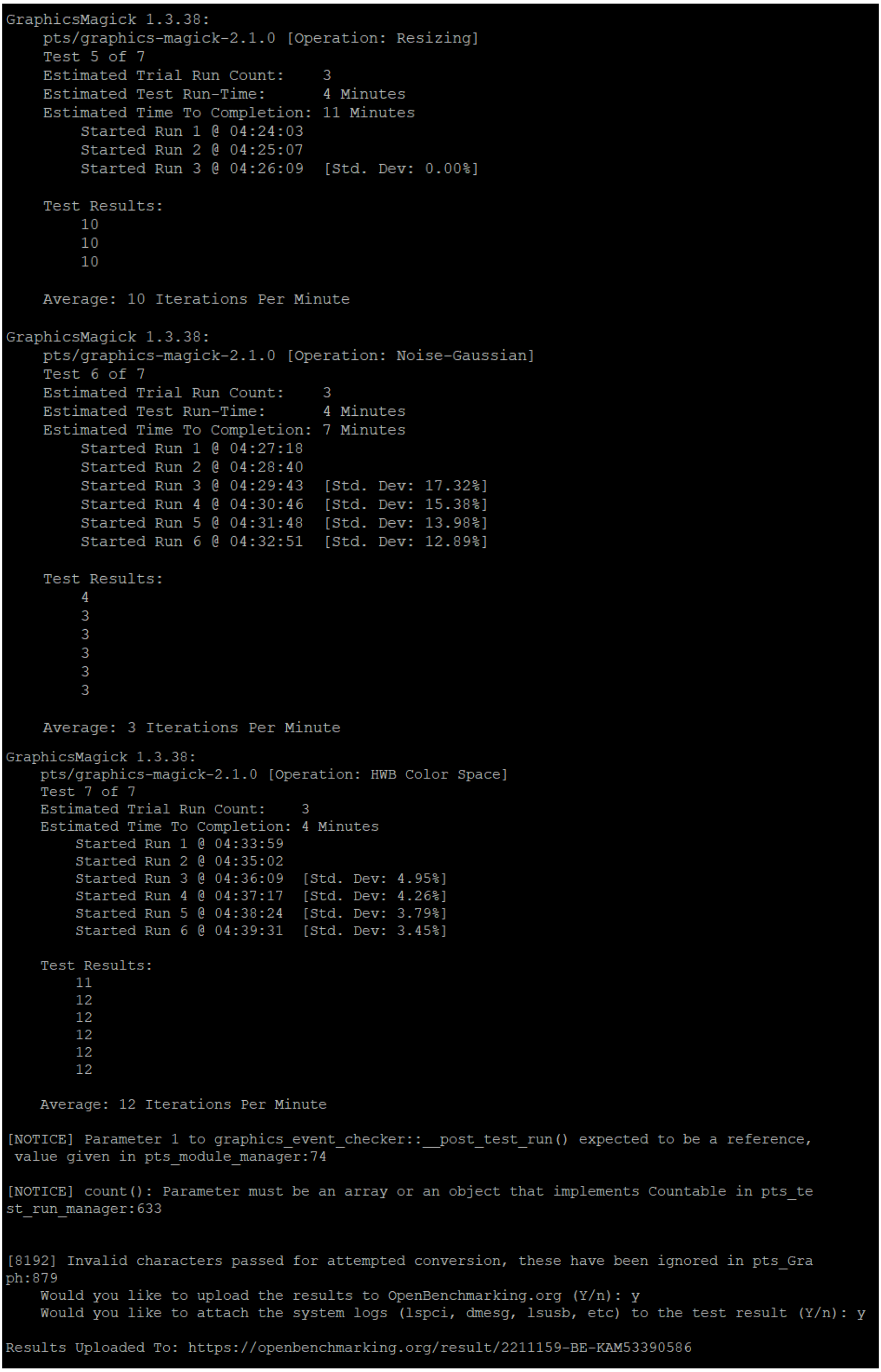
```
GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: Rotate]
    Test 2 of 7
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 21 Minutes
        Started Run 1 @ 04:11:39
        Started Run 2 @ 04:12:44
        Started Run 3 @ 04:13:48  [Std. Dev: 0.00%]

    Test Results:
        21
        21
        21

    Average: 21 Iterations Per Minute
```



```
GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: Sharpen]
    Test 3 of 7
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 17 Minutes
        Started Run 1 @ 04:14:58
        Started Run 2 @ 04:16:18
        Started Run 3 @ 04:17:39  [Std. Dev: 0.00%]

    Test Results:
        4
        4
        4

    Average: 4 Iterations Per Minute
```

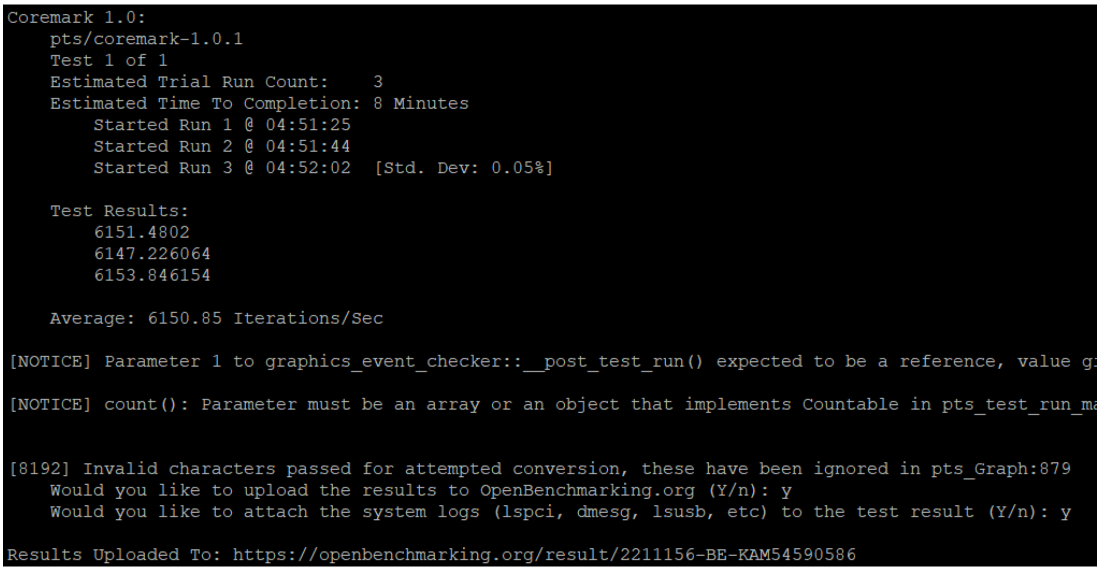


```
GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: Enhanced]
    Test 4 of 7
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 14 Minutes
        Started Run 1 @ 04:19:04
        Started Run 2 @ 04:20:42
        Started Run 3 @ 04:22:20  [Std. Dev: 0.00%]

    Test Results:
        2
        2
        2

    Average: 2 Iterations Per Minute
```



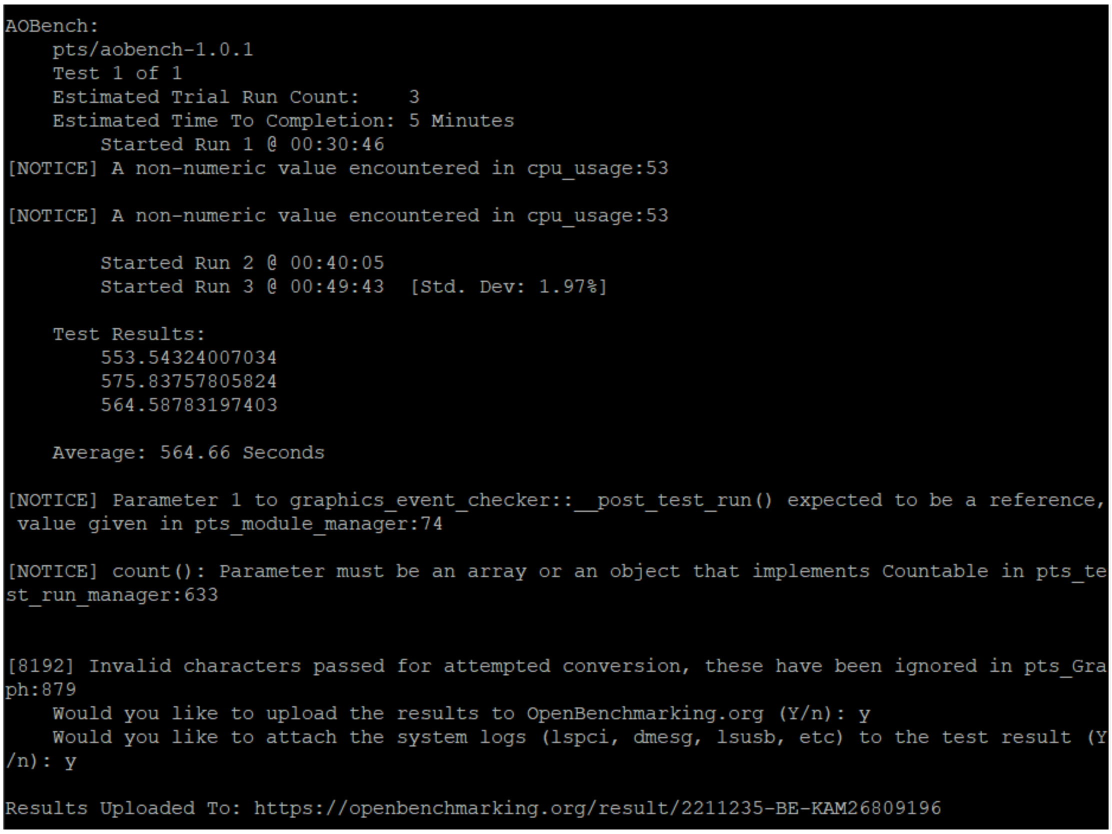
```
GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: Resizing]
    Test 5 of 7
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 11 Minutes
        Started Run 1 @ 04:24:03
        Started Run 2 @ 04:25:07
        Started Run 3 @ 04:26:09  [Std. Dev: 0.00%]

    Test Results:
        10
        10
        10

    Average: 10 Iterations Per Minute

GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: Noise-Gaussian]
    Test 6 of 7
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 7 Minutes
        Started Run 1 @ 04:27:18
        Started Run 2 @ 04:28:40
        Started Run 3 @ 04:29:43  [Std. Dev: 17.32%]
        Started Run 4 @ 04:30:46  [Std. Dev: 15.38%]
        Started Run 5 @ 04:31:48  [Std. Dev: 13.98%]
        Started Run 6 @ 04:32:51  [Std. Dev: 12.89%]

    Test Results:
        4
        3
        3
        3
        3
        3

    Average: 3 Iterations Per Minute

GraphicsMagick 1.3.38:
    pts/graphics-magick-2.1.0 [Operation: HWB Color Space]
    Test 7 of 7
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 4 Minutes
        Started Run 1 @ 04:33:59
        Started Run 2 @ 04:35:02
        Started Run 3 @ 04:36:09  [Std. Dev: 4.95%]
        Started Run 4 @ 04:37:17  [Std. Dev: 4.26%]
        Started Run 5 @ 04:38:24  [Std. Dev: 3.79%]
        Started Run 6 @ 04:39:31  [Std. Dev: 3.45%]

    Test Results:
        11
        12
        12
        12
        12
        12

    Average: 12 Iterations Per Minute

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211159-BE-KAM53390586
```

A12: Coremark



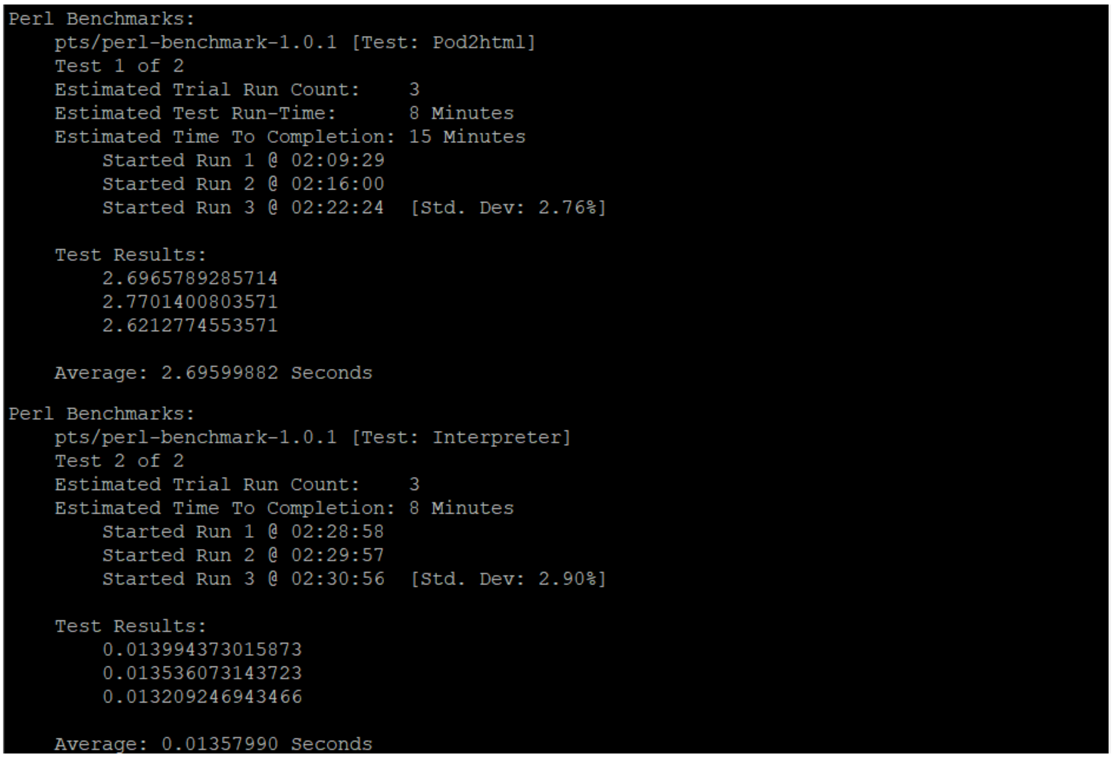
```
Coremark 1.0:
    pts/coremark-1.0.1
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 8 Minutes
        Started Run 1 @ 04:51:25
        Started Run 2 @ 04:51:44
        Started Run 3 @ 04:52:02  [Std. Dev: 0.05%]

    Test Results:
        6151.4802
        6147.226064
        6153.846154

    Average: 6150.85 Iterations/Sec

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference, value g

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_test_run_ma


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Graph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211156-BE-KAM54590586
```

A13: AOBench



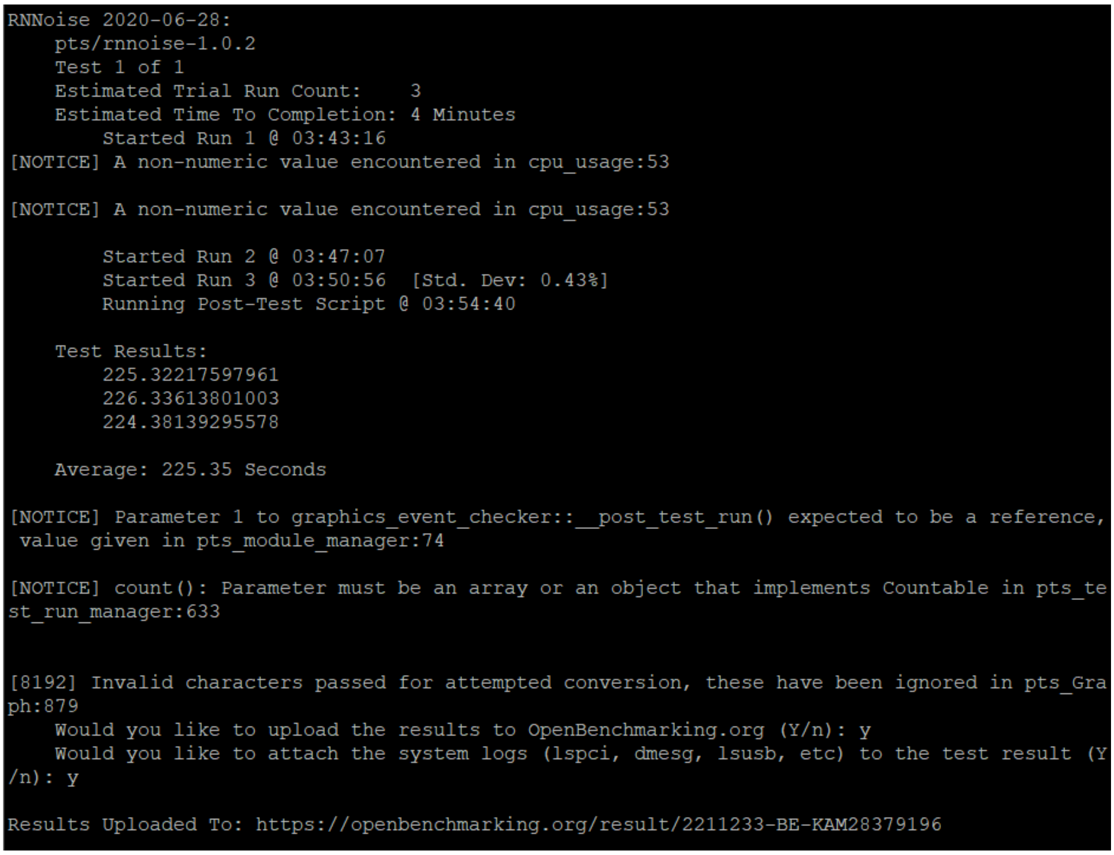
```
AOBench:
    pts/aobench-1.0.1
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 5 Minutes
        Started Run 1 @ 00:30:46
[NOTICE] A non-numeric value encountered in cpu_usage:53

[NOTICE] A non-numeric value encountered in cpu_usage:53

        Started Run 2 @ 00:40:05
        Started Run 3 @ 00:49:43  [Std. Dev: 1.97%]

    Test Results:
        553.54324007034
        575.83757805824
        564.58783197403

    Average: 564.66 Seconds

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211235-BE-KAM26809196
```

A14: Perl Benchmarks



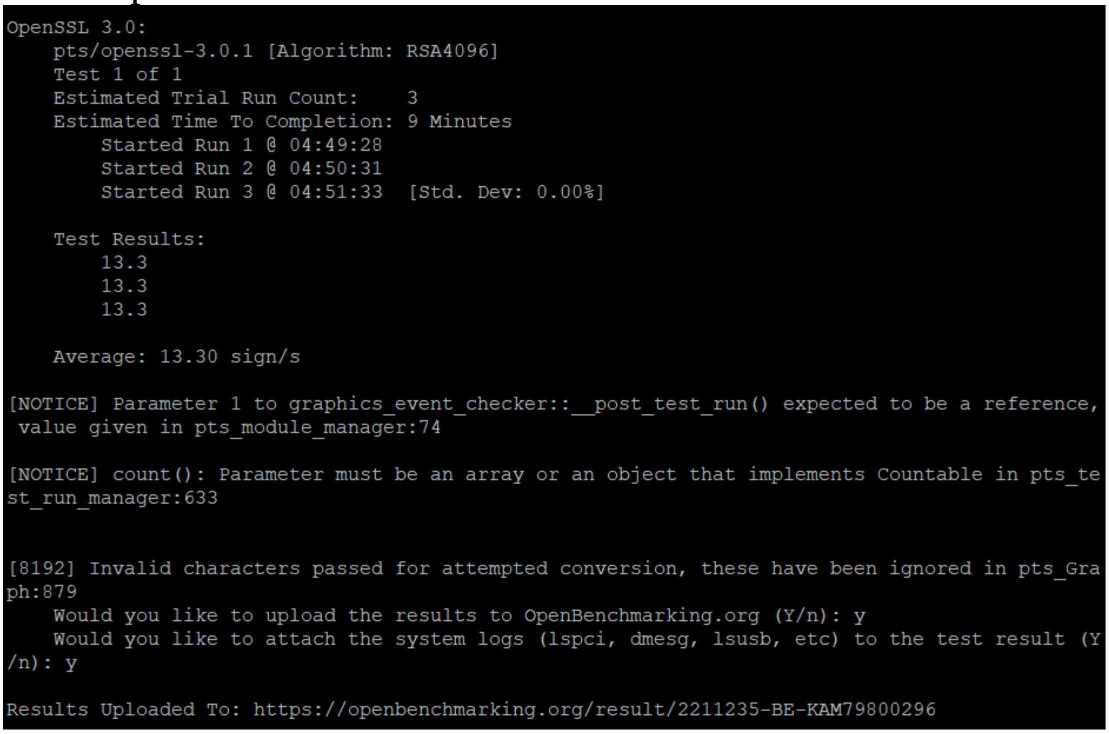
```
Perl Benchmarks:
    pts/perl-benchmark-1.0.1 [Test: Pod2html]
    Test 1 of 2
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      8 Minutes
    Estimated Time To Completion: 15 Minutes
        Started Run 1 @ 02:09:29
        Started Run 2 @ 02:16:00
        Started Run 3 @ 02:22:24  [Std. Dev: 2.76%]

    Test Results:
        2.6965789285714
        2.7701400803571
        2.6212774553571

    Average: 2.69599882 Seconds

Perl Benchmarks:
    pts/perl-benchmark-1.0.1 [Test: Interpreter]
    Test 2 of 2
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 8 Minutes
        Started Run 1 @ 02:28:58
        Started Run 2 @ 02:29:57
        Started Run 3 @ 02:30:56  [Std. Dev: 2.90%]

    Test Results:
        0.013994373015873
        0.013536073143723
        0.013209246943466

    Average: 0.01357990 Seconds
```

A15: RNNoice



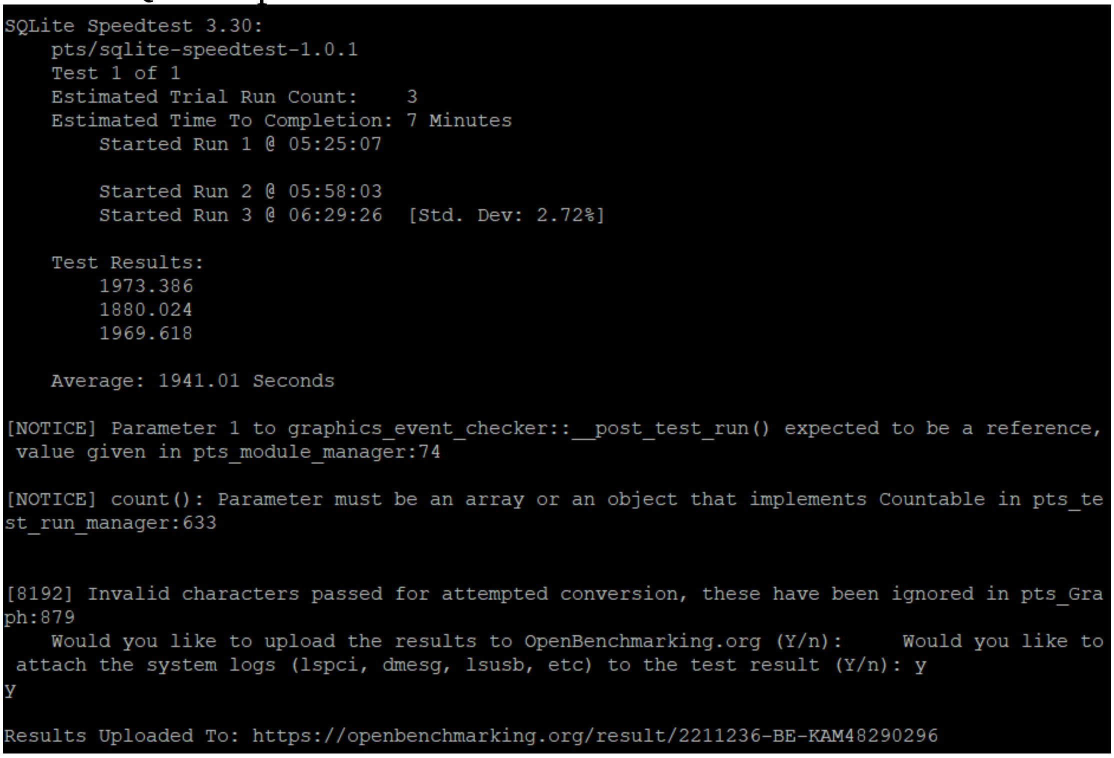
```
RNNoise 2020-06-28:
    pts/rnnoise-1.0.2
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 4 Minutes
        Started Run 1 @ 03:43:16
[NOTICE] A non-numeric value encountered in cpu_usage:53

[NOTICE] A non-numeric value encountered in cpu_usage:53

        Started Run 2 @ 03:47:07
        Started Run 3 @ 03:50:56  [Std. Dev: 0.43%]
        Running Post-Test Script @ 03:54:40

    Test Results:
        225.32217597961
        226.33613801003
        224.38139295578

    Average: 225.35 Seconds

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211233-BE-KAM28379196
```

A16: OpenSSL



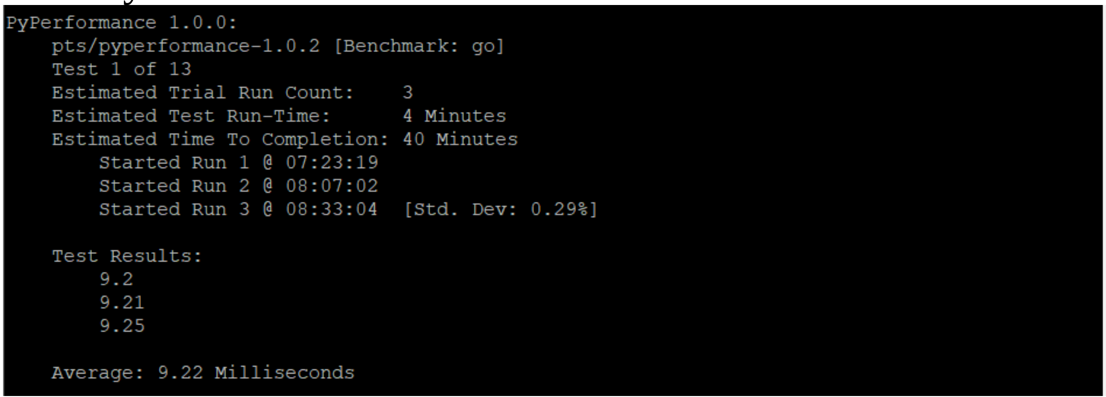
```
OpenSSL 3.0:
    pts/openssl-3.0.1 [Algorithm: RSA4096]
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 9 Minutes
        Started Run 1 @ 04:49:28
        Started Run 2 @ 04:50:31
        Started Run 3 @ 04:51:33  [Std. Dev: 0.00%]

    Test Results:
        13.3
        13.3
        13.3

    Average: 13.30 sign/s

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211235-BE-KAM79800296
```

A17: SQLite Speedtest

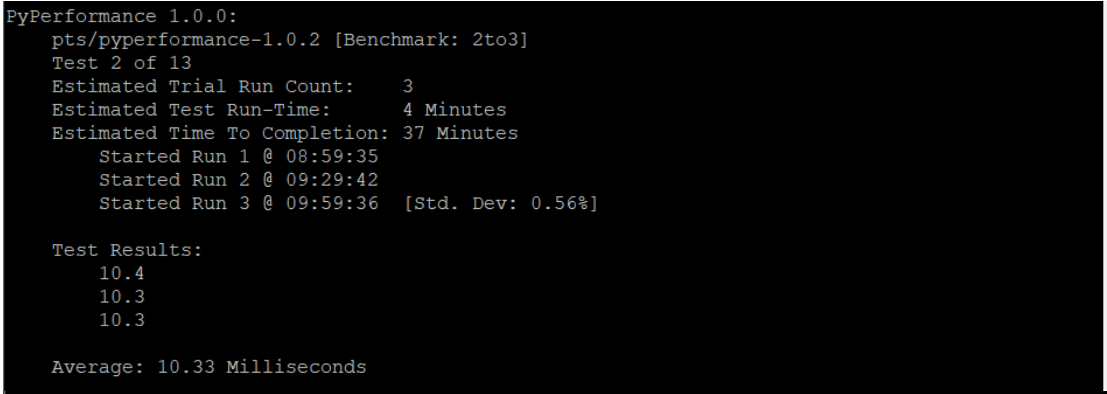


```
SQLite Speedtest 3.30:
    pts/sqlite-speedtest-1.0.1
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 7 Minutes
        Started Run 1 @ 05:25:07

        Started Run 2 @ 05:58:03
        Started Run 3 @ 06:29:26  [Std. Dev: 2.72%]

    Test Results:
        1973.386
        1880.024
        1969.618

    Average: 1941.01 Seconds

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n):     Would you like to
 attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/n): y
y

Results Uploaded To: https://openbenchmarking.org/result/2211236-BE-KAM48290296
```

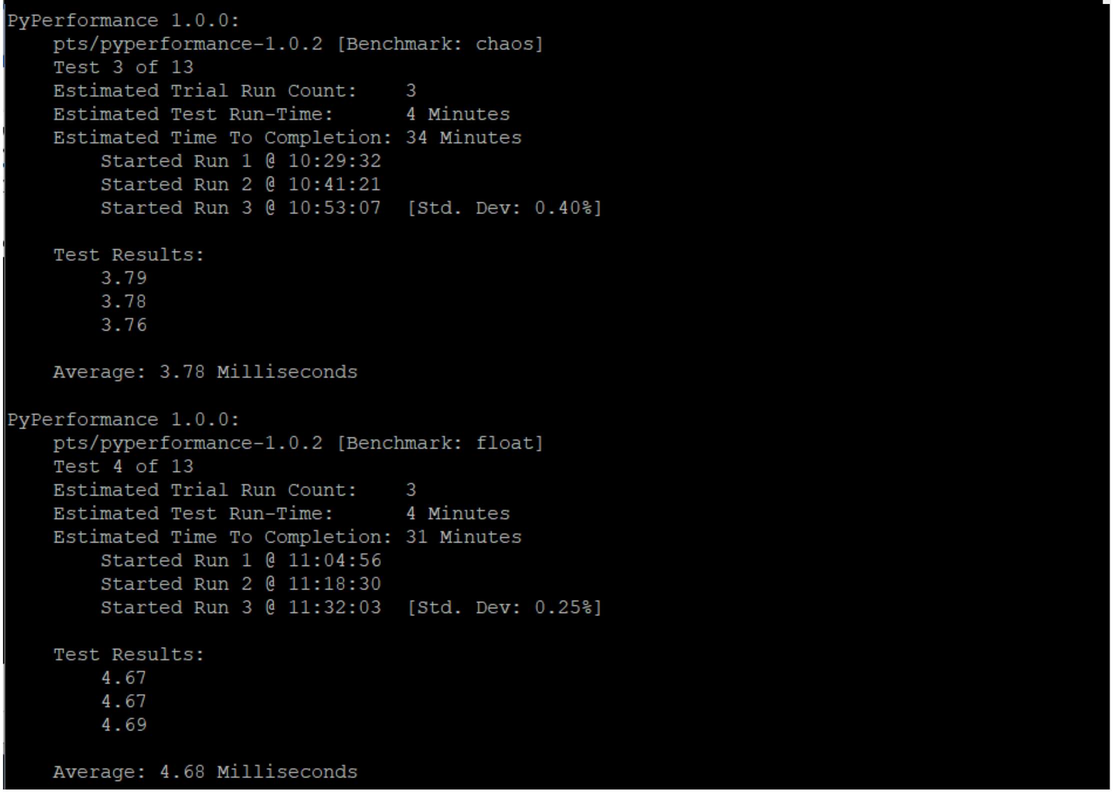
A18: PyPerformance



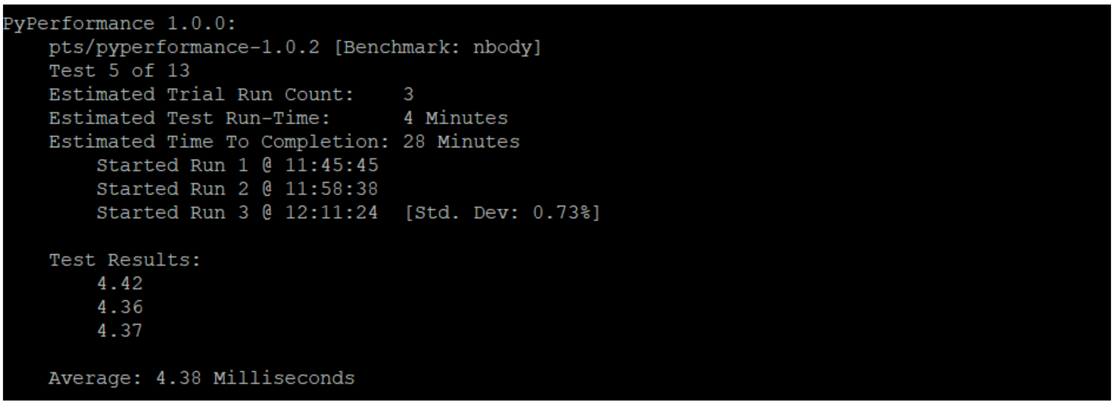
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: go]
    Test 1 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 40 Minutes
        Started Run 1 @ 07:23:19
        Started Run 2 @ 08:07:02
        Started Run 3 @ 08:33:04  [Std. Dev: 0.29%]

    Test Results:
        9.2
        9.21
        9.25

    Average: 9.22 Milliseconds
```



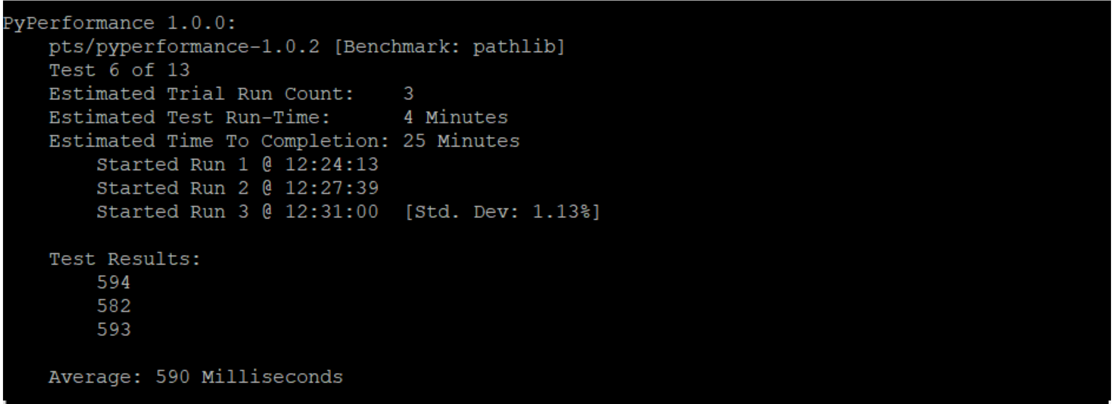
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: 2to3]
    Test 2 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 37 Minutes
        Started Run 1 @ 08:59:35
        Started Run 2 @ 09:29:42
        Started Run 3 @ 09:59:36  [Std. Dev: 0.56%]

    Test Results:
        10.4
        10.3
        10.3

    Average: 10.33 Milliseconds
```



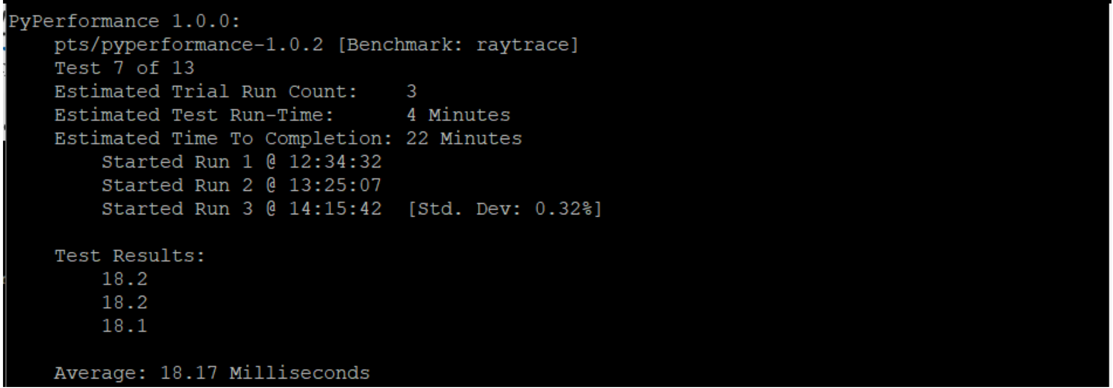
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: chaos]
    Test 3 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 34 Minutes
        Started Run 1 @ 10:29:32
        Started Run 2 @ 10:41:21
        Started Run 3 @ 10:53:07  [Std. Dev: 0.40%]

    Test Results:
        3.79
        3.78
        3.76

    Average: 3.78 Milliseconds

PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: float]
    Test 4 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 31 Minutes
        Started Run 1 @ 11:04:56
        Started Run 2 @ 11:18:30
        Started Run 3 @ 11:32:03  [Std. Dev: 0.25%]

    Test Results:
        4.67
        4.67
        4.69

    Average: 4.68 Milliseconds
```



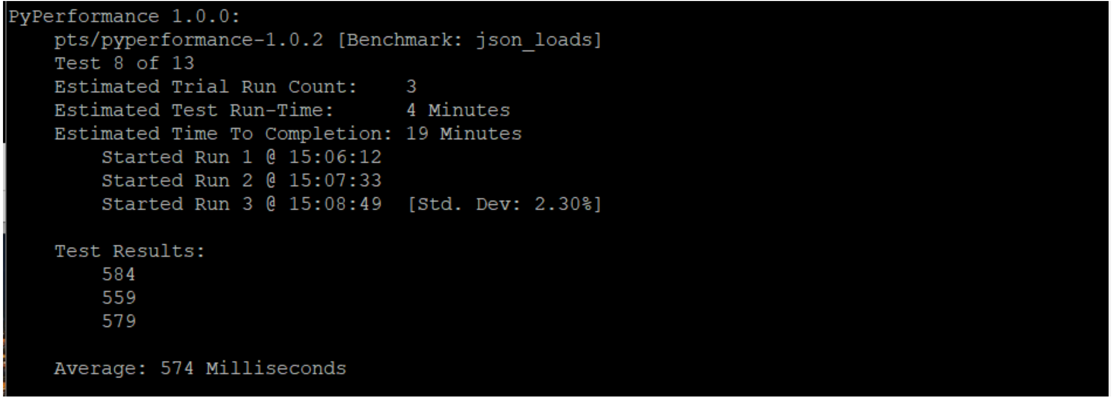
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: nbody]
    Test 5 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 28 Minutes
        Started Run 1 @ 11:45:45
        Started Run 2 @ 11:58:38
        Started Run 3 @ 12:11:24  [Std. Dev: 0.73%]

    Test Results:
        4.42
        4.36
        4.37

    Average: 4.38 Milliseconds
```



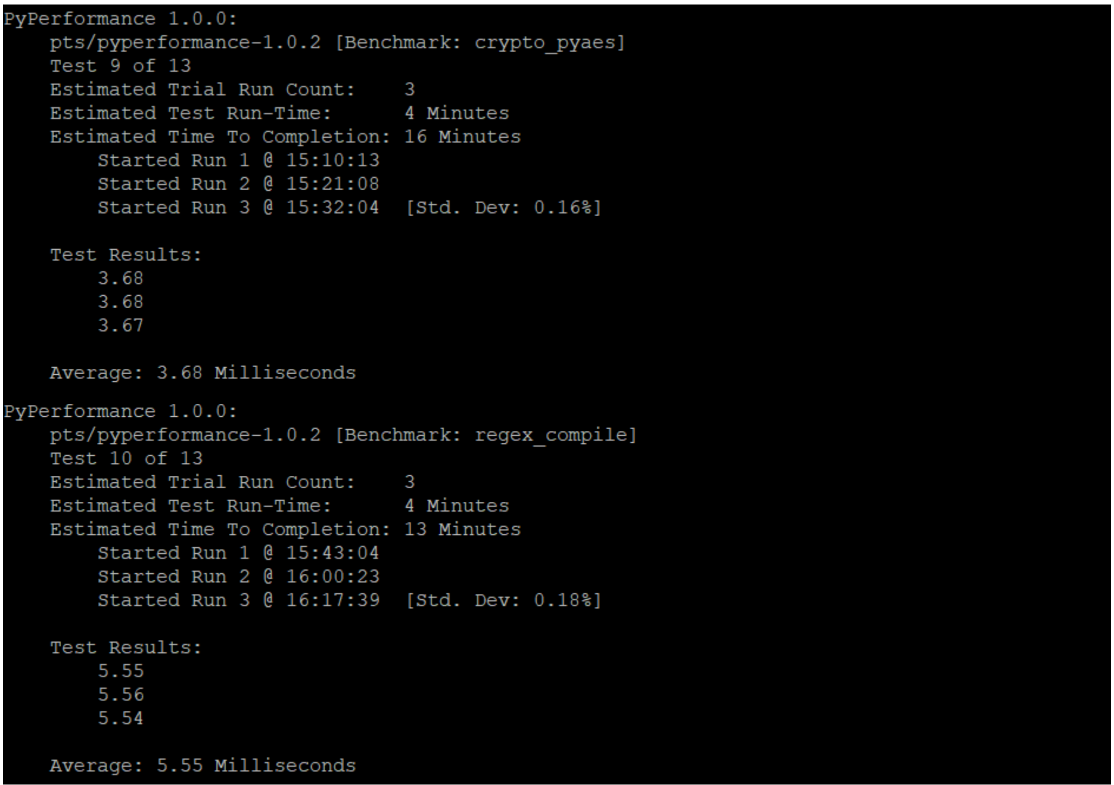
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: pathlib]
    Test 6 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 25 Minutes
        Started Run 1 @ 12:24:13
        Started Run 2 @ 12:27:39
        Started Run 3 @ 12:31:00  [Std. Dev: 1.13%]

    Test Results:
        594
        582
        593

    Average: 590 Milliseconds
```



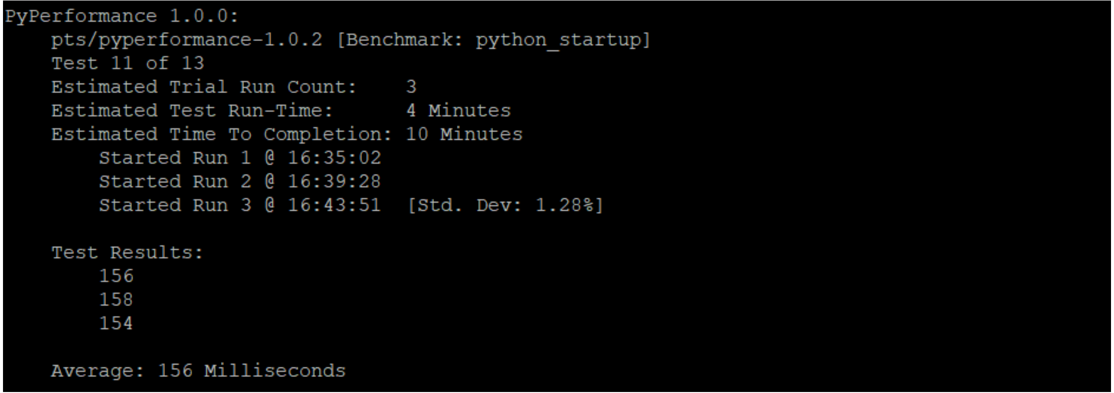
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: raytrace]
    Test 7 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 22 Minutes
        Started Run 1 @ 12:34:32
        Started Run 2 @ 13:25:07
        Started Run 3 @ 14:15:42  [Std. Dev: 0.32%]

    Test Results:
        18.2
        18.2
        18.1

    Average: 18.17 Milliseconds
```



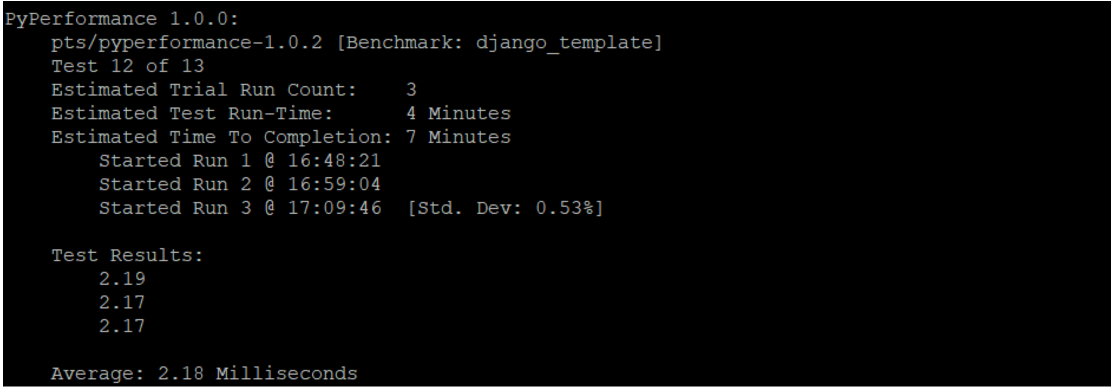
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: json_loads]
    Test 8 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 19 Minutes
        Started Run 1 @ 15:06:12
        Started Run 2 @ 15:07:33
        Started Run 3 @ 15:08:49  [Std. Dev: 2.30%]

    Test Results:
        584
        559
        579

    Average: 574 Milliseconds
```



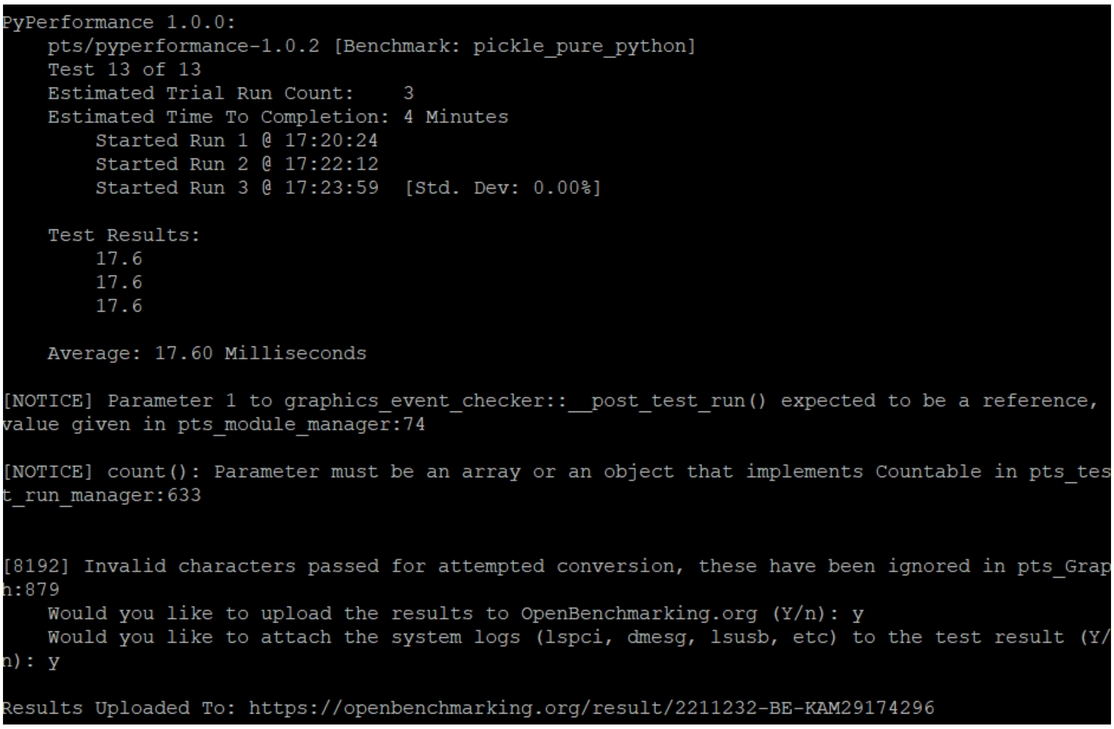
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: crypto_pyaes]
    Test 9 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 16 Minutes
        Started Run 1 @ 15:10:13
        Started Run 2 @ 15:21:08
        Started Run 3 @ 15:32:04  [Std. Dev: 0.16%]

    Test Results:
        3.68
        3.68
        3.67

    Average: 3.68 Milliseconds

PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: regex_compile]
    Test 10 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 13 Minutes
        Started Run 1 @ 15:43:04
        Started Run 2 @ 16:00:23
        Started Run 3 @ 16:17:39  [Std. Dev: 0.18%]

    Test Results:
        5.55
        5.56
        5.54

    Average: 5.55 Milliseconds
```



```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: python_startup]
    Test 11 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 10 Minutes
        Started Run 1 @ 16:35:02
        Started Run 2 @ 16:39:28
        Started Run 3 @ 16:43:51  [Std. Dev: 1.28%]

    Test Results:
        156
        158
        154

    Average: 156 Milliseconds
```

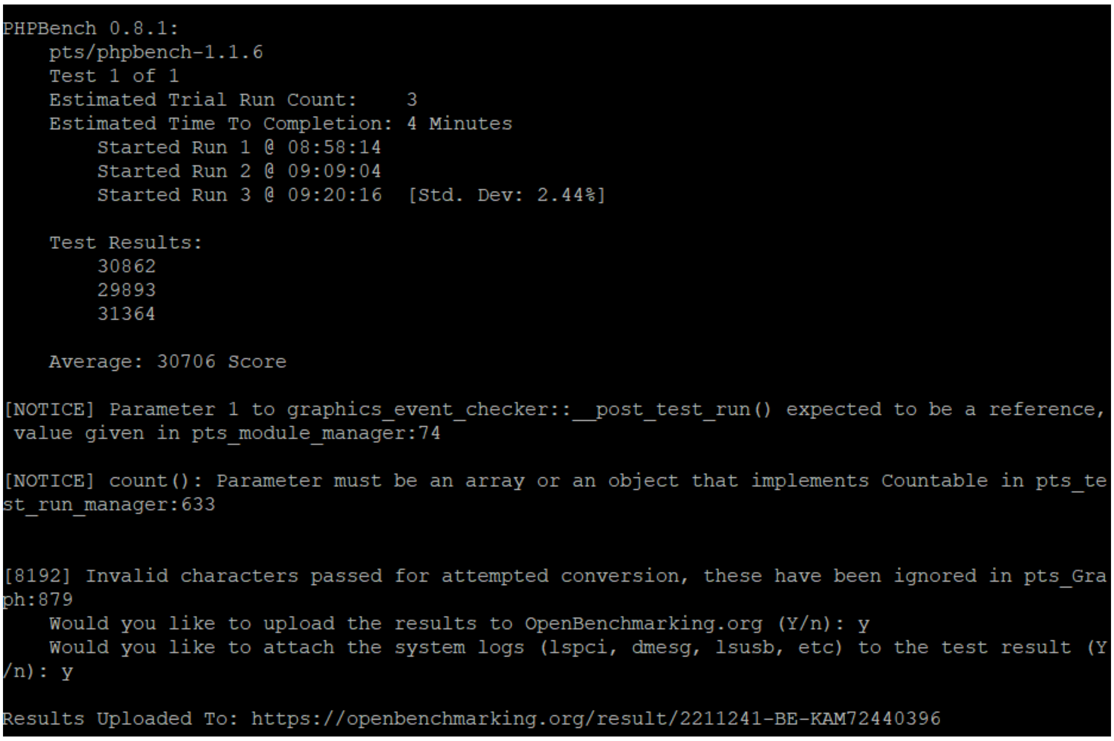


```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: django_template]
    Test 12 of 13
    Estimated Trial Run Count:    3
    Estimated Test Run-Time:      4 Minutes
    Estimated Time To Completion: 7 Minutes
        Started Run 1 @ 16:48:21
        Started Run 2 @ 16:59:04
        Started Run 3 @ 17:09:46  [Std. Dev: 0.53%]

    Test Results:
        2.19
        2.17
        2.17

    Average: 2.18 Milliseconds
```



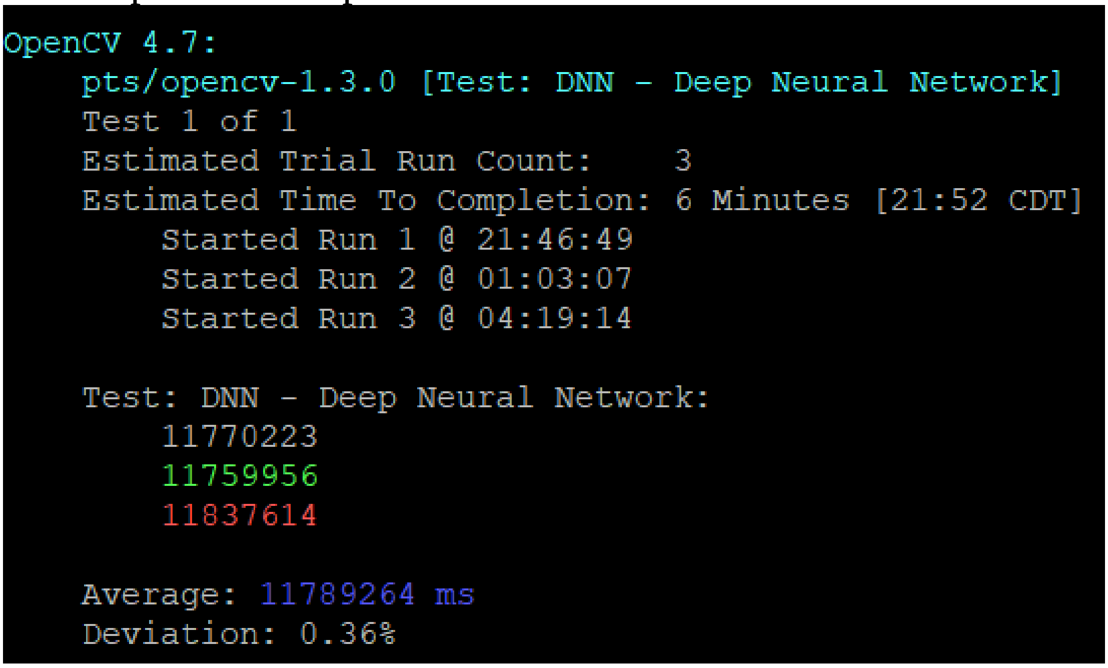
```
PyPerformance 1.0.0:
    pts/pyperformance-1.0.2 [Benchmark: pickle_pure_python]
    Test 13 of 13
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 4 Minutes
        Started Run 1 @ 17:20:24
        Started Run 2 @ 17:22:12
        Started Run 3 @ 17:23:59  [Std. Dev: 0.00%]

    Test Results:
        17.6
        17.6
        17.6

    Average: 17.60 Milliseconds

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_tes
t_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Grap
h:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y/
n): y

Results Uploaded To: https://openbenchmarking.org/result/2211232-BE-KAM29174296
```

A19: PHPBench



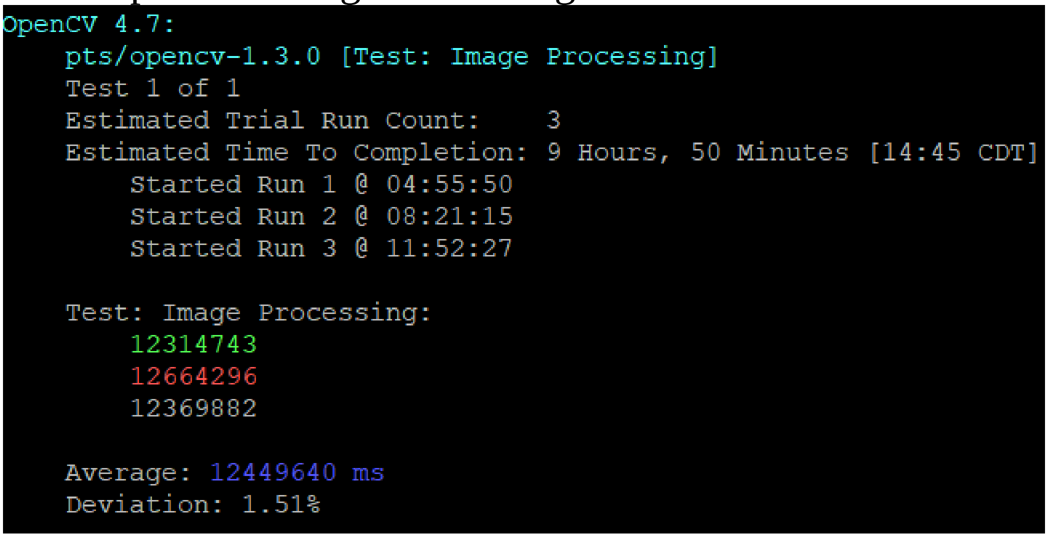
```
PHPBench 0.8.1:
    pts/phpbench-1.1.6
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 4 Minutes
        Started Run 1 @ 08:58:14
        Started Run 2 @ 09:09:04
        Started Run 3 @ 09:20:16  [Std. Dev: 2.44%]

    Test Results:
        30862
        29893
        31364

    Average: 30706 Score

[NOTICE] Parameter 1 to graphics_event_checker::__post_test_run() expected to be a reference,
 value given in pts_module_manager:74

[NOTICE] count(): Parameter must be an array or an object that implements Countable in pts_te
st_run_manager:633


[8192] Invalid characters passed for attempted conversion, these have been ignored in pts_Gra
ph:879
    Would you like to upload the results to OpenBenchmarking.org (Y/n): y
    Would you like to attach the system logs (lspci, dmesg, lsusb, etc) to the test result (Y
/n): y

Results Uploaded To: https://openbenchmarking.org/result/2211241-BE-KAM72440396
```

A20: OpenCV: Deep Neural Network



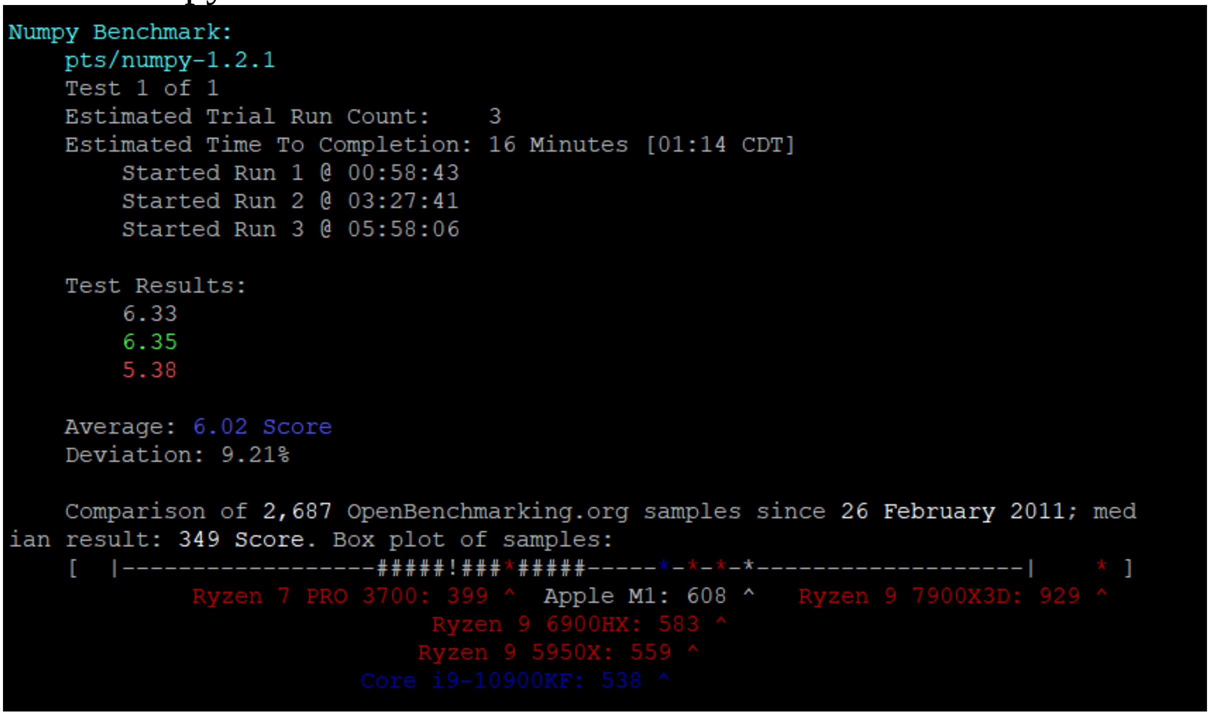
```
OpenCV 4.7:
    pts/opencv-1.3.0 [Test: DNN - Deep Neural Network]
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 6 Minutes [21:52 CDT]
        Started Run 1 @ 21:46:49
        Started Run 2 @ 01:03:07
        Started Run 3 @ 04:19:14

    Test: DNN - Deep Neural Network:
        11770223
        11759956
        11837614

    Average: 11789264 ms
    Deviation: 0.36%
```

A21: OpenCV: Image Processing



```
OpenCV 4.7:
    pts/opencv-1.3.0 [Test: Image Processing]
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 9 Hours, 50 Minutes [14:45 CDT]
        Started Run 1 @ 04:55:50
        Started Run 2 @ 08:21:15
        Started Run 3 @ 11:52:27

    Test: Image Processing:
        12314743
        12664296
        12369882

    Average: 12449640 ms
    Deviation: 1.51%
```

A22: Numpy Benchmark

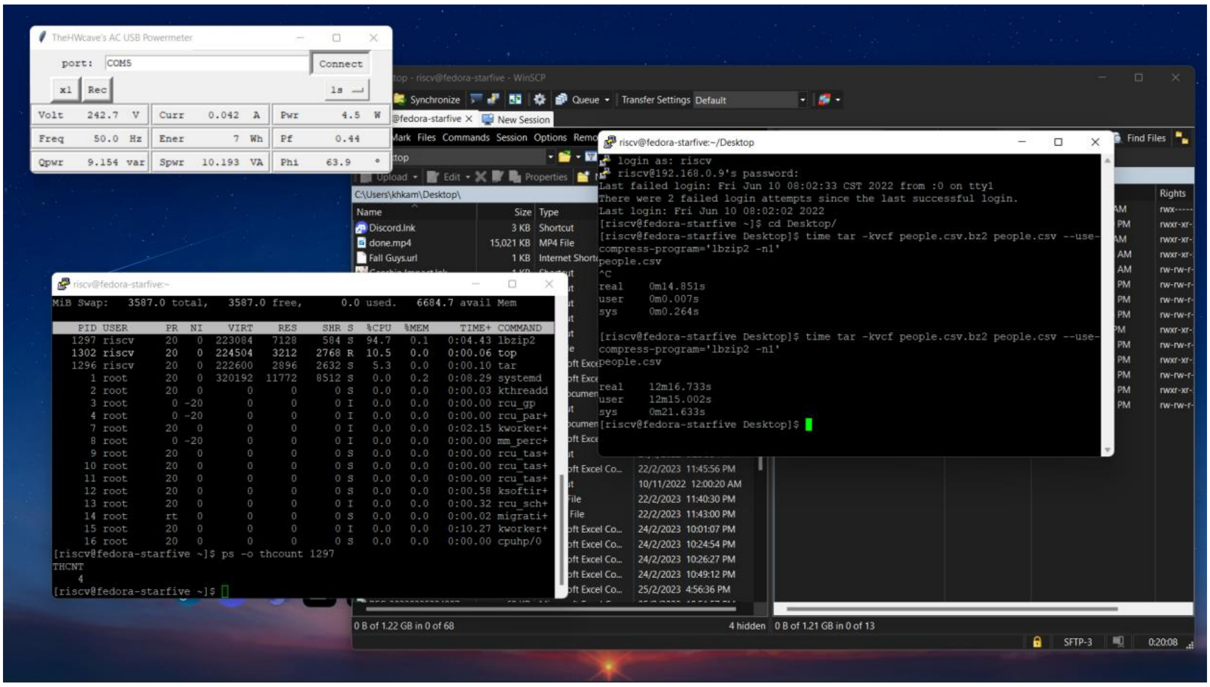


```
Numpy Benchmark:
    pts/numpy-1.2.1
    Test 1 of 1
    Estimated Trial Run Count:    3
    Estimated Time To Completion: 16 Minutes [01:14 CDT]
        Started Run 1 @ 00:58:43
        Started Run 2 @ 03:27:41
        Started Run 3 @ 05:58:06

    Test Results:
        6.33
        6.35
        5.38

    Average: 6.02 Score
    Deviation: 9.21%

    Comparison of 2,687 OpenBenchmarking.org samples since 26 February 2011; med
ian result: 349 Score. Box plot of samples:
    [  |------------------#####!###*#####-----*-*-*-*-------------------|    * ]
             Ryzen 7 PRO 3700: 399 ^  Apple M1: 608 ^   Ryzen 9 7900X3D: 929 ^
                              Ryzen 9 6900HX: 583 ^
                             Ryzen 9 5950X: 559 ^
                         Core i9-10900KF: 538 ^
```

APPENDIX B – RESULT OF DATASET COMPRESSION AND POWER CONSUMPTION

Lbzip2 Dataset Compression performed by VisionFive V1 Board:

1 Thread

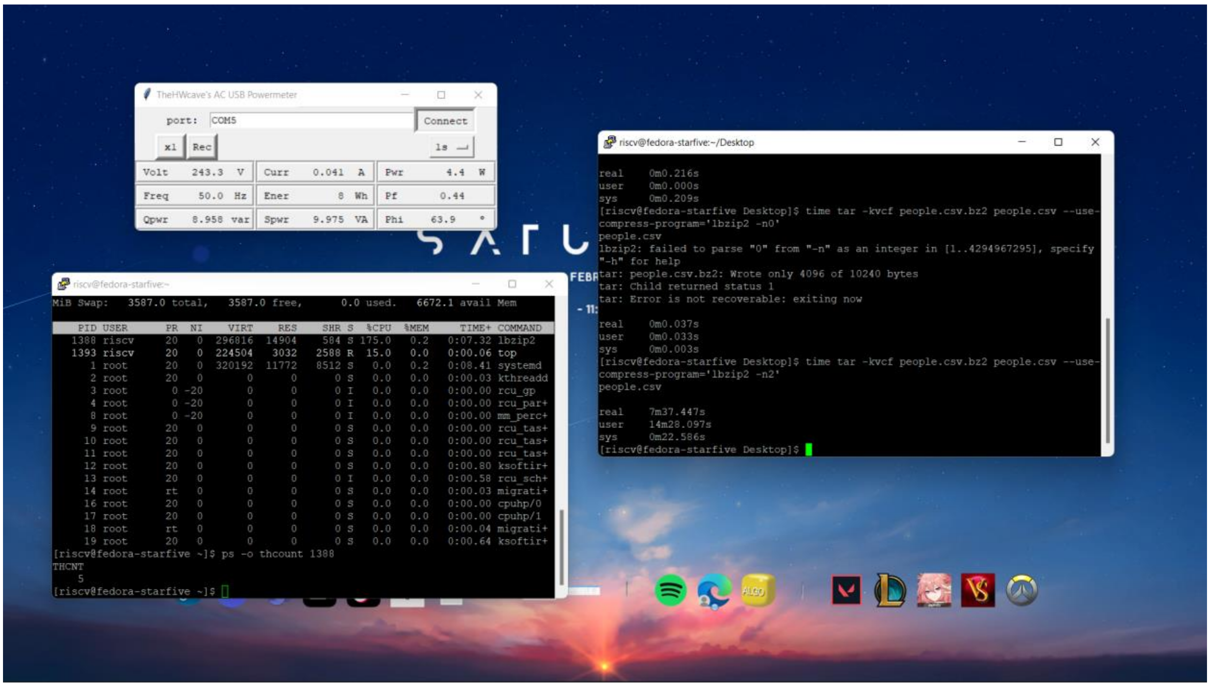


```
TheHWcave's AC USB Powermeter
port: COM5                    Connect
xl  Rec                       1s
Volt   242.7  V    Curr    0.042  A    Pwr      4.5  W
Freq    50.0  Hz   Ener        7  Wh   Pf      0.44
Qpwr   9.154  var  Spwr   10.193  VA   Phi     63.9  °
```

```
riscv@fedora-starfive:~/Desktop
login as: riscv
riscv@192.168.0.9's password:
Last failed login: Fri Jun 10 08:02:33 CST 2022 from :0 on tty1
There were 2 failed login attempts since the last successful login.
Last login: Fri Jun 10 08:02:02 2022
[riscv@fedora-starfive ~]$ cd Desktop/
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n1'
people.csv
^C
real    0m14.851s
user    0m0.007s
sys     0m0.264s

[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n1'
people.csv

real    12m16.733s
user    12m15.002s
sys     0m21.633s
[riscv@fedora-starfive Desktop]$
```

```
riscv@fedora-starfive:~
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6684.7 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1297 riscv     20   0  223084   7128    584 S  94.7   0.1   0:04.43 lbzip2
 1302 riscv     20   0  224504   3212   2768 R  10.5   0.0   0:00.06 top
 1296 riscv     20   0  222600   2896   2632 S   5.3   0.0   0:00.10 tar
    1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.29 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
    7 root      20   0       0      0      0 I   0.0   0.0   0:02.15 kworker+
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
    9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.58 ksoftir+
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.32 rcu_sch+
   14 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migrati+
   15 root      20   0       0      0      0 I   0.0   0.0   0:10.27 kworker+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
[riscv@fedora-starfive ~]$ ps -o thcount 1297
THCNT
    4
[riscv@fedora-starfive ~]$
```

2 Threads

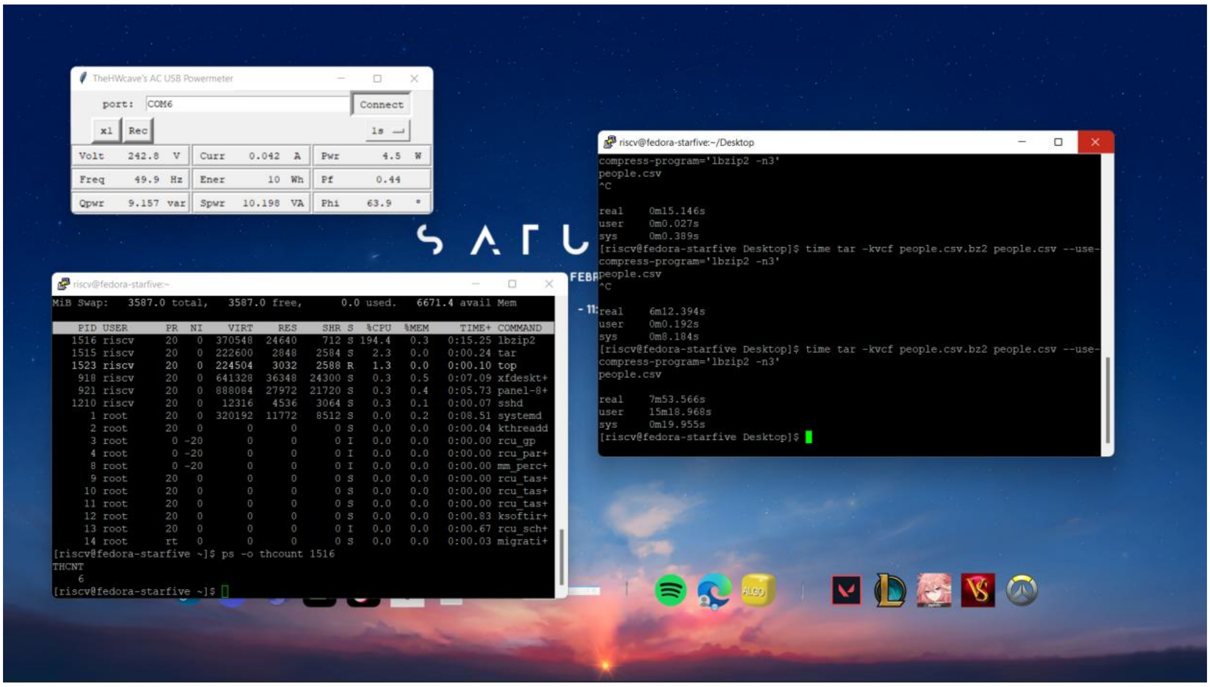


```
TheHWcave's AC USB Powermeter
port: COM5                    Connect
xl  Rec                       1s
Volt   243.3  V    Curr    0.041  A    Pwr      4.4  W
Freq    50.0  Hz   Ener        8  Wh   Pf      0.44
Qpwr   8.958  var  Spwr    9.975  VA   Phi     63.9  °
```

```
riscv@fedora-starfive:~/Desktop
real    0m0.216s
user    0m0.000s
sys     0m0.209s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n0'
people.csv
lbzip2: failed to parse "0" from "-n" as an integer in [1..4294967295], specify
"-h" for help
tar: people.csv.bz2: Wrote only 4096 of 10240 bytes
tar: Child returned status 1
tar: Error is not recoverable: exiting now

real    0m0.037s
user    0m0.033s
sys     0m0.003s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n2'
people.csv

real    7m37.447s
user    14m28.097s
sys     0m22.586s
[riscv@fedora-starfive Desktop]$
```

```
riscv@fedora-starfive:~
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6672.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1388 riscv     20   0  296816  14904    584 S 175.0   0.2   0:07.32 lbzip2
 1393 riscv     20   0  224504   3032   2588 R  15.0   0.0   0:00.06 top
    1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.41 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
    9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.80 ksoftir+
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.58 rcu_sch+
   14 root      rt   0       0      0      0 S   0.0   0.0   0:00.03 migrati+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   18 root      rt   0       0      0      0 S   0.0   0.0   0:00.04 migrati+
   19 root      20   0       0      0      0 S   0.0   0.0   0:00.64 ksoftir+
[riscv@fedora-starfive ~]$ ps -o thcount 1388
THCNT
    5
[riscv@fedora-starfive ~]$
```

3 Threads

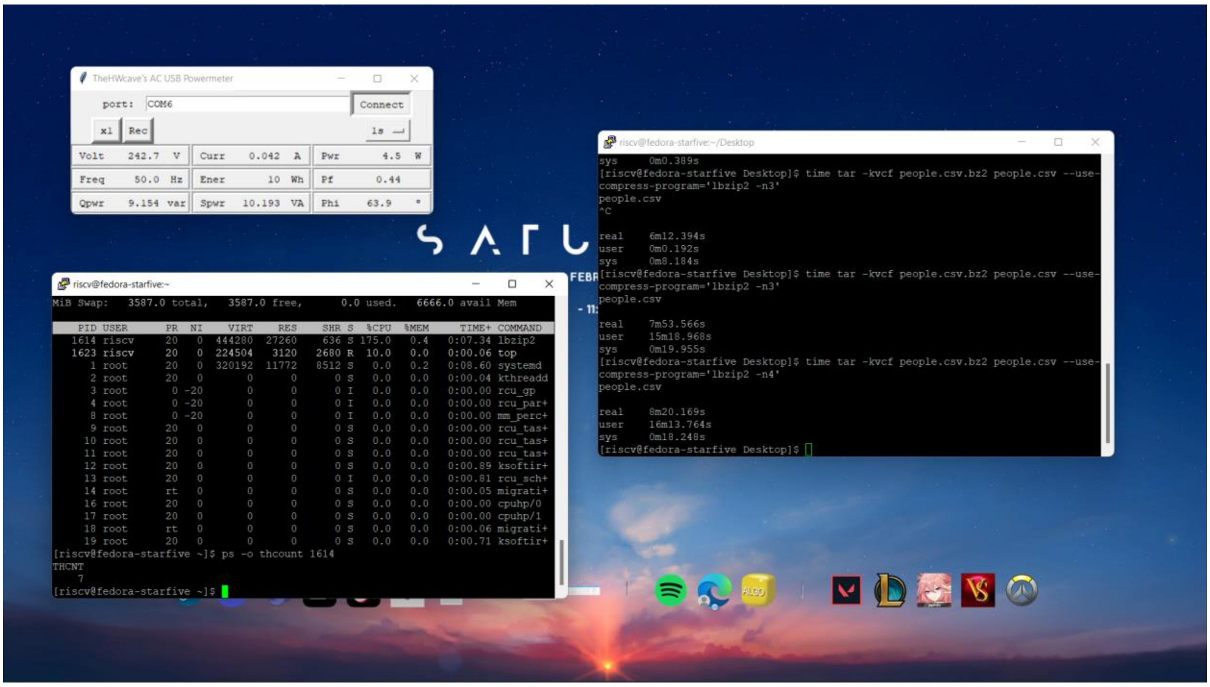

TheHWcave's AC USB Powermeter
Connect
x1
Rec
1s
riscv@fedora-starfive:~
riscv@fedora-starfive:~/Desktop

```
port: COM6
Volt   242.8  V    Curr   0.042  A    Pwr      4.5  W
Freq    49.9  Hz   Ener      10  Wh   Pf      0.44
Qpwr   9.157  var  Spwr  10.198  VA   Phi     63.9  °
```

```
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6671.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1516 riscv     20   0  370548  24640    712 S 194.4   0.3   0:15.25 lbzip2
   1515 riscv     20   0  222600   2848   2584 S   2.3   0.0   0:00.24 tar
   1523 riscv     20   0  224504   3032   2588 R   1.3   0.0   0:00.10 top
    918 riscv     20   0  641328  36348  24300 S   0.3   0.5   0:07.09 xfdeskt+
    921 riscv     20   0  888084  27972  21720 S   0.3   0.4   0:05.73 panel-8+
   1210 riscv     20   0   12316   4536   3064 S   0.3   0.1   0:00.07 sshd
      1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.51 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.83 ksoftir+
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.67 rcu_sch+
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.03 migrati+
[riscv@fedora-starfive ~]$ ps -o thcount 1516
THCNT
    6
[riscv@fedora-starfive ~]$
```

```
compress-program='lbzip2 -n3'
people.csv
^C

real    0m15.146s
user    0m0.027s
sys     0m0.389s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n3'
people.csv
^C

real    6m12.394s
user    0m0.192s
sys     0m8.184s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n3'
people.csv

real    7m53.566s
user    15m18.968s
sys     0m19.955s
[riscv@fedora-starfive Desktop]$
```

4 Threads


TheHWcave's AC USB Powermeter
Connect
x1
Rec
1s
riscv@fedora-starfive:~
riscv@fedora-starfive:~/Desktop

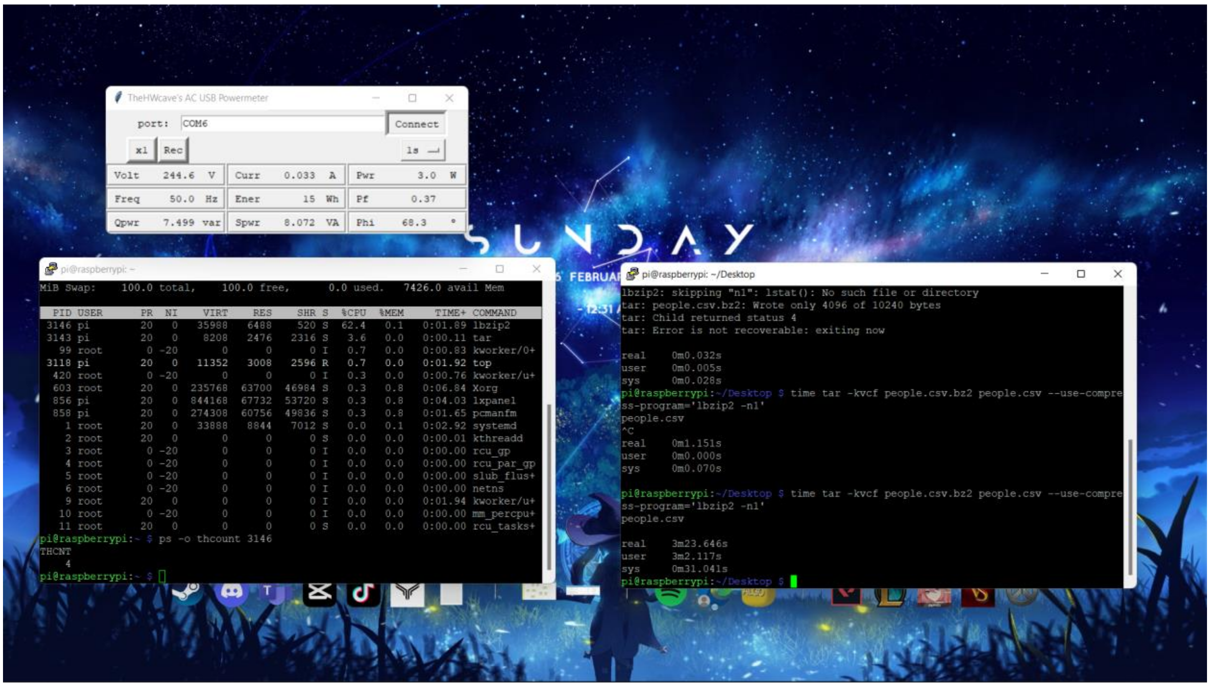
```
port: COM6
Volt   242.7  V    Curr   0.042  A    Pwr      4.5  W
Freq    50.0  Hz   Ener      10  Wh   Pf      0.44
Qpwr   9.154  var  Spwr  10.193  VA   Phi     63.9  °
```

```
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6666.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1614 riscv     20   0  444280  27260    636 S 175.0   0.4   0:07.34 lbzip2
   1623 riscv     20   0  224504   3120   2680 R  10.0   0.0   0:00.06 top
      1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.60 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.89 ksoftir+
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.81 rcu_sch+
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.05 migrati+
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     18 root      rt   0       0      0      0 S   0.0   0.0   0:00.06 migrati+
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.71 ksoftir+
[riscv@fedora-starfive ~]$ ps -o thcount 1614
THCNT
    7
[riscv@fedora-starfive ~]$
```

```
sys     0m0.389s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n3'
people.csv
^C

real    6m12.394s
user    0m0.192s
sys     0m8.184s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n3'
people.csv

real    7m53.566s
user    15m18.968s
sys     0m19.955s
[riscv@fedora-starfive Desktop]$ time tar -kvcf people.csv.bz2 people.csv --use-
compress-program='lbzip2 -n4'
people.csv

real    8m20.169s
user    16m13.764s
sys     0m18.248s
[riscv@fedora-starfive Desktop]$
```

Lbzip2 Dataset Compression performed by RPi 4B:

1 Thread

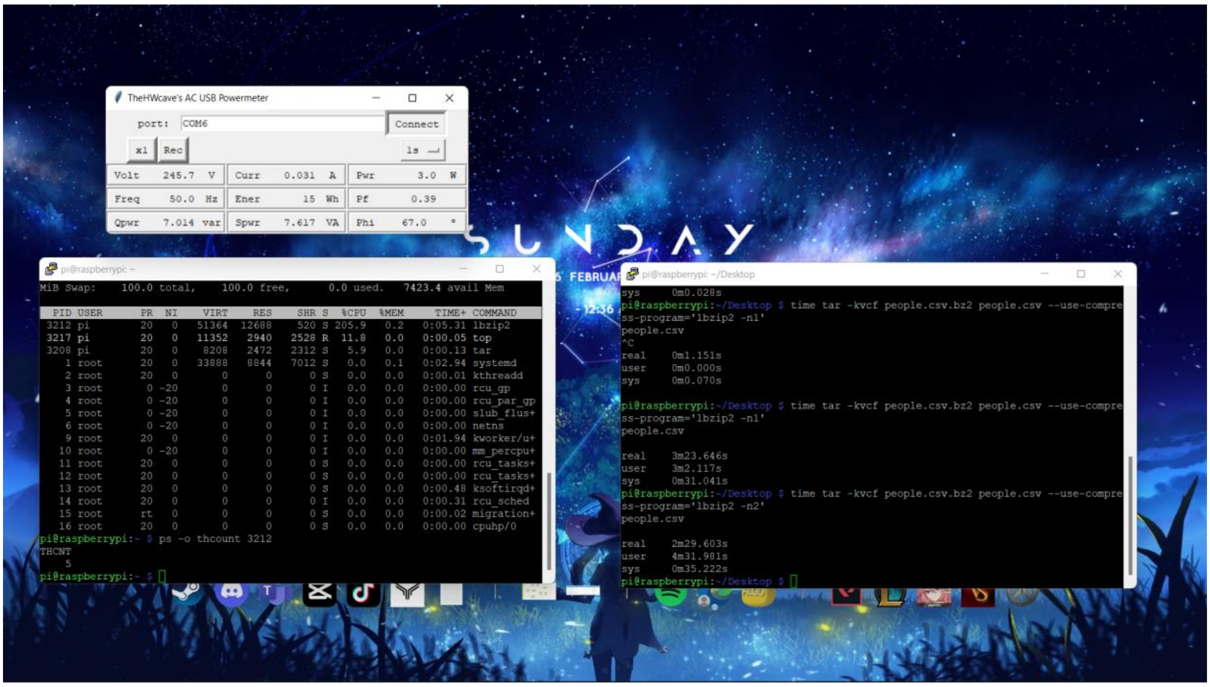


```
port: COM6
Volt   244.6  V    Curr  0.033  A    Pwr    3.0  W
Freq    50.0  Hz   Ener     15  Wh   Pf    0.37
Qpwr   7.499  var  Spwr  8.072  VA   Phi   68.3  °
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7426.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 3146 pi        20   0   35988   6488    520 S  62.4   0.1   0:01.89 lbzip2
 3143 pi        20   0    8208   2476   2316 S   3.6   0.0   0:00.11 tar
   99 root       0 -20       0      0      0 I   0.7   0.0   0:00.83 kworker/0+
 3118 pi        20   0   11352   3008   2596 R   0.7   0.0   0:01.92 top
  420 root       0 -20       0      0      0 I   0.3   0.0   0:00.76 kworker/u+
  603 root      20   0  235768  63700  46984 S   0.3   0.8   0:06.84 Xorg
  856 pi        20   0  844168  67732  53720 S   0.3   0.8   0:04.03 lxpanel
  858 pi        20   0  274308  60756  49836 S   0.3   0.8   0:01.65 pcmanfm
    1 root      20   0   33888   8844   7012 S   0.0   0.1   0:02.92 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.94 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
pi@raspberrypi:~ $ ps -o thcount 3146
THCNT
    4
pi@raspberrypi:~ $ 
```

```
lbzip2: skipping "n1": lstat(): No such file or directory
tar: people.csv.bz2: Wrote only 4096 of 10240 bytes
tar: Child returned status 4
tar: Error is not recoverable: exiting now

real    0m0.032s
user    0m0.005s
sys     0m0.028s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n1'
people.csv
^C
real    0m1.151s
user    0m0.000s
sys     0m0.070s

pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n1'
people.csv

real    3m23.646s
user    3m2.117s
sys     0m31.041s
pi@raspberrypi:~/Desktop $ 
```

2 Threads

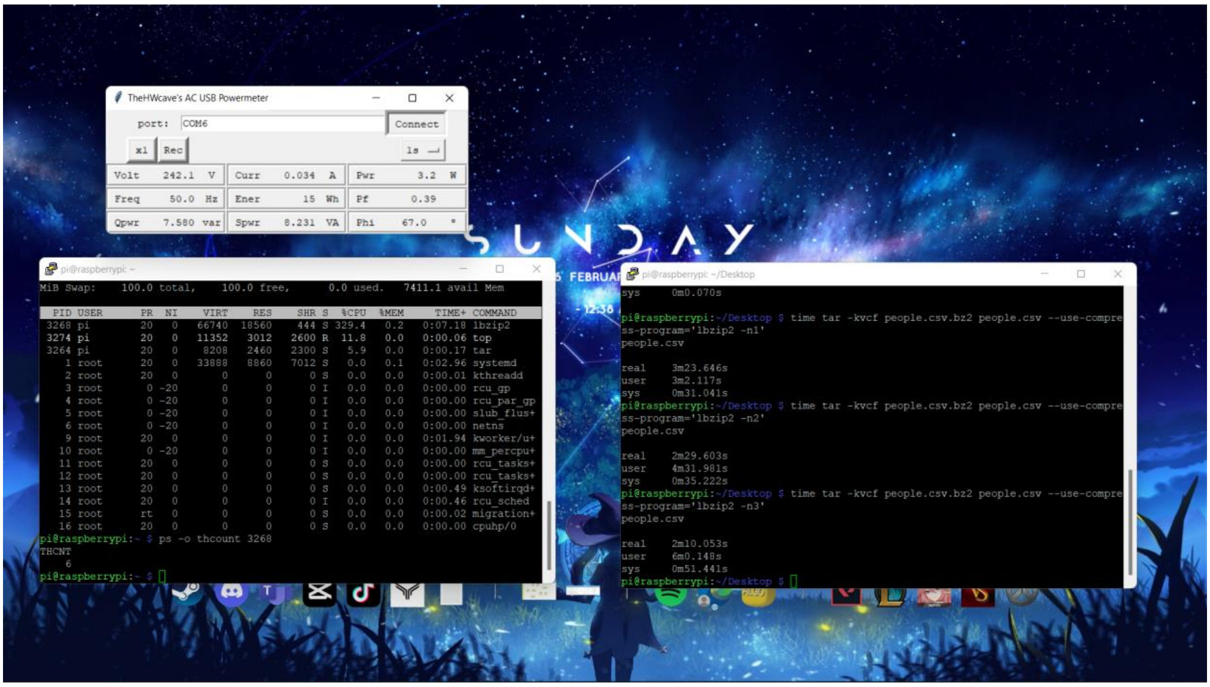


```
port: COM6
Volt   245.7  V    Curr  0.031  A    Pwr    3.0  W
Freq    50.0  Hz   Ener     15  Wh   Pf    0.39
Qpwr   7.014  var  Spwr  7.617  VA   Phi   67.0  °
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7423.4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 3212 pi        20   0   51364  12688    520 S 205.9   0.2   0:05.31 lbzip2
 3217 pi        20   0   11352   2940   2528 R  11.8   0.0   0:00.05 top
 3208 pi        20   0    8208   2472   2312 S   5.9   0.0   0:00.13 tar
    1 root      20   0   33888   8844   7012 S   0.0   0.1   0:02.94 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.94 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.48 ksoftirqd+
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.31 rcu_sched
   15 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
pi@raspberrypi:~ $ ps -o thcount 3212
THCNT
    5
pi@raspberrypi:~ $ 
```

```
sys     0m0.028s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n1'
people.csv
^C
real    0m1.151s
user    0m0.000s
sys     0m0.070s

pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n1'
people.csv

real    3m23.646s
user    3m2.117s
sys     0m31.041s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n2'
people.csv

real    2m29.603s
user    4m31.981s
sys     0m35.222s
pi@raspberrypi:~/Desktop $ 
```

3 Threads

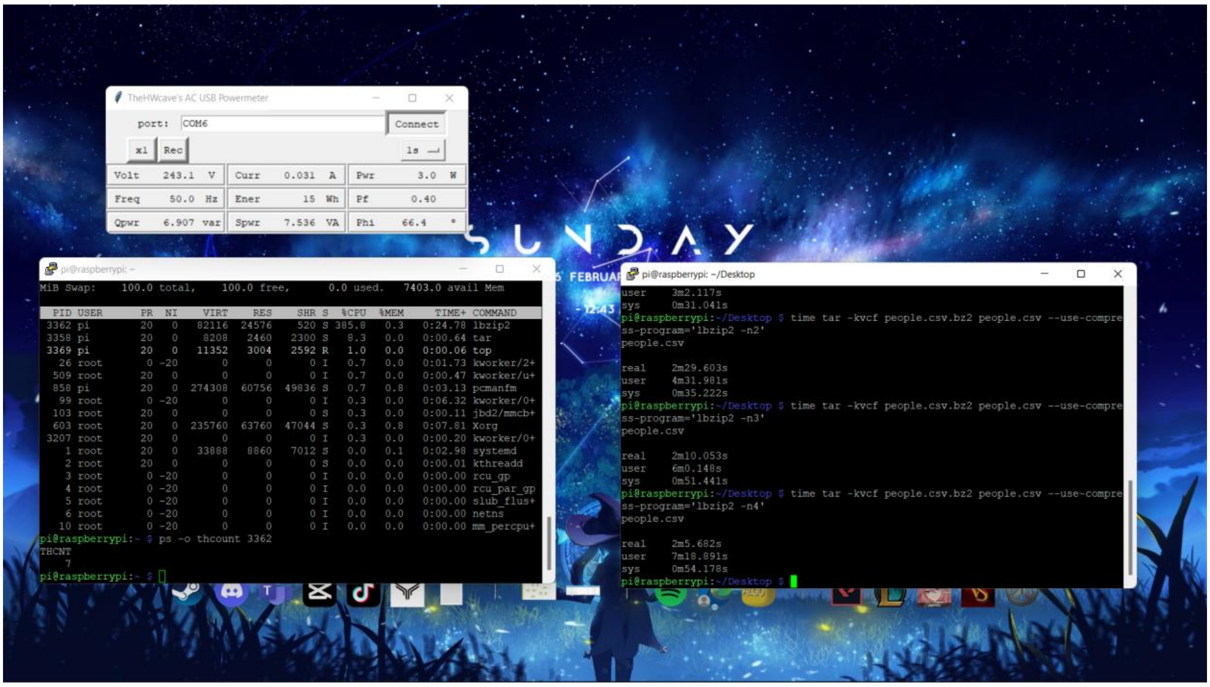


```
port: COM6
Volt    242.1   V | Curr   0.034  A | Pwr     3.2  W
Freq     50.0  Hz | Ener      15 Wh | Pf     0.39
Qpwr    7.580 var | Spwr   8.231 VA | Phi    67.0  °
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7411.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 3268 pi        20   0   66740  18560    444 S 329.4   0.2   0:07.18 lbzip2
 3274 pi        20   0   11352   3012   2600 R  11.8   0.0   0:00.06 top
 3264 pi        20   0    8208   2460   2300 S   5.9   0.0   0:00.17 tar
    1 root      20   0   33888   8860   7012 S   0.0   0.1   0:02.96 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.94 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.49 ksoftirqd+
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.46 rcu_sched
   15 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
pi@raspberrypi:~ $ ps -o thcount 3268
THCNT
    6
pi@raspberrypi:~ $
```

```
sys     0m0.070s

pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n1'
people.csv

real    3m23.646s
user    3m2.117s
sys     0m31.041s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n2'
people.csv

real    2m29.603s
user    4m31.981s
sys     0m35.222s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n3'
people.csv

real    2m10.053s
user    6m0.148s
sys     0m51.441s
pi@raspberrypi:~/Desktop $
```

4 Threads



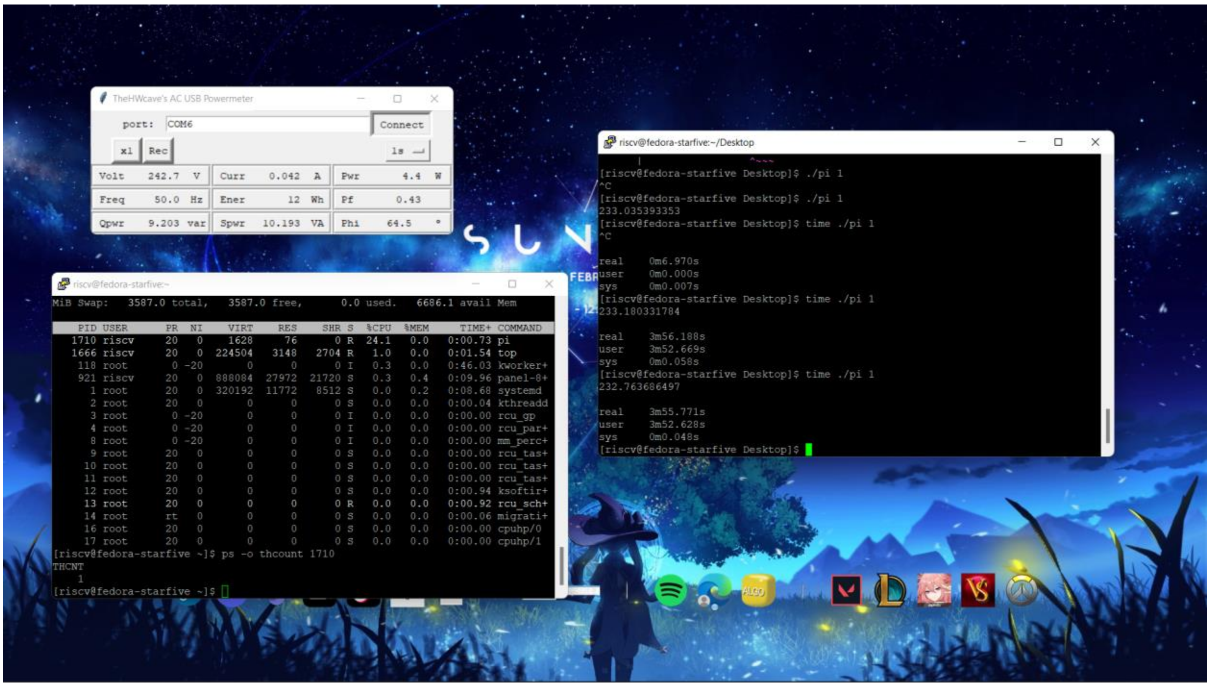
```
port: COM6
Volt    243.1   V | Curr   0.031  A | Pwr     3.0  W
Freq     50.0  Hz | Ener      15 Wh | Pf     0.40
Qpwr    6.907 var | Spwr   7.536 VA | Phi    66.4  °
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7403.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 3362 pi        20   0   82116  24576    520 S 385.8   0.3   0:24.78 lbzip2
 3358 pi        20   0    8208   2460   2300 S   8.3   0.0   0:00.64 tar
 3369 pi        20   0   11352   3004   2592 R   1.0   0.0   0:00.06 top
   26 root       0 -20       0      0      0 I   0.7   0.0   0:01.73 kworker/2+
  509 root      20   0       0      0      0 I   0.7   0.0   0:00.47 kworker/u+
  858 pi        20   0  274308  60756  49836 S   0.7   0.8   0:03.13 pcmanfm
   99 root       0 -20       0      0      0 I   0.3   0.0   0:06.32 kworker/0+
  103 root      20   0       0      0      0 S   0.3   0.0   0:00.11 jbd2/mmcb+
  603 root      20   0  235760  63760  47044 S   0.3   0.8   0:07.81 Xorg
 3207 root      20   0       0      0      0 I   0.3   0.0   0:00.20 kworker/0+
    1 root      20   0   33888   8860   7012 S   0.0   0.1   0:02.98 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
pi@raspberrypi:~ $ ps -o thcount 3362
THCNT
    7
pi@raspberrypi:~ $
```

```
user    3m2.117s
sys     0m31.041s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n2'
people.csv

real    2m29.603s
user    4m31.981s
sys     0m35.222s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n3'
people.csv

real    2m10.053s
user    6m0.148s
sys     0m51.441s
pi@raspberrypi:~/Desktop $ time tar -kvcf people.csv.bz2 people.csv --use-compre
ss-program='lbzip2 -n4'
people.csv

real    2m5.682s
user    7m18.891s
sys     0m54.178s
pi@raspberrypi:~/Desktop $
```

Pi.c microbenchmark performed by VisionFive V1 Board:

1 Thread

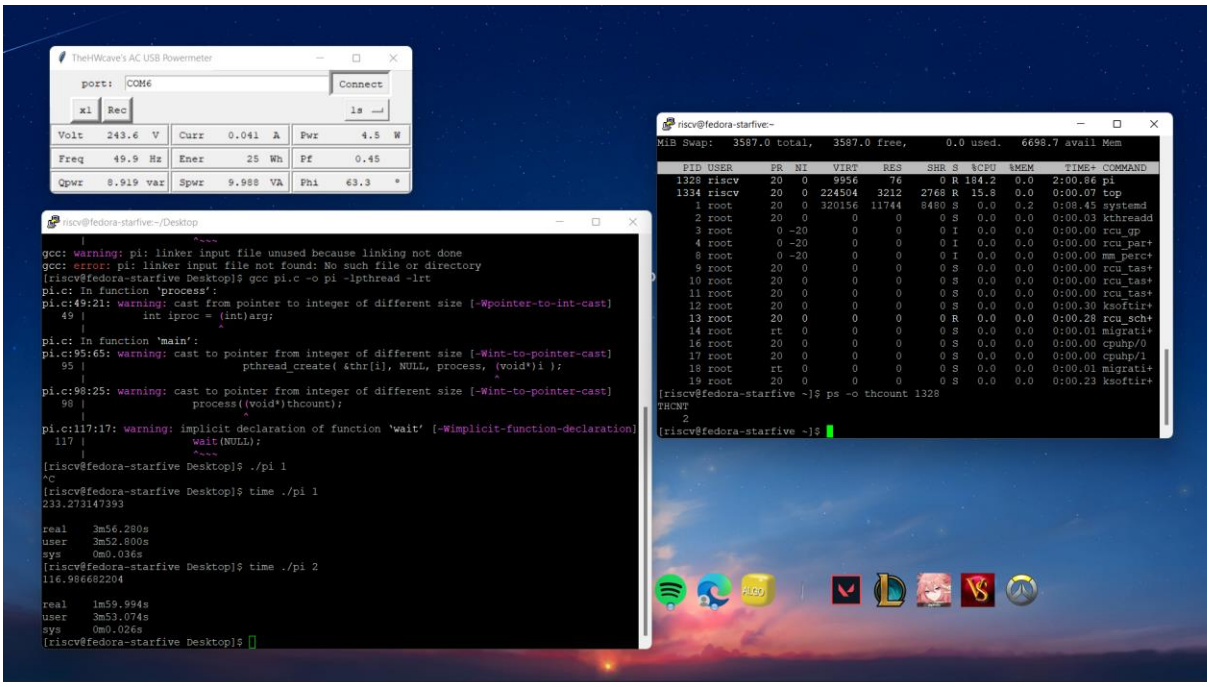

TheHWcave's AC USB Powermeter
Connect
x1
Rec
1s
riscv@fedora-starfive:~/Desktop
riscv@fedora-starfive:~

```
port: COM6
Volt   242.7   V   Curr    0.042  A   Pwr       4.4   W
Freq    50.0  Hz   Ener       12  Wh  Pf        0.43
Qpwr   9.203 var   Spwr   10.193  VA  Phi       64.5  °
```

```
      |                 ^~~~
[riscv@fedora-starfive Desktop]$ ./pi 1
^C
[riscv@fedora-starfive Desktop]$ ./pi 1
233.035393353
[riscv@fedora-starfive Desktop]$ time ./pi 1
^C

real    0m6.970s
user    0m0.000s
sys     0m0.007s
[riscv@fedora-starfive Desktop]$ time ./pi 1
233.180331784

real    3m56.188s
user    3m52.669s
sys     0m0.058s
[riscv@fedora-starfive Desktop]$ time ./pi 1
232.763686497

real    3m55.771s
user    3m52.628s
sys     0m0.048s
[riscv@fedora-starfive Desktop]$ 
```

```
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6686.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1710 riscv     20   0    1628     76      0 R  24.1   0.0   0:00.73 pi
   1666 riscv     20   0  224504   3148   2704 R   1.0   0.0   0:01.54 top
    118 root       0 -20       0      0      0 I   0.3   0.0   0:46.03 kworker+
    921 riscv     20   0  888084  27972  21720 S   0.3   0.4   0:09.96 panel-8+
      1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.68 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.94 ksoftir+
     13 root      20   0       0      0      0 R   0.0   0.0   0:00.92 rcu_sch+
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.06 migrati+
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
[riscv@fedora-starfive ~]$ ps -o thcount 1710
THCNT
    1
[riscv@fedora-starfive ~]$ 
```

2 Threads

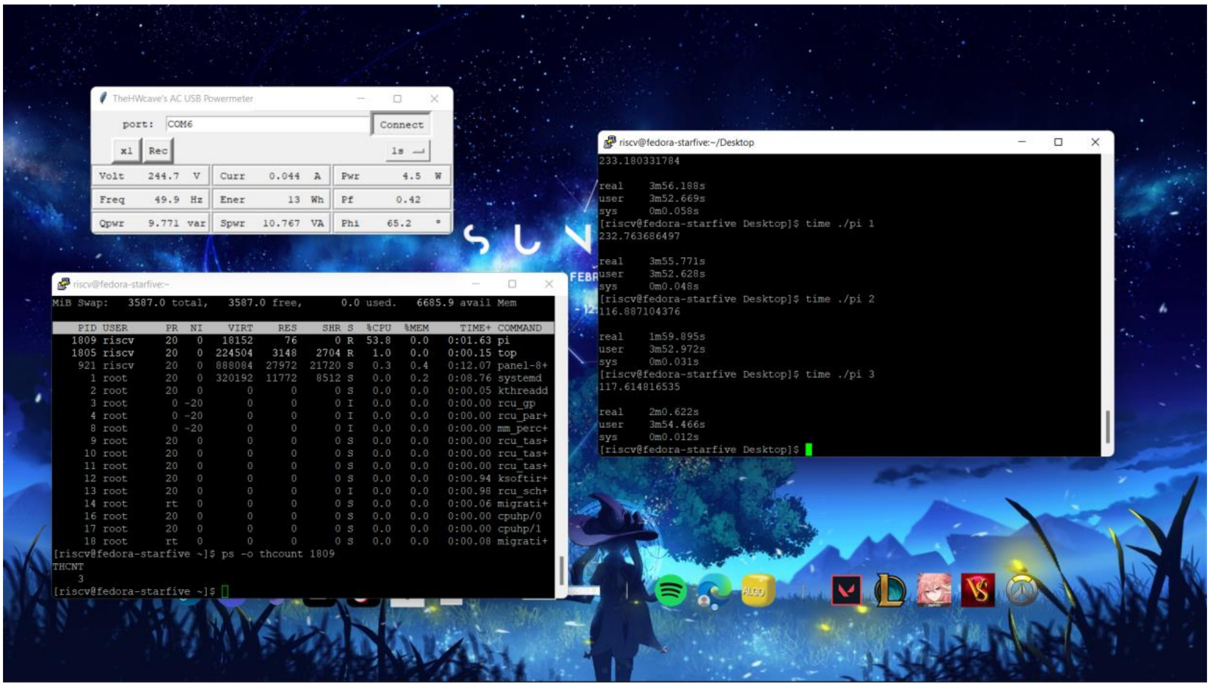

TheHWcave's AC USB Powermeter
Connect
x1
Rec
1s
riscv@fedora-starfive:~/Desktop
riscv@fedora-starfive:~

```
port: COM6
Volt   243.6   V   Curr    0.041  A   Pwr       4.5   W
Freq    49.9  Hz   Ener       25  Wh  Pf        0.45
Qpwr   8.919 var   Spwr    9.988  VA  Phi       63.3  °
```

```
      |                 ^~~~
gcc: warning: pi: linker input file unused because linking not done
gcc: error: pi: linker input file not found: No such file or directory
[riscv@fedora-starfive Desktop]$ gcc pi.c -o pi -lpthread -lrt
pi.c: In function 'process':
pi.c:49:21: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
   49 |          int iproc = (int)arg;
      |                      ^
pi.c: In function 'main':
pi.c:95:65: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
   95 |                      pthread_create( &thr[i], NULL, process, (void*)i );
      |                                                              ^
pi.c:98:25: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
   98 |                 process((void*)thcount);
      |                         ^
pi.c:117:17: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
  117 |                 wait(NULL);
      |                 ^~~~
[riscv@fedora-starfive Desktop]$ ./pi 1
^C
[riscv@fedora-starfive Desktop]$ time ./pi 1
233.273147393

real    3m56.280s
user    3m52.800s
sys     0m0.036s
[riscv@fedora-starfive Desktop]$ time ./pi 2
116.986682204

real    1m59.994s
user    3m53.074s
sys     0m0.026s
[riscv@fedora-starfive Desktop]$ 
```

```
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6698.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1328 riscv     20   0    9956     76      0 R 184.2   0.0   2:00.86 pi
   1334 riscv     20   0  224504   3212   2768 R  15.8   0.0   0:00.07 top
      1 root      20   0  320156  11744   8480 S   0.0   0.2   0:08.45 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.30 ksoftir+
     13 root      20   0       0      0      0 R   0.0   0.0   0:00.28 rcu_sch+
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migrati+
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     18 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migrati+
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.23 ksoftir+
[riscv@fedora-starfive ~]$ ps -o thcount 1328
THCNT
    2
[riscv@fedora-starfive ~]$ 
```

3 Threads

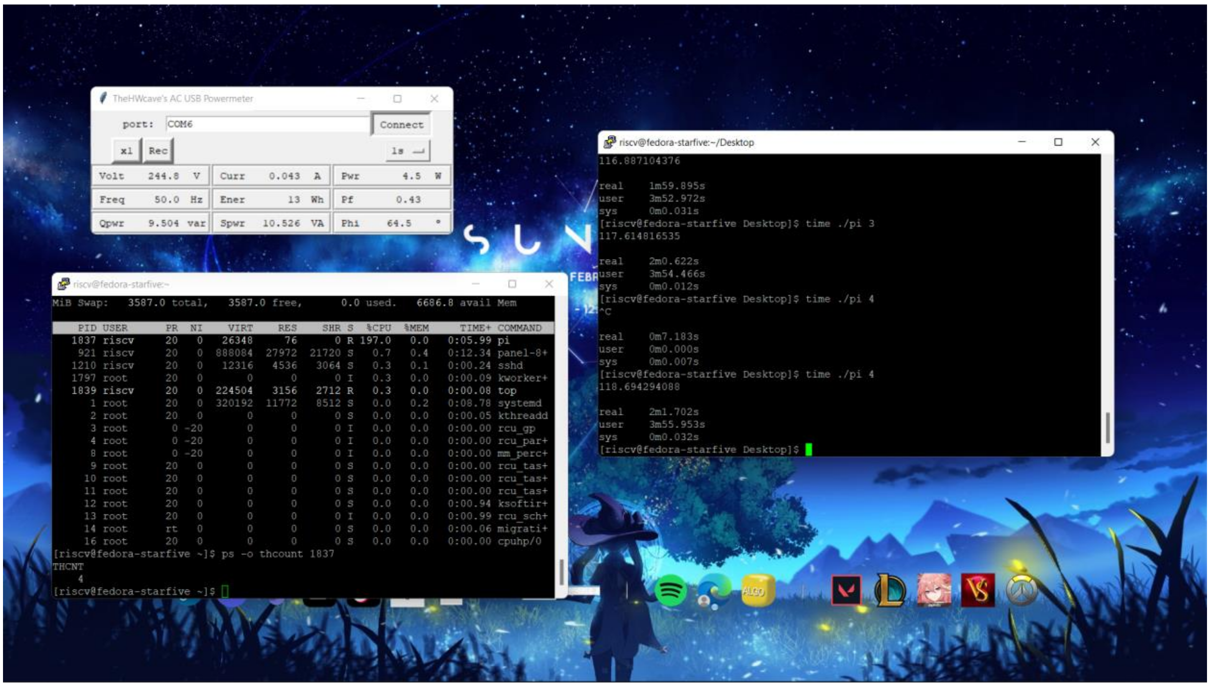

TheHWcave's AC USB Powermeter
Connect
x1
Rec
1s
riscv@fedora-starfive:~
riscv@fedora-starfive:~/Desktop

```
port: COM6
Volt    244.7  V    Curr    0.044  A    Pwr     4.5  W
Freq     49.9  Hz   Ener       13  Wh   Pf     0.42
Qpwr    9.771  var  Spwr   10.767  VA   Phi    65.2  °
```

```
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6685.9 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1809 riscv     20   0   18152     76      0 R  53.8   0.0   0:01.63 pi
 1805 riscv     20   0  224504   3148   2704 R   1.0   0.0   0:00.15 top
  921 riscv     20   0  888084  27972  21720 S   0.3   0.4   0:12.07 panel-8+
    1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.76 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
    9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.94 ksoftir+
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.98 rcu_sch+
   14 root      rt   0       0      0      0 S   0.0   0.0   0:00.06 migrati+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   18 root      rt   0       0      0      0 S   0.0   0.0   0:00.08 migrati+
[riscv@fedora-starfive ~]$ ps -o thcount 1809
THCNT
    3
[riscv@fedora-starfive ~]$
```

```
233.180331784

real    3m56.188s
user    3m52.669s
sys     0m0.058s
[riscv@fedora-starfive Desktop]$ time ./pi 1
232.763686497

real    3m55.771s
user    3m52.628s
sys     0m0.048s
[riscv@fedora-starfive Desktop]$ time ./pi 2
116.887104376

real    1m59.895s
user    3m52.972s
sys     0m0.031s
[riscv@fedora-starfive Desktop]$ time ./pi 3
117.614816535

real    2m0.622s
user    3m54.466s
sys     0m0.012s
[riscv@fedora-starfive Desktop]$
```

4 Threads


TheHWcave's AC USB Powermeter
Connect
x1
Rec
1s
riscv@fedora-starfive:~
riscv@fedora-starfive:~/Desktop

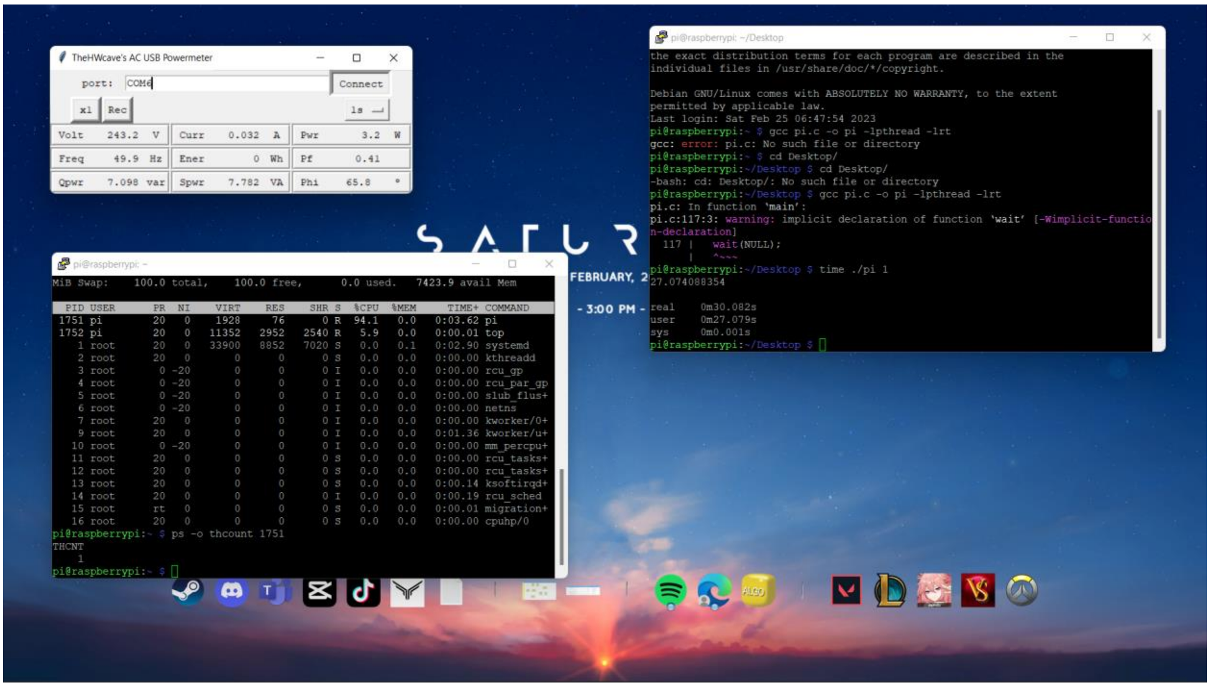
```
port: COM6
Volt    244.8  V    Curr    0.043  A    Pwr     4.5  W
Freq     50.0  Hz   Ener       13  Wh   Pf     0.43
Qpwr    9.504  var  Spwr   10.526  VA   Phi    64.5  °
```

```
MiB Swap:   3587.0 total,   3587.0 free,      0.0 used.   6686.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1837 riscv     20   0   26348     76      0 R 197.0   0.0   0:05.99 pi
  921 riscv     20   0  888084  27972  21720 S   0.7   0.4   0:12.34 panel-8+
 1210 riscv     20   0   12316   4536   3064 S   0.3   0.1   0:00.24 sshd
 1797 root      20   0       0      0      0 I   0.3   0.0   0:00.09 kworker+
 1839 riscv     20   0  224504   3156   2712 R   0.3   0.0   0:00.08 top
    1 root      20   0  320192  11772   8512 S   0.0   0.2   0:08.78 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.05 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par+
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_perc+
    9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tas+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.94 ksoftir+
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.99 rcu_sch+
   14 root      rt   0       0      0      0 S   0.0   0.0   0:00.06 migrati+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
[riscv@fedora-starfive ~]$ ps -o thcount 1837
THCNT
    4
[riscv@fedora-starfive ~]$
```

```
116.887104376

real    1m59.895s
user    3m52.972s
sys     0m0.031s
[riscv@fedora-starfive Desktop]$ time ./pi 3
117.614816535

real    2m0.622s
user    3m54.466s
sys     0m0.012s
[riscv@fedora-starfive Desktop]$ time ./pi 4
^C

real    0m7.183s
user    0m0.000s
sys     0m0.007s
[riscv@fedora-starfive Desktop]$ time ./pi 4
118.694294088

real    2m1.702s
user    3m55.953s
sys     0m0.032s
[riscv@fedora-starfive Desktop]$
```

Pi.c microbenchmark performed by RPi 4B:

1 Thread

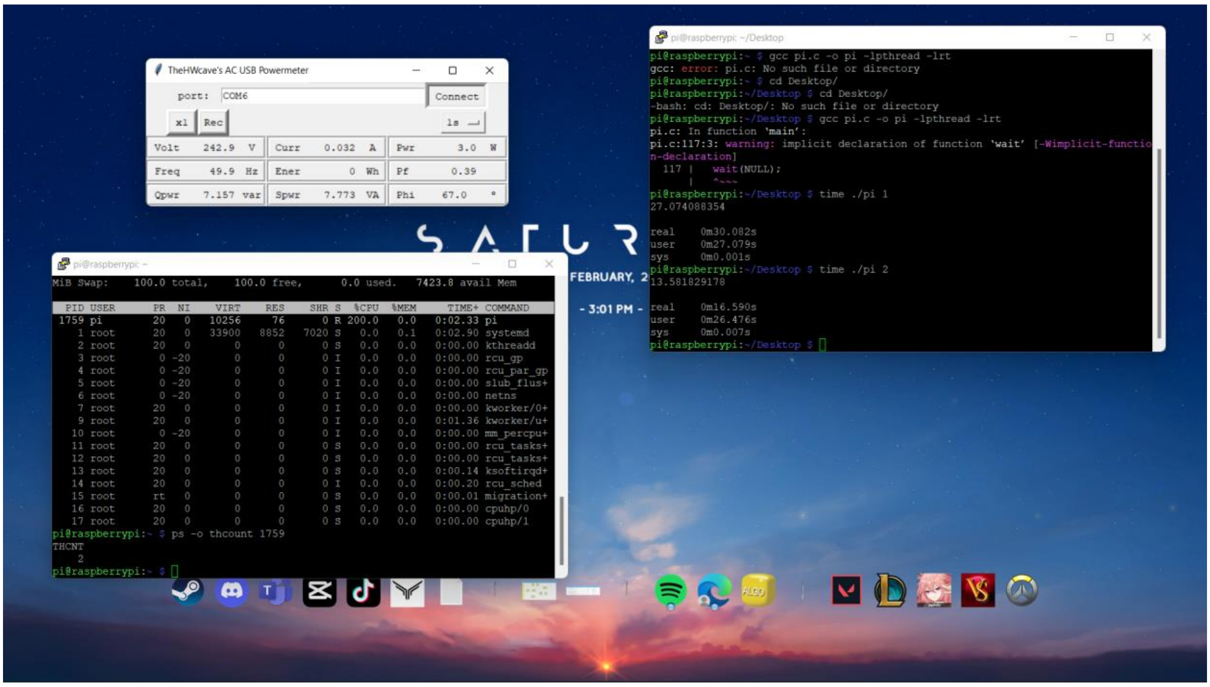


```
port: COM6
Volt    243.2   V    Curr    0.032   A    Pwr      3.2   W
Freq     49.9  Hz    Ener        0  Wh    Pf      0.41
Qpwr    7.098 var    Spwr    7.782  VA    Phi     65.8   °
```

```
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Feb 25 06:47:54 2023
pi@raspberrypi:~ $ gcc pi.c -o pi -lpthread -lrt
gcc: error: pi.c: No such file or directory
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ cd Desktop/
-bash: cd: Desktop/: No such file or directory
pi@raspberrypi:~/Desktop $ gcc pi.c -o pi -lpthread -lrt
pi.c: In function 'main':
pi.c:117:3: warning: implicit declaration of function 'wait' [-Wimplicit-functio
n-declaration]
  117 |   wait(NULL);
      |   ^~~~
pi@raspberrypi:~/Desktop $ time ./pi 1
27.074088354

real    0m30.082s
user    0m27.079s
sys     0m0.001s
pi@raspberrypi:~/Desktop $ 
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7423.9 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1751 pi        20   0    1928     76      0 R  94.1   0.0   0:03.62 pi
 1752 pi        20   0   11352   2952   2540 R   5.9   0.0   0:00.01 top
    1 root      20   0   33900   8852   7020 S   0.0   0.1   0:02.90 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0+
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.36 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.14 ksoftirqd+
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.19 rcu_sched
   15 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
pi@raspberrypi:~ $ ps -o thcount 1751
THCNT
    1
pi@raspberrypi:~ $ 
```

2 Threads

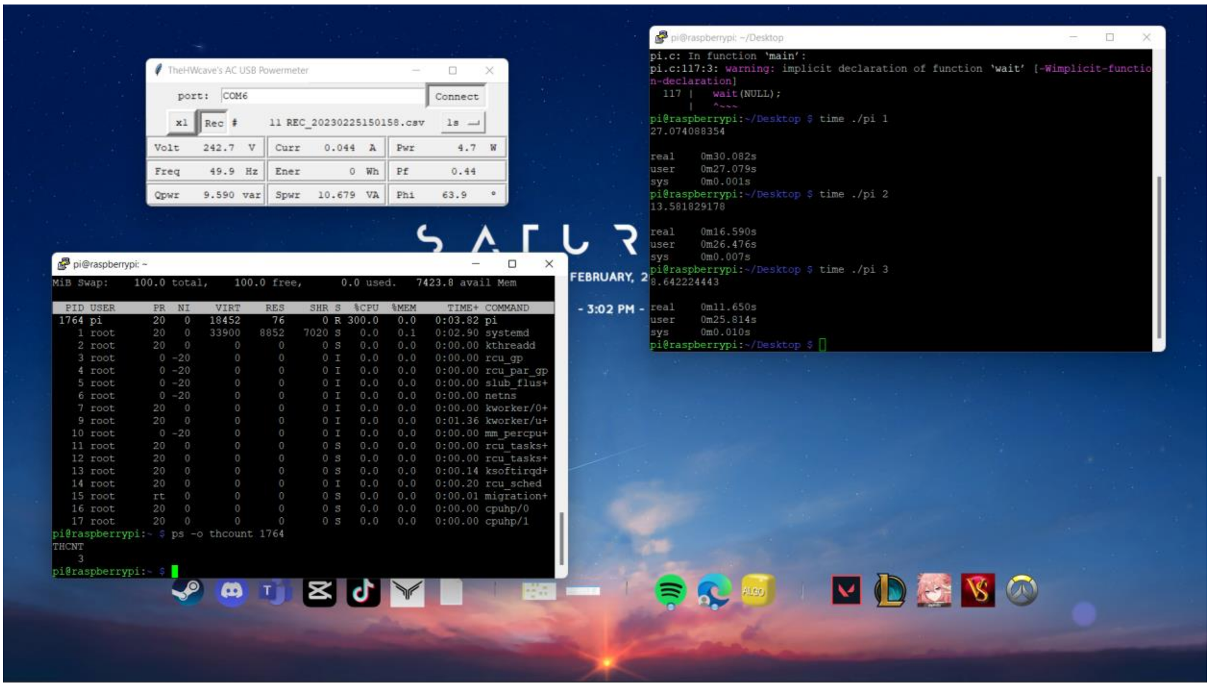


```
port: COM6
Volt    242.9   V    Curr    0.032   A    Pwr      3.0   W
Freq     49.9  Hz    Ener        0  Wh    Pf      0.39
Qpwr    7.157 var    Spwr    7.773  VA    Phi     67.0   °
```

```
pi@raspberrypi:~ $ gcc pi.c -o pi -lpthread -lrt
gcc: error: pi.c: No such file or directory
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ cd Desktop/
-bash: cd: Desktop/: No such file or directory
pi@raspberrypi:~/Desktop $ gcc pi.c -o pi -lpthread -lrt
pi.c: In function 'main':
pi.c:117:3: warning: implicit declaration of function 'wait' [-Wimplicit-functio
n-declaration]
  117 |   wait(NULL);
      |   ^~~~
pi@raspberrypi:~/Desktop $ time ./pi 1
27.074088354

real    0m30.082s
user    0m27.079s
sys     0m0.001s
pi@raspberrypi:~/Desktop $ time ./pi 2
13.581829178

real    0m16.590s
user    0m26.476s
sys     0m0.007s
pi@raspberrypi:~/Desktop $ 
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7423.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1759 pi        20   0   10256     76      0 R 200.0   0.0   0:02.33 pi
    1 root      20   0   33900   8852   7020 S   0.0   0.1   0:02.90 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0+
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.36 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.14 ksoftirqd+
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.20 rcu_sched
   15 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
pi@raspberrypi:~ $ ps -o thcount 1759
THCNT
    2
pi@raspberrypi:~ $ 
```

3 Threads

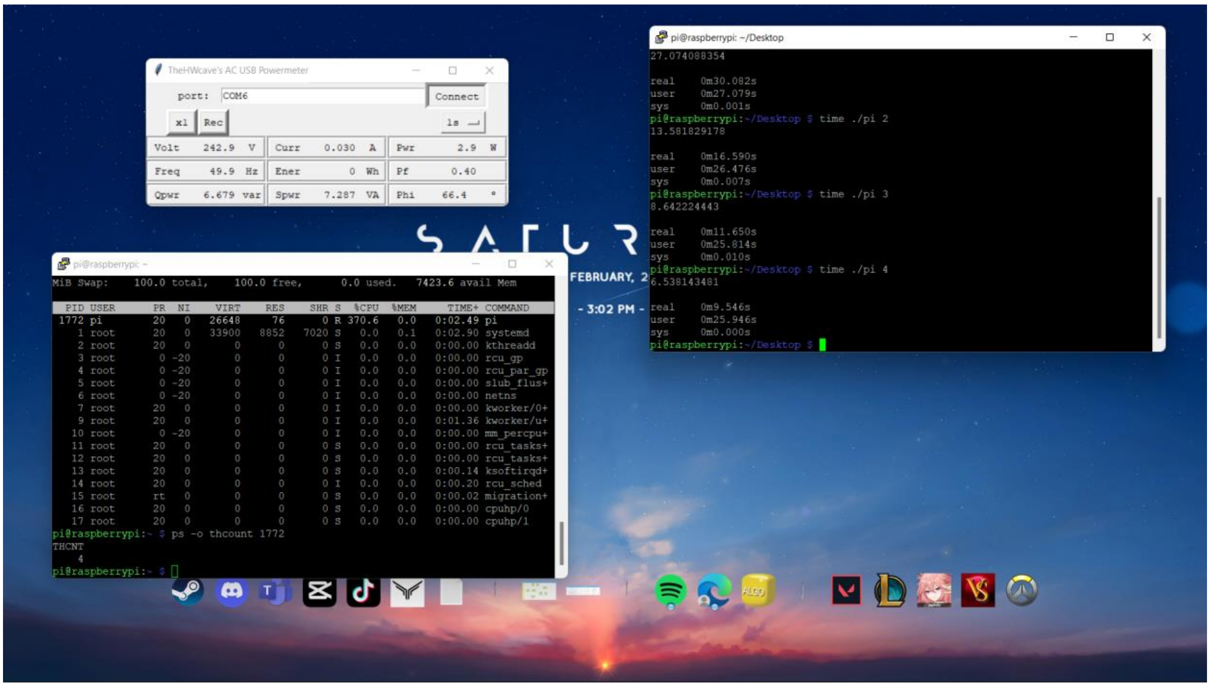


```
port: COM6                                   Connect
 xl  Rec #      11 REC_20230225150158.csv      1s
Volt    242.7   V | Curr    0.044   A | Pwr       4.7   W
Freq     49.9  Hz | Ener        0  Wh | Pf       0.44
Qpwr    9.590 var | Spwr   10.679  VA | Phi      63.9   °
```

```
pi.c: In function ‘main’:
pi.c:117:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-functio
n-declaration]
  117 |   wait(NULL);
      |   ^~~~
pi@raspberrypi:~/Desktop $ time ./pi 1
27.074088354

real    0m30.082s
user    0m27.079s
sys     0m0.001s
pi@raspberrypi:~/Desktop $ time ./pi 2
13.581829178

real    0m16.590s
user    0m26.476s
sys     0m0.007s
pi@raspberrypi:~/Desktop $ time ./pi 3
8.642224443

real    0m11.650s
user    0m25.814s
sys     0m0.010s
pi@raspberrypi:~/Desktop $ 
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7423.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1764 pi        20   0   18452     76      0 R 300.0   0.0   0:03.82 pi
    1 root      20   0   33900   8852   7020 S   0.0   0.1   0:02.90 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0+
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.36 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.14 ksoftirqd+
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.20 rcu_sched
   15 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
pi@raspberrypi:~ $ ps -o thcount 1764
THCNT
    3
pi@raspberrypi:~ $ 
```

4 Threads



```
port: COM6                                   Connect
 xl  Rec                                       1s
Volt    242.9   V | Curr    0.030   A | Pwr       2.9   W
Freq     49.9  Hz | Ener        0  Wh | Pf       0.40
Qpwr    6.679 var | Spwr    7.287  VA | Phi      66.4   °
```

```
27.074088354

real    0m30.082s
user    0m27.079s
sys     0m0.001s
pi@raspberrypi:~/Desktop $ time ./pi 2
13.581829178

real    0m16.590s
user    0m26.476s
sys     0m0.007s
pi@raspberrypi:~/Desktop $ time ./pi 3
8.642224443

real    0m11.650s
user    0m25.814s
sys     0m0.010s
pi@raspberrypi:~/Desktop $ time ./pi 4
6.538143481

real    0m9.546s
user    0m25.946s
sys     0m0.000s
pi@raspberrypi:~/Desktop $ 
```

```
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   7423.6 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1772 pi        20   0   26648     76      0 R 370.6   0.0   0:02.49 pi
    1 root      20   0   33900   8852   7020 S   0.0   0.1   0:02.90 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flus+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0+
    9 root      20   0       0      0      0 I   0.0   0.0   0:01.36 kworker/u+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.14 ksoftirqd+
   14 root      20   0       0      0      0 I   0.0   0.0   0:00.20 rcu_sched
   15 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration+
   16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
pi@raspberrypi:~ $ ps -o thcount 1772
THCNT
    4
pi@raspberrypi:~ $ 
```

APPENDIX C – WEEKLY LOG

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Perform the all the tests in FYP1 on RPi 4B model and upload the result onto the openbenchmarking.org website.

2. WORK TO BE DONE

LwHBench benchmark
Conference paper preparation
Power consumption test
Thread performance by both board

3. PROBLEMS ENCOUNTERED

No problem encountered when benchmarking the Rpi 4B model.

4. SELF EVALUATION OF THE PROGRESS

Satisfied

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 3**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Research on linpack benchmarking install procedure.
Research on LwHBench Benchmark.

2. WORK TO BE DONE

LwHBench benchmark
Conference paper preparation
Power consumption test
Thread performance by both board
Linpack benchmarking on both board

3. PROBLEMS ENCOUNTERED

The documentation to install linpack benchmarking does not works on VisionFive V1 board. Need to find other way to install it.

4. SELF EVALUATION OF THE PROGRESS

Acceptable

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE [Please write the details of the work done in the last fortnight.] Tried to install the linpack benchmark using PTS and it shows that the VisionFive V1 board is incompatible with the test on PTS. Retrieve the power meter from supervisor to start the power consumption test.
2. WORK TO BE DONE LwHBench benchmark Conference paper preparation Power consumption test Thread performance by both board
3. PROBLEMS ENCOUNTERED The power meter driver cannot be used. Need to find an older version of driver from the internet.
4. SELF EVALUATION OF THE PROGRESS Good

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 5**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Wrote a python code to generate dataset with a 1GB file size for the use of dataset compression.
Perform dataset compression test using Lbzip2 on both board.

2. WORK TO BE DONE

LwHBench benchmark
Conference paper preparation
Power consumption test
Thread performance by both board

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

So far so good

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 6**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Generate bar chart that represents the result of dataset compression. It allows any public to have a better view to check on the result.
Try to perform sysbench on both board.

2. WORK TO BE DONE

LwHBench benchmark
Conference paper preparation
Power consumption test
Thread performance by both board
Machine learning test on both board.

3. PROBLEMS ENCOUNTERED

RISCV architecture is incompatible with sysbench. Therefore, need to find an alternative benchmark program to determine the thread performance.

4. SELF EVALUATION OF THE PROGRESS

Acceptable

Supervisor's signature

Student's signature

PLAGIARISM CHECK RESULT

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 7**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Manipulate on the number of working thread on lbzip2 to determine the scalability of thread performance.
Generate the bar chart of the thread performance together with power consumption.
Start to draft the content of conference paper.
Try to make compiling optimization on VisionFive V1 board.

2. WORK TO BE DONE

LwHBench benchmark
Conference paper finalization
Machine learning test on both board.
Perform pi calculation micro-benchmark for thread performance.

3. PROBLEMS ENCOUNTERED

No problem encountered.

4. SELF EVALUATION OF THE PROGRESS

Good

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Done the test on pi calculation on both board together with the power consumption recording.
Amend the result into the conference paper.

2. WORK TO BE DONE

LwHBench benchmark
Conference paper finalization
Machine learning test on both board.
Perform pi calculation micro-benchmark for thread performance.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

Slow progress due to assignments and midterms.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 9**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Research on LwHBench benchmark
Submitted the conference paper.
Perform partition resizing on VisionFive V1 board as I found that the disk space does not fully utilized by the system.

2. WORK TO BE DONE

LwHBench benchmark
Other machine learning test on both board.

3. PROBLEMS ENCOUNTERED

LwHBench dataset can only be unlocked using subscription account of IEEE dataport.

4. SELF EVALUATION OF THE PROGRESS

So far so good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 10**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Decide to give up on LwHBench benchmark.
Focus on machine learning test available on PTS.

2. WORK TO BE DONE

Machine learning test on both board.
OpenCV test, numpy benchmark, tnn benchmark
FYP report preparation.

3. PROBLEMS ENCOUNTERED

Machine learning takes a long period to finish the tests and it consumes high CPU usage which may make the board to have throttling problem.

4. SELF EVALUATION OF THE PROGRESS

Acceptable

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 11**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Done the OpenCV deep neural network test on both VisionFive V1 board and RPi 4b model.

2. WORK TO BE DONE

Machine learning test on both board.
OpenCV test, numpy benchmark, tnn benchmark
FYP report preparation.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

Good.

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 12**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Done the tnn benchmark on VisionFive V1 board.
Done the image processing and numpy benchmark on both board.
Start to write the content of FYP report.

2. WORK TO BE DONE

FYP report preparation.

3. PROBLEMS ENCOUNTERED

TNN machine learning benchmark cannot be performed by RPi 4B model. Need to exclude it from the project scope.

4. SELF EVALUATION OF THE PROGRESS

Satisfied.

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 13**
Student Name & ID: Kam Kai Heng 19ACB03056	
Supervisor: Ts. Wong Chee Siang	
Project Title: Performance Evaluation of Starfive VisionFive V1 Board	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

The content of FYP report had been finalized and ready to submit.

2. WORK TO BE DONE

FYP presentation preparation.

3. PROBLEMS ENCOUNTERED

No.

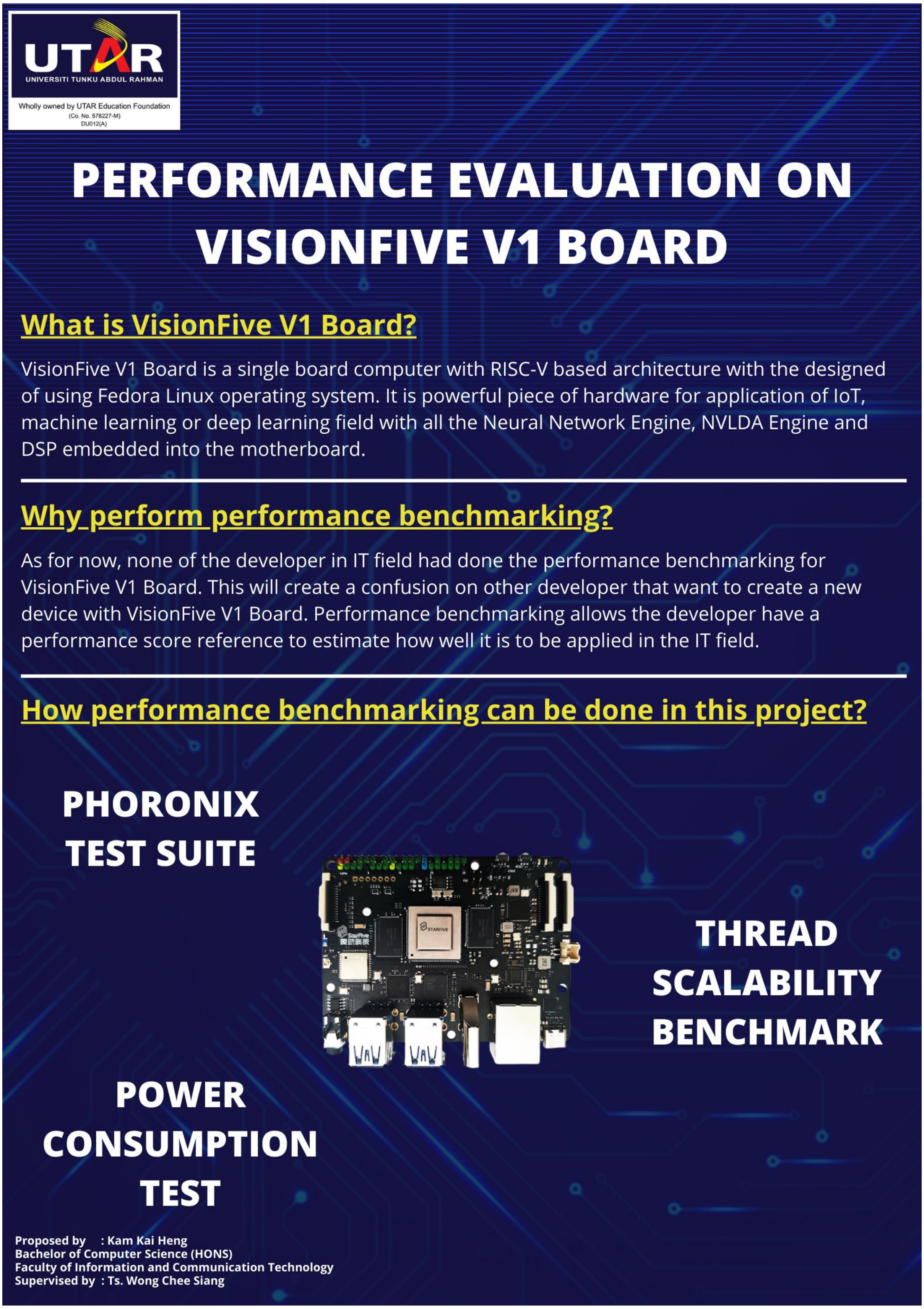
4. SELF EVALUATION OF THE PROGRESS

Acceptable.

Supervisor's signature

Student's signature

APPENDIX D - POSTER



UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

PERFORMANCE EVALUATION ON VISIONFIVE V1 BOARD

What is VisionFive V1 Board?

VisionFive V1 Board is a single board computer with RISC-V based architecture with the designed of using Fedora Linux operating system. It is powerful piece of hardware for application of IoT, machine learning or deep learning field with all the Neural Network Engine, NVLDA Engine and DSP embedded into the motherboard.

Why perform performance benchmarking?

As for now, none of the developer in IT field had done the performance benchmarking for VisionFive V1 Board. This will create a confusion on other developer that want to create a new device with VisionFive V1 Board. Performance benchmarking allows the developer have a performance score reference to estimate how well it is to be applied in the IT field.

How performance benchmarking can be done in this project?

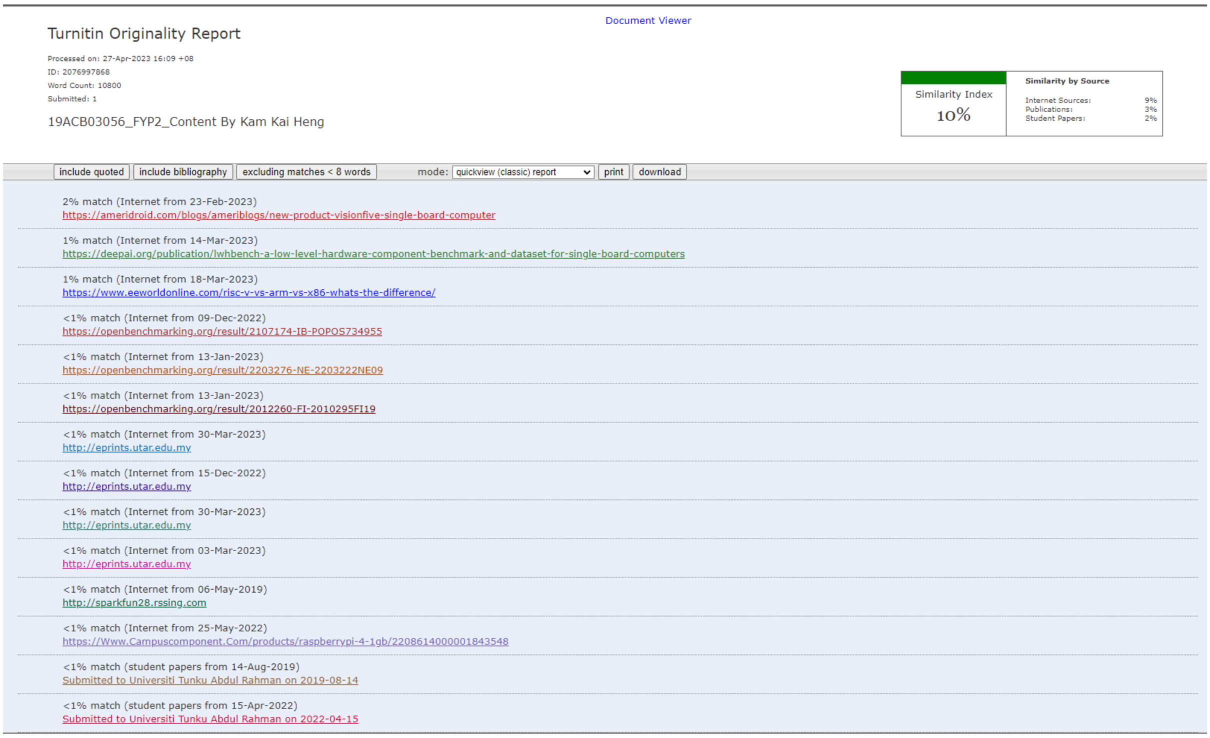
PHORONIX TEST SUITE

THREAD SCALABILITY BENCHMARK

POWER CONSUMPTION TEST

Proposed by : Kam Kai Heng
Bachelor of Computer Science (HONS)
Faculty of Information and Communication Technology
Supervised by : Ts. Wong Chee Siang

PLAGIARISM CHECK RESULT



Turnitin Originality Report

Processed on: 27-Apr-2023 16:09 +08
ID: 2076997868
Word Count: 10800
Submitted: 1

19ACB03056_FYP2_Content By Kam Kai Heng

Similarity Index	Similarity by Source	
10%	Internet Sources:	9%
	Publications:	3%
	Student Papers:	2%

mode: quickview (classic) report

2% match (Internet from 23-Feb-2023)
https://ameridroid.com/blogs/ameriblogs/new-product-visionfive-single-board-computer

1% match (Internet from 14-Mar-2023)
https://deepai.org/publication/lwhbench-a-low-level-hardware-component-benchmark-and-dataset-for-single-board-computers

1% match (Internet from 18-Mar-2023)
https://www.eeworldonline.com/risc-v-vs-arm-vs-x86-whats-the-difference/

<1% match (Internet from 09-Dec-2022)
https://openbenchmarking.org/result/2107174-IB-POPOS734955

<1% match (Internet from 13-Jan-2023)
https://openbenchmarking.org/result/2203276-NE-2203222NE09

<1% match (Internet from 13-Jan-2023)
https://openbenchmarking.org/result/2012260-FI-2010295FI19

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 15-Dec-2022)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 03-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 06-May-2019)
http://sparkfun28.rssing.com

<1% match (Internet from 25-May-2022)
https://Www.Campuscomponent.Com/products/raspberrypi-4-1gb/2208614000001843548

<1% match (student papers from 14-Aug-2019)
Submitted to Universiti Tunku Abdul Rahman on 2019-08-14

<1% match (student papers from 15-Apr-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-04-15

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

F C	KAM KAI HENG
I	010925-01-1289
Programme / Course	Computer Science
Title of Final Year Project	PERFORMANCE EVALUATION OF STARFIVE VISIONFIVE V1 BOARD

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: ___10___ % **Similarity by source** Internet Sources: _______9_______% Publications: _______3_______% Student Papers: _______2_______%	
Number of individual sources listed of more than 3% similarity: _____0____	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: ___Ts. Wong Chee Siang___

Date: ____28 APRIL 2023__________

____
Signature of Co-Supervisor

Name: __________________________

Date: __________________________

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB03056
Student Name	Kam Kai Heng
Supervisor Name	Ts. Wong Chee Siang

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____
(Signature of Student)
Date: 27/04/2023