



#### Document Identification

Identifier	E1451-20.00.TP-IEP.00
Issue	1.2
Date	2024-03-15

# COMPUTER EFFICIENCY AND PERFORMANCE

## Software Guide and Measurement Procedure – PCEET

---

## Revisions

Revision	Description	Date
0.1	Webinar draft	2020-07-07
0.2	Webinar draft 2 Updated version containing the following changes: <ul style="list-style-type: none"> <li>• Add newly supported power meter models</li> <li>• Describe local time synchronization</li> <li>• Fix various typos</li> </ul>	2020-10-26
0.3	Updated version containing the following changes: <ul style="list-style-type: none"> <li>• Replace UUT with EUT (Equipment under Test)</li> <li>• Add instructions for ChromeOS EUT setup</li> <li>• Update list of included worklets</li> </ul>	2021-11-29
0.4	Updated version containing the following changes: <ul style="list-style-type: none"> <li>• Update supported operating systems</li> <li>• Update instructions for ChromeOS EUT</li> <li>• Update list of included worklets</li> </ul>	2022-09-19
1.0	Updated version containing the following changes: <ul style="list-style-type: none"> <li>• Update supported operating systems</li> <li>• Update list of included worklets</li> <li>• Remove need for external monitor for EUT with integrated monitor</li> <li>• Update instructions for ChromeOS EUT</li> </ul>	2023-03-08
1.1	Updated version containing the following changes: <ul style="list-style-type: none"> <li>• Remove ChromeOS content for pending major changes</li> <li>• Update list of included worklets</li> <li>• Add troubleshooting section</li> <li>• Add Metric Calculation and Example Data section</li> </ul>	2023-10-12
1.2	Updated version containing the following changes: <ul style="list-style-type: none"> <li>• Power data and test results are automatically transferred between controller PC and EUT</li> <li>• Extend troubleshooting section</li> </ul>	2024-03-15

# Contents

1	Background, Context and Contacts	6
2	Acknowledgements	7
3	Disclaimer	7
4	Introduction	8
4.1	Purpose . . . . .	8
4.2	References . . . . .	8
4.3	Acronyms and Abbreviations . . . . .	8
5	Hardware Setup	9
5.1	Required Equipment . . . . .	9
5.2	Hardware Connection Procedure . . . . .	11
6	Creating USB Key	12
6.1	Microsoft Windows . . . . .	12
6.2	Apple MacOSX . . . . .	13
6.3	Linux . . . . .	14
6.4	Updating an existing Universal Serial Bus (USB) key . . . . .	14
7	Software Installation	15
7.1	General Remarks . . . . .	15
7.2	Controller Computer . . . . .	15
7.2.1	Microsoft Windows . . . . .	16
7.2.2	Apple MacOSX . . . . .	17
7.2.3	Linux . . . . .	17
7.2.4	Check the Connection to the Power Meter . . . . .	18
7.3	Equipment Under Test (EUT) Software Preparation . . . . .	19
7.3.1	Microsoft Windows . . . . .	20
7.3.2	Apple MacOSX . . . . .	21
7.3.3	Linux . . . . .	21
8	Test Suite Execution	22
8.1	Preparing the EUT . . . . .	22
8.2	Preparing the Controller Computer . . . . .	23
8.3	Running the EUT . . . . .	23
8.3.1	Check the Tests Progress . . . . .	24
8.3.2	Troubleshooting . . . . .	25

9	Complete List of Worklets	26
10	Results Analysis	27
10.1	Technical Details about the EUT(s)	28
10.2	Performance and Efficiency Comparison	28
10.3	Table of Performance Results	29
10.4	Detailed Test Results for each Worklet	29
10.5	Meta Values and System Power Monitor	31
10.6	Exporting Results	32
11	Efficiency Metric Calculation and Example Data	34
11.1	Calculation of Performance, Power and Efficiency Scores of a Worklet	34
11.2	Meta Values Calculation	36
11.3	Other Metrics	36
11.4	Sample Data for a Gaming Computer	36

## List of Figures

Figure 5.1	Test Hardware Setup	10
Figure 6.1	USB key configuration with Rufus	13
Figure 7.1	Controller Computer Graphical User Interface (GUI)	16
Figure 7.2	Power Meter found	18
Figure 7.3	Test Suite installation GUI	19
Figure 7.4	Test Suite installation GUI with terminal deployed	20
Figure 8.1	Test Suite execution GUI	24
Figure 10.1	List of example test results	27
Figure 10.2	Circular Efficiency Graph	28
Figure 10.3	Example Worklet performance	29
Figure 10.4	Example Worklet power	30
Figure 10.5	Example Worklet efficiency	30
Figure 10.6	Example Meta Efficiency	31
Figure 10.7	Export Data Menu	32
Figure 10.8	Imported Data into Spreadsheet	33

## List of Tables

Table 4.1	Reference Documents . . . . .	8
Table 5.1	Test equipment and requirements . . . . .	10
Table 9.1	Test Suite software Worklets and description . . . . .	26
Table 11.1	Worklet Measurement Details . . . . .	34
Table 11.2	Test Results of a Gaming PC . . . . .	37
Table 11.3	Meta Test Results of a Gaming PC . . . . .	37

# 1 Background, Context and Contacts

In 2019, the European Commission indicated that it would welcome support developing an on-mode test method for computers. Collaborative Labeling and Appliance Standards Program (CLASP) contacted the Commission and offered to assist on this topic, by developing a first version of a software tool that would quantify both energy consumption and performance of different computer configurations across several operating systems. The Commission then invited CLASP and its Team (GTD and Intertek UK) to present their concept to key stakeholders to determine whether this idea should be taken forward.

On Monday, 13 January 2020, a three-hour meeting/webinar was held at Directorate General Energy, Rue Jean-André de Mot 24, Brussels with more than 50 participants including DigitalEurope’s members and other experts. The conclusion of that meeting was an agreement for CLASP and its team to work on developing this software tool in consultation with DigitalEurope and other key stakeholders.

This document is part of the package of draft deliverables being shared with participants of the 7th July webinar for review. It encompasses the following: (1) downloading and installing the test suite; (2) test set-up and connecting the test equipment; (3) conducting the test on a computer; (4) exporting and interpreting the test results. Reviewers of the software tool are invited to provide comments using the online comment tool:

<https://redmine.gtd-gmbh.de/projects/computer-energy-efficiency-test/issues/new>

User accounts for the comment tool must be requested via Email to [Andoni.Arregui@gtd-gmbh.de](mailto:Andoni.Arregui@gtd-gmbh.de).

Following that review and comments received, CLASP/GTD/Intertek will work with this input to prepare a first draft of the software tool and deliver that to the Commission. This delivery will consist of all of the (uncompiled) software code, this test procedure document and anonymous test results being given to the Commission. The Commission has announced that it then intends to hold a Consultation Forum meeting on Computers and may present the tool for stakeholder review.

The PCEET software has been reviewed by Digital Europe Working Group. Digital Europe has contributed to the Test Procedure based on member’s feedback.

If you have any questions about the project or technical questions / issues that arise while conducting the test, please contact one of the following individuals:

	Technical / Software Issues	Project-related Questions
<b>Name</b>	Andoni Arregui	Nicole Kearney
<b>Organisation</b>	GTD GmbH	CLASP
<b>Email</b>	<a href="mailto:andoni.arregui@gtd-gmbh.de">andoni.arregui@gtd-gmbh.de</a>	<a href="mailto:nkearney@clasp.ngo">nkearney@clasp.ngo</a>
<b>Telephone</b>	+49 7544 964 4022	

## 2 Acknowledgements

CLASP, GTD and Intertek would like to warmly thank two companies who have kindly loaned equipment to us on very short notice to facilitate the development of this software test suite. We would like to thank Apple Computer, and especially Frank Lenderink and Thomas Barillot, for lending us a MacBook Pro. We would also like to thank the Yokogawa Test & Measurement Corporation, and especially Michael Rietvelt and Lee Thomas for lending us a Yokogawa WT310. We greatly appreciate their support because having this equipment available at GTD in Germany, enabled us to conduct parallel testing sequences with Intertek (UK) and thereby verify and confirm test results.

## 3 Disclaimer

While every precaution has been followed in the preparation of this document, including careful internal review by competent and experienced technical experts, this work does involve measurements of mains voltage and current in a test laboratory. The procedure set out in this document should only be attempted by competent, experienced technical experts who have worked with power measurements of computers or similar equipment in the past and are familiar with the best laboratory safety procedures and practices.

CLASP, GTD GmbH and Intertek do not accept any liability for any direct, indirect, punitive, incidental, special, or consequential damages arising out of, or in connection with, the use or misuse of any information provided in this report, the accompanying software and any other instructions or deliverables provided in association with this project. This information is provided as a free resource for measuring the energy efficiency and performance of computers and users of this test method acknowledge and agree that any use of this information is done at their own risk.

## 4 Introduction

### 4.1 Purpose

This document sets out the procedure to follow to test the energy efficiency and performance of a computer. The equipment necessary for conducting the testing is specified as well as all the steps related to hardware setup and software execution which need to be followed by the technician.

The last chapters give overviews over how the results produced by the software can be analyzed and how those results are calculated in the first place.

### 4.2 References

Table 4.1: Reference Documents

Reference	Identification	Issue	Title
RD01	Web Site	2014-08-20	Understanding And Performing Standby Power Measurements
RD02	IEC 62623	2022	Desktop and notebook computers - Measurement of energy consumption
RD03	TR-PCEEI-Metrics	2021-01-22	White Paper on the Definition of Efficiency Metrics for Computers

### 4.3 Acronyms and Abbreviations

A	Ampere
CLASP	Collaborative Labeling and Appliance Standards Program
EUT	Equipment Under Test
GUI	Graphical User Interface
HIB	Higher is better
LIB	Lower is better
OS	Operating System
PC	Personal Computer
USB	Universal Serial Bus
V	Volt



## 5 Hardware Setup

### 5.1 Required Equipment

The following list identifies the required equipment for testing:

- Regulated Power Supply (laboratory quality, as per EN 62623, see [RD02])
- Power meter (laboratory quality, as per EN 62623, see [RD02]); at this time the software is compatible with the following power meters:
  - Yokogawa WT310 and WT310E Power Meter (USB connection)
  - Yokogawa WT333 (USB connection)
  - Yokogawa WT500 (USB connection)
  - Hioki PW3335 (only serial connection, 38400 bps 8n1)
  - Hioki 3332 (only serial connection, 9600 bps 8n1)
  - Yokogawa WT3000 (only GPIB connection on Windows)
  - Yokogawa WT210 (only GPIB connection on Windows)
- Measurement Adapter to connect the computer being tested, also called the Equipment Under Test (EUT), and a power meter with power socket.
- EUT including an external monitor with 1080p resolution, mouse and keyboard for desktop computers. In case of an EUT with integrated monitor and peripherals (e.g. Laptop), the integrated monitor and peripherals can be used.
- Controller Computer (Windows, Mac or Linux Operating System (OS); see above for power meters only supported on Windows) to read power values from the power meter.
- For power meters connected via serial or GPIB: A serial to USB or GPIB to USB adapter is recommended. Make sure that the appropriate driver for your adapter is installed before running any parts of the test software. The following adapters have been confirmed working:
  - Keyspan USA-19HS serial to USB adapter
  - National Instruments GPIB-USB-HS GPIB to USB adapter with NI-VISA 20.0
- For power meters connected via USB: USB cable to connect the power meter to the Controller Computer. No previous driver installation is necessary.
- USB key containing the test software

Figure 5.1 presents a scheme of the equipment and the connections.

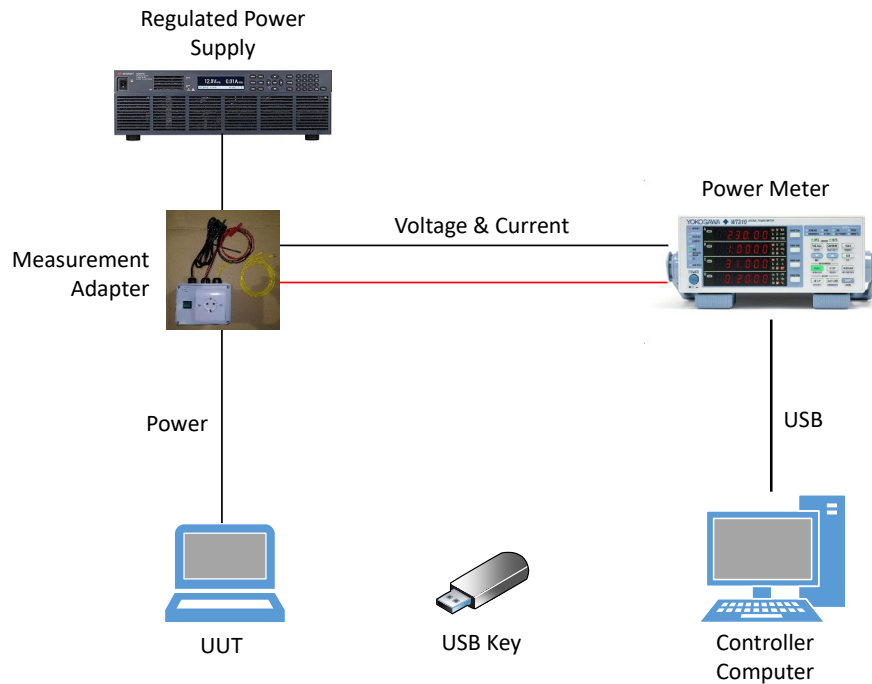
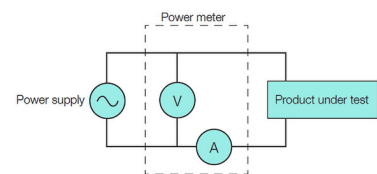


Figure 5.1: Test Hardware Setup

Table 5.1: Test equipment and requirements

Name and Description of Equipment	Requirements Defined	Notes
Power Meter	EN 62623, [RD02]	Power consumption measurement
Regulated Power Supply	EN 62623, [RD02]	Clean power for testing
Controller Computer	Two USB ports (2.0 or higher) for interface with power meter and USB key	Can be running Microsoft Windows 10 and 11 x86 64-bit, MacOSX 12 Intel or ARM or Ubuntu 22.04 Linux x86 64-bit or newer.
USB key	16 GB or larger	USB 3.0 or higher recommended
Measurement Adapter	A socket with separate measurement leads for voltage and current. See article in Power Electronics for circuit diagram.	See also [RD01].



## 5.2 Hardware Connection Procedure

When connecting all the elements, standard laboratory safety procedures must be followed to avoid risk to the technician as well as risk of damage to the test equipment and EUT from static electricity. The work surface must be connected to ground and the operator should use an anti-static wrist strap.

Connect the Controller Computer and the EUT to the Power Meter using the Measurement Adapter by carrying out the following steps:

- Make sure the Measurement Adapter is not connected to the Regulated Power Supply.
- Make sure the Power Meter is not plugged into a wall socket.
- Connect the Measurement Adapter to the Power Meter:
  - Connect the voltage sensing cables of the Measurement Adapter to the voltage input terminals located at the back of the Power Meter.
  - Connect the current cables to the current input terminals at the back of the Power Meter.
  - Consult the user's manual of your measurement adapter and power meter to verify the connections.
- Plug the EUT into the Measurement Adapter socket.
- Plug the Power Meter into a wall socket, connected directly to the mains.
- Plug the Measurement Adapter into the Regulated Power Supply.
- Use the USB cable to connect the output USB channel of the Power Meter to a USB socket of the Controller Computer
- Turn-on the Power Meter.
- Navigate to the settings menu on the Power Meter and set the measurement ranges to cover the whole range of power draw expected during the benchmark. (Note: For a normal desktop computer, the 300 Volt (V) and 0.5 Ampere (A) range should be fine, a high end desktop computer might need the 1 A or even 2 A settings). Also adjust the crest factor accordingly.
- No further manual configuration of the Power Meter is required as other settings will be set automatically by the test-suite software.
- At this point we are ready to start both computers (Controller Computer and EUT).

## 6 Creating USB Key

The whole Test Suite is delivered as a zipped image file (.img.zip). This file has to be unzipped and then flashed on a USB key with a capacity of at least 16 GB. This can be done on any computer and on any major operating system you have available:

- Microsoft Windows - see § 6.1
- Apple MacOSX - see § 6.2
- Linux - see § 6.3

It is recommended to use a reasonably fast USB 3.0 key (or newer), to speed up flashing the image and installation on the EUT.

**WARNING:** Following the flashing procedure will erase all data on the target device. Make sure to select your USB key and not a hard drive containing any important files!

A single 16GB partition is created on the USB key independently of its size. If the capacity of the stick is higher, a tool such as GParted (<https://gparted.org/livecd.php>) can be used to expand the size of the partition.

### 6.1 Microsoft Windows

A third party tool to configure USB keys is needed. In this case we explain how to use a free multi-platform application called Rufus that can be downloaded from <https://rufus.ie>. This test procedure has been verified with Rufus version 3.10.

Perform the following steps (see Figure 6.1):

- Download and launch Rufus. The portable version that does not require installation is recommended.
- Insert the USB key in one of the Personal Computer (PC)'s USB connectors
- Select the USB device from the Device field's drop-down menu
- Select Disk or ISO Image from the Boot selection menu
- Click the SELECT button to select the delivered and unzipped .img file in the window that pops-up and then click open.
- Write the image to the USB stick by clicking on START
- If when clicking START you get the mode selection dialog, select DD Image mode.
- It could happen that Rufus requires some additional files, if this is the case, a new dialog box will appear, select Yes to continue.
- If additional warning messages appear, keep the default values and select Yes to continue
- The image is now being written and a progress bar is displayed at the bottom of the main Rufus window, under Status.

- When the image writing process is finished, the word **READY** appears in the centre of the progress bar (now shaded green). Select **CLOSE**. Your USB stick now contains the Test Suite.

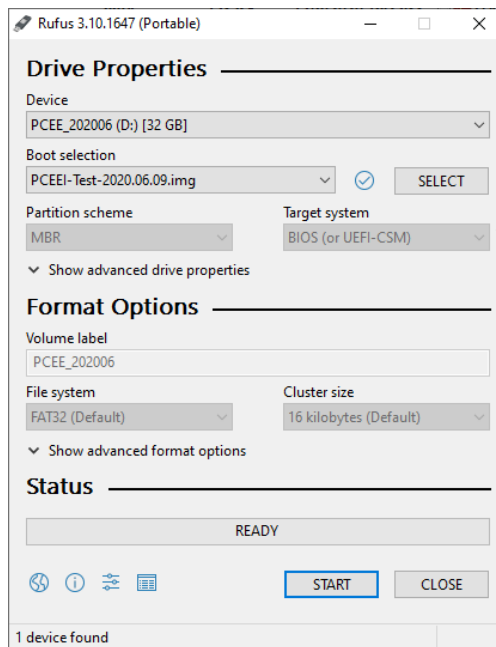


Figure 6.1: USB key configuration with Rufus

## 6.2 Apple MacOSX

Perform the following steps:

- Insert the USB key in one of the Computer's USB sockets
- Open a terminal
- Find the device identifier of the USB key by typing:
  - `diskutil list`
  - The device identifier of the USB key will be something like e.g. `disk2`
  - Check that you have selected the correct device by comparing the size and name in the `diskutil` output to the expected values depending on your USB key.
- Unmount this device by typing (replace `<device-id>` with the device identifier of the USB key from the previous step):
  - `diskutil unmount <device-id>`
- Type the following command to actually write the image (replace `<device-id>` with the device identifier of the USB key from the previous step):
  - `sudo dd if="name of the .img file" of=/dev/<device-id>`

## 6.3 Linux

For Linux it is assumed that you have `sudo` set up to elevate privileges. If this is not the case, use `su` to open a root shell first, then enter the commands without prefixing `sudo`.

Perform the following steps:

- Insert the USB key in one of the PC's USB connectors
- Open a Linux terminal
- Find the device identifier of the USB key by typing:
  - `sudo fdisk -l`
  - The device identifier of the USB device will be something like e.g. `/dev/sdc`.
  - Check that you have selected the correct device by comparing the size and name in the `fdisk` output to the expected values depending on your USB key.
- Unmount this device by typing (replace `<device-id>` with the device identifier from the previous step and with the partition number) for each auto-mounted partition on the USB key.
  - `sudo umount <device-id><X>`
  - Example: `sudo umount /dev/sdc1`
- Type the following command to write the image (replace `<device-id>` with the value from the previous step). Make sure to select the device itself (e.g. `/dev/sdc`) and not the first partition (`/dev/sdc1`) in this step.
  - `sudo dd if="name of the image file" of=<device-id>`
  - Example: `sudo dd if="PCEEI-Test-2021.01.19.img" of=/dev/sdc`

## 6.4 Updating an existing USB key

Once a USB key has been created it can be easily updated once a new version of the test suite software is released, without having to download the new full image and going through the flashing procedure.

To update the USB key connect it to an internet connected computer, then run `update.bat` for a Windows Computer or `update.command` for MacOSX and Linux computers. The update mechanism will only download changed files, so an update usually only takes a few minutes.

**IMPORTANT:** The update procedure will not remove your `test-results` folder from the USB stick, but will remove all other files and folders which are not part of the test suite software in its initial state.

**IMPORTANT:** To use the newly updated software you have to install it as described in § 7.

## 7 Software Installation

In this section we detail how to install the PCEET controller software on the Controller Computer and the Test Suite on the EUT. All the required software is in the USB stick you have flashed on the previous step.

### 7.1 General Remarks

Before installing the Controller Computer software and the EUT Test Suite you have to take into account the following remarks:

- An internet connection is required. Most files will be served from the USB media, though, so data transfer volume is limited.
- The installation procedure has to be run only once for each particular Controller Computer and EUT. If the installation produces an error, it can be safely repeated; only the missing packages and applications will be installed.
- The user account on the Controller Computer and the EUT which is used to carry out the test procedure should have administrative permissions. On Windows and MacOSX this means to use an administrator account, on Linux this means to use an account which can use `pkexec` to elevate privileges.

### 7.2 Controller Computer

The PCEET controller software is multi-platform; it runs on:

- Microsoft Windows 10 or Windows 11 x86 64-bit.
- MacOSX 12 Intel and ARM
- Ubuntu 22.04 LTS 64-bit Linux

This software allows the user to:

- Synchronize the Controller Computer and the EUT clock
- Check the connection between the Power Meter and the Controller Computer
- Start/Stop the acquisition of power measurements
- Examine the results via a web browser application

Figure 7.1 below serves as a reference as this GUI is the same for the 3 versions for each supported OS.

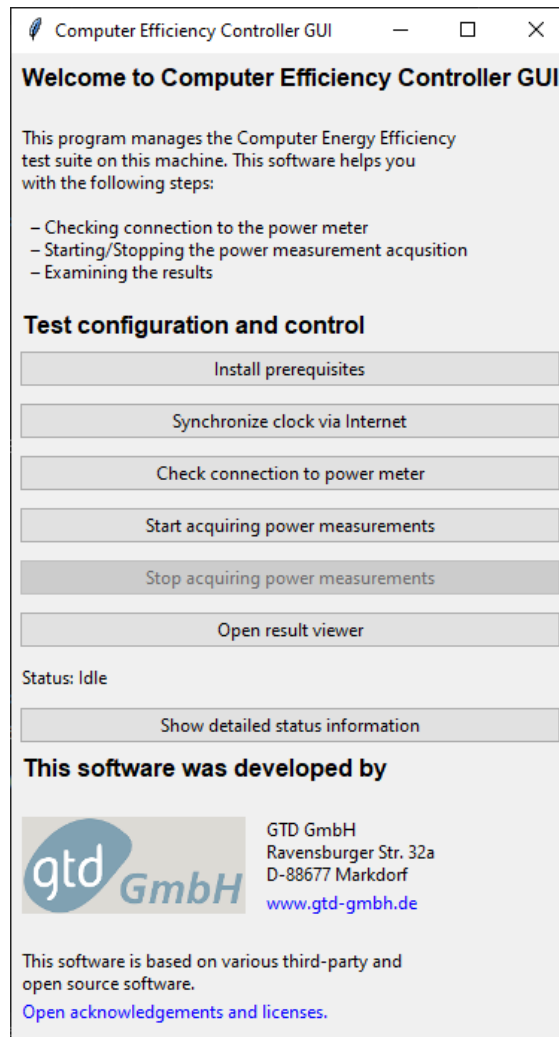


Figure 7.1: Controller Computer GUI

The PCEET Controller software can be installed on any operating system you have available:

- Microsoft Windows - see § 7.2.1
- Apple MacOSX - see § 7.2.2
- Linux - see § 7.2.3

NOTE: The installation procedures below only install the required software dependencies and drivers. The Controller Computer application has to be executed from the USB key, a procedure explained in a subsequent section.

## 7.2.1 Microsoft Windows

- Insert the USB key in one of the PC's USB connectors
- Open a file explorer and navigate to the root directory of the USB key
- Double click on the `windows_controller.exe` file.



- Click “yes” when asked if you want to allow this app to make changes on your computer. The Controller Computer GUI is now displayed (see Figure 7.1)
- Click “Install prerequisites”. You can check the current status of the installation on the “Status” field on the bottom left side of the GUI. If you want more detailed information, click on “Show detailed status information”. A text box containing more log messages is then presented.
- When a driver-installation utility called Zadig pops up do the following, if you have a USB connected power meter (otherwise just close the Zadig utility):
  - Select the meter on the top deployable bar. Usually it is listed as “WT series” and already selected. If it doesn’t show up, click “List all devices” from the “Options” menu and select it.
  - Select the “WinUSB” driver in the text field on the right side of the green arrow. Usually the correct driver is already selected
  - Click “Install driver”
  - Once the installation is complete, close the Zadig utility
- During the rest of the installation no more user interaction is required
- When the installation is completed a success window pops up and the Status is reported as “Idle”

### 7.2.2 Apple MacOSX

- Insert the USB key in one of the Computer’s USB connectors
- Open a Finder window and navigate to the root directory of the USB key
- Double click the `macosx_controller` file. The Controller Computer GUI pops up (see Figure 7.1)
- Click “Install prerequisites”. You can check the current status of the installation on the “Status” field on the bottom left side of the GUI. If you want more detailed information, click “Show detailed status information”. A text box containing more log messages is then presented
- When the installation is completed a success window pops up and the Status is reported as “Idle”

### 7.2.3 Linux

The installation procedure is the same independently of the Linux distribution as the software will be installed with the distribution’s native package manager:

- Insert the USB key in one of the PC’s USB connectors
- Open a file explorer
- Double click the `linux_controller` file. The Controller Computer GUI pops-up (see Figure 7.1). If you get an error message, the reason might be that the USB stick was automatically mounted without execute permissions. In this case do the following steps:
  - Get the device id of the USB key (e.g. `sdb1`) in a Terminal with the `lsblk` command

- Unmount the USB key with the following command:  
`sudo umount /dev/<device-id>`
  - Mount the USB key again with default mounting options:  
`sudo mount -o uid=$(id -u) /dev/<device-id> /mnt`
  - Navigate to the /mnt folder in your file browser and double click the linux\_controller file
- Click “Install prerequisites”. You can check the current status of the installation on the “Status” field on the bottom left side of the GUI. If you want more detailed information, click “Show detailed status information”. A text box containing more log messages is then presented
  - When the installation is completed a success window pops up and the Status is reported as “Idle”

## 7.2.4 Check the Connection to the Power Meter

Once the Controller Computer software has been installed, we want to make sure that the Power Meter can be accessed by the test framework. To do so, select your power meter and, in case of serial or GPIB connection, the port to use via the dropdown menus on the Controller Computer GUI, then click the button on labeled “Check connection to power meter”. A message should be shown stating that the meter was found and detailing the model, serial number and channels of your device, as you can see in Figure 7.2.

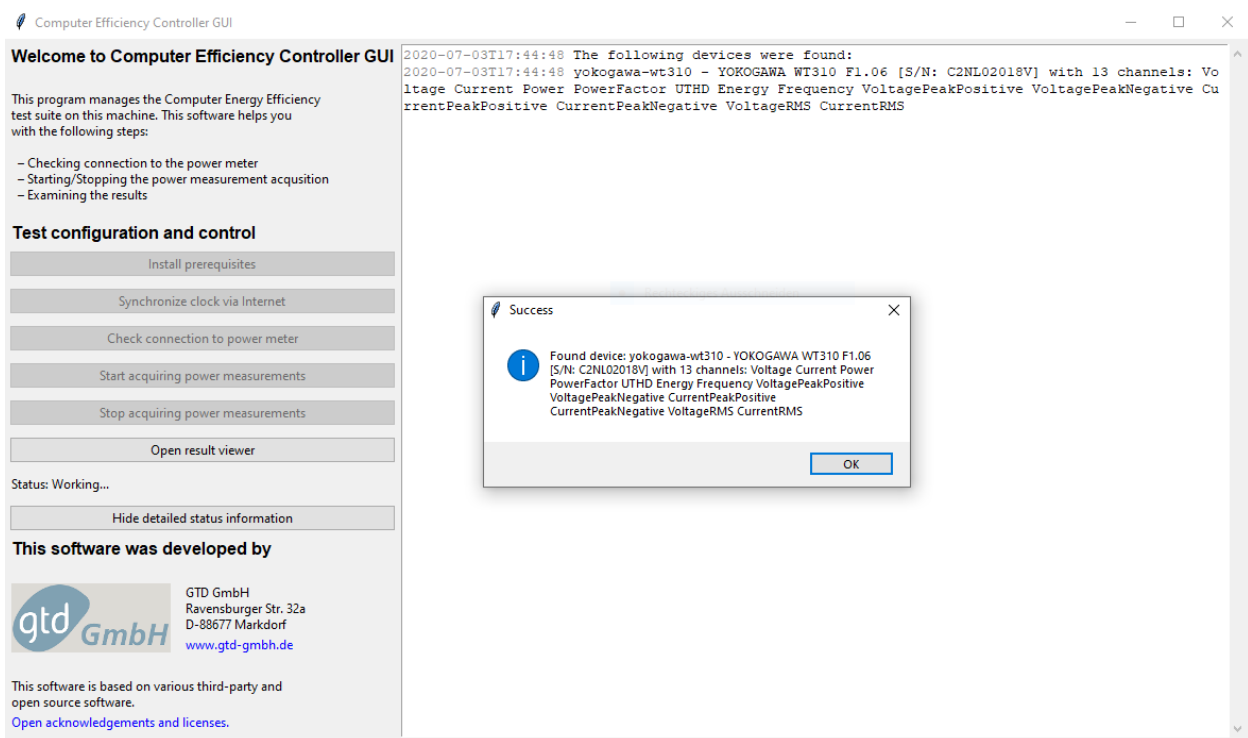


Figure 7.2: Power Meter found

If the meter is not detected correctly, make sure that it is properly connected via USB to the EUT, is switched on and that drivers have been set correctly.

### 7.3 EUT Software Preparation

The Computer Efficiency Test Suite is multi-platform; it runs on:

- Microsoft Windows 10 or Windows 11 x86 64-bit.
- MacOSX 12 Intel and ARM
- Ubuntu 22.04 x86 64-bit Linux

Figure 7.3 below serves as a reference as this GUI is the same for the 3 proposed OS versions of the software.

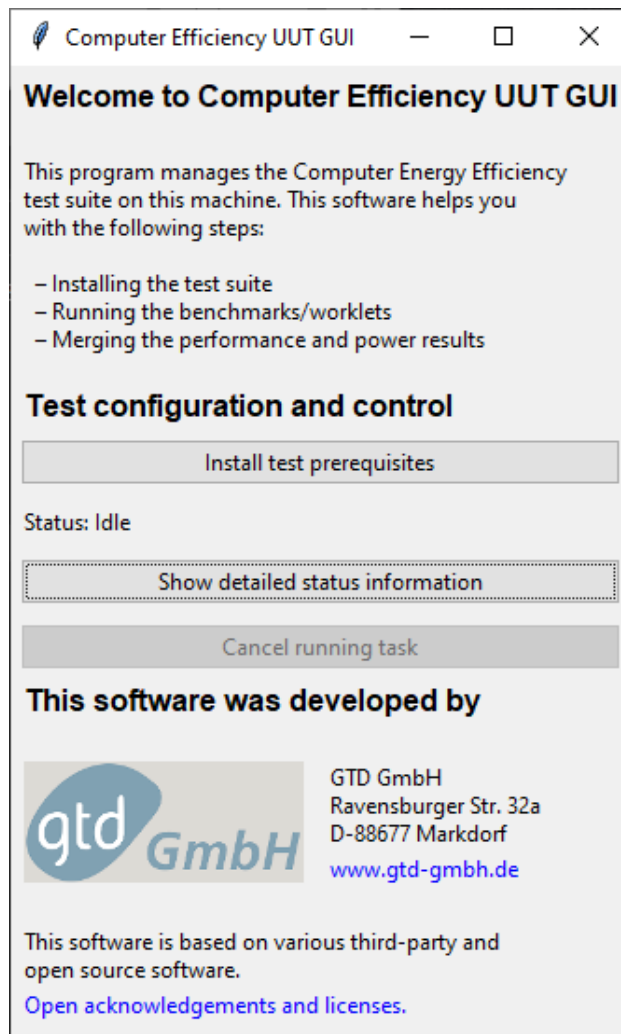


Figure 7.3: Test Suite installation GUI

The Computer Efficiency Test Suite can be used to test a new computer running one of the following operating systems:

- Microsoft Windows - see § 7.3.1
- Apple MacOSX - see § 7.3.2
- Linux - see § 7.3.3

**NOTE:** Please be aware that the installation can take a long time as some software packages might be downloaded from the Internet. In case a download fails, this will issue an error on the console identifying the software packages that could not be downloaded in a reasonable time. The installer was designed to take this into account, and may be run as many times as is necessary with only the missing packages being installed on subsequent runs.

### 7.3.1 Microsoft Windows

- Insert the USB key in one of the EUT’s USB connectors
- Open a file explorer and navigate to the root directory of the USB key
- Double click the windows\_eut\_install.exe file.
- Click “yes” when asked if you want to allow this app to make changes on your computer. The EUT installation GUI is now displayed (see Figure 7.3)
- Click on “Install test prerequisites”. You can check the current status of the installation on the “Status” field on the bottom left side of the GUI. If you want more detailed information, click “Show detailed status information”. A text box containing more log messages is then presented (see Figure 7.4)
- When the installation is completed a success window pops up and the Status is reported as “Idle”
- Close the GUI and remove the USB key

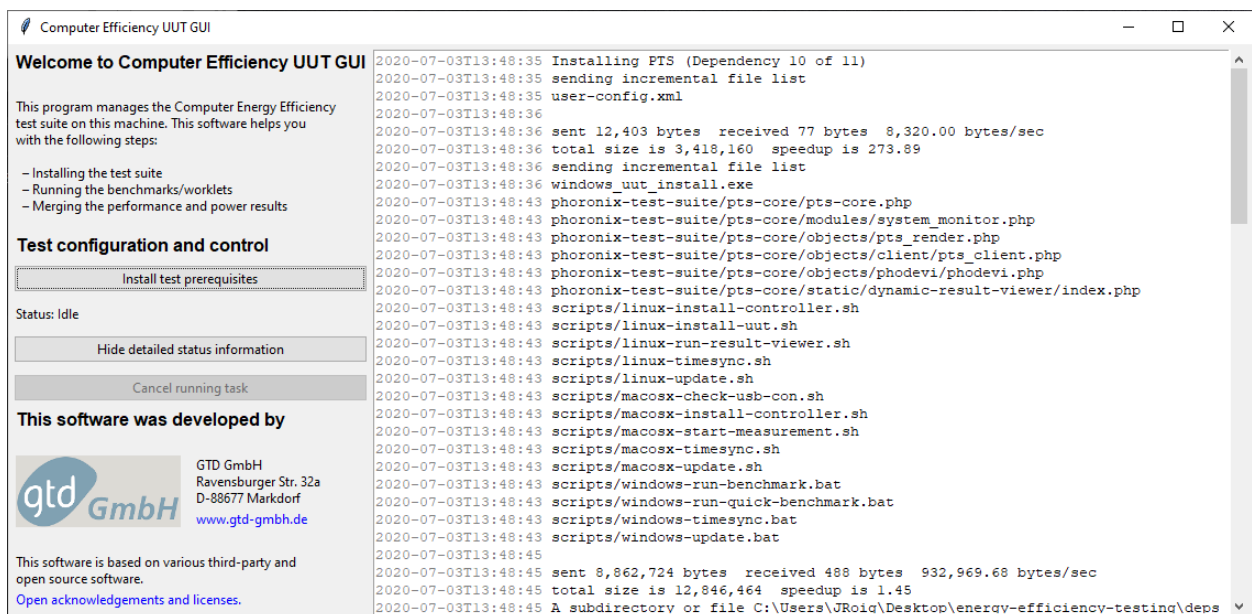


Figure 7.4: Test Suite installation GUI with terminal deployed

### 7.3.2 Apple MacOSX

- Insert the USB key in one of the EUT's USB connectors
- Open a Finder window and navigate to the root directory of the USB key
- Double click the `macosx_eut_install` file. The EUT installation GUI pops-up (see Figure 7.3)
- Click "Install test prerequisites". You can check the current status of the installation on the "Status" field on the bottom left side of the GUI. If you want more detailed information, click "Show detailed status information". A text box containing more log messages is then presented (see Figure 7.4)
- When the installation is completed a success window pops up and the Status is reported as "Idle"
- Close the GUI and remove the USB key

### 7.3.3 Linux

The installation procedure is the same independently of the Linux distribution as the software will be installed using the distribution's native package manager as necessary:

- Insert the USB key in one of the PC's USB connectors
- Open a file explorer
- Double click the `linux_eut_install` file. The Test-Suite GUI pops-up (see Figure 7.3). If you get an error message, the reason might be that the USB stick was automatically mounted without execute permissions. In this case do the following steps:
  - Get the device id of the USB key (e.g. `sdb1`) in a Terminal with the `lsblk` command
  - Unmount the USB key with the following command:  
`sudo umount /dev/<device-id>`
  - Mount the USB key again with default mounting options:  
`sudo mount -o uid=$(id -u) /dev/<device-id> /mnt`
  - Navigate to the `/mnt` folder in your file browser and double click the `linux_eut_install` file
- Click on "Install test prerequisites". You can check the current status of the installation on the "Status" field on the bottom left side of the GUI. If you want more detailed information, click "Show detailed status information". A text box containing more log messages is then presented (see Figure 7.4)
- When the installation is completed a success window pops up and the Status is reported as "Idle"
- Close the GUI and remove the USB key

## 8 Test Suite Execution

As seen in Figure 5.1 you need two computers to run the Test Suite:

- The Equipment Under Test (EUT)
- A Controller Computer

Each of these computers runs its own application: the Controller Computer application on one side and the EUT Test Suite on the other. In this chapter we explain how to prepare the computers for testing and how to run the required applications.

**IMPORTANT:** The Controller Computer application **must** be started **before** running the EUT Test Suite.

### 8.1 Preparing the EUT

Before running the Controller Computer Software and EUT Test Suite benchmarks, please take into account the following remarks:

- Screen saver, screen blanking and automatic standby mode must be switched off.
- The brightness of the Notebook and Integrated Desktop Display should be set to 90cd/m<sup>2</sup> and 150cd/m<sup>2</sup> respectively according to EN 62623.
- A desktop wallpaper according to EN 62623 shall be set on the EUT.
- All currently available OS updates must be installed and after that automatic OS updates must be disabled
- Disable any firewall software installed on the EUT (e.g. on Windows disable the Windows Defender Firewall).
- Driver settings and power profiles shall remain unchanged, i.e. use the same settings as when the system is shipped to the end-user.
- If the EUT has a rechargeable battery, i.e. it is a laptop, remove the battery for the test. If the battery cannot be removed, make sure that the battery is charged to 100% before starting a test run.
- The EUT shall not be connected to the internet when running the benchmarks, but should be connected to a wireless access point or network switch supporting the highest speed available by the EUT. In order to restrict influence of external factors, the EUT shall be confined within a local test network without external network connections.
- The USB key used during the Test Suite software installation shall **not** remain connected to the EUT during the test run.
- The Controller Computer and EUT, need to be connected to the same local network to synchronize the clocks and to transfer power and result data between the two computers.

## 8.2 Preparing the Controller Computer

Before running the EUT Test Suite, we first have to synchronize both computers' clocks:

- Follow the steps explained in the corresponding installation section to open the Controller Computer Software GUI from the USB stick (i.e., Microsoft Windows – see § 7.2.1); Apple MacOSX – see § 7.2.2; Linux – see § 7.2.3)
- Click “Start time server” on the Controller GUI and note the IP address of the controller computer.
- Note: once the time synchronization is complete, no more user interaction is required on the Controller Computer until the end of the EUT Test Suite execution

## 8.3 Running the EUT

The complete Test Suite is installed in a folder called “energy-efficiency-testing” located on the Desktop of the EUT. The Test Suite must be run from this folder.

Open a file viewing tool, navigate to the “energy-efficiency-testing” folder and double-click the executable file that corresponds to the OS of the EUT:

- Microsoft Windows: windows\_eut\_run.exe
- MacOSX: macosx\_eut\_run
- Linux: linux\_eut\_run

**Note:** the current duration of the Test Suite execution is about 1 hour depending on the performance of the tested computer.

The benchmark tests are started from within the EUT Testing Suite GUI (see Figure 8.1) by following these steps:

- Enter the IP address of the controller computer in the provided text box and click “Synchronize clock”. When the process is finished a success message pops up.
- Enter a test configuration name in the corresponding text field (text box next to “Name of test configuration”). This name must not contain any spaces. This name is used as an identifier for your EUT in the test results, e.g. “EUT-EC-Test-Results-2020-001”.
- Write a test description (optional) in the corresponding text field. This can describe your EUT with a sentence, e.g. “Testing the Energy Efficiency of my computer, model number xxx-xxx” and it can contain spaces
- Configure which worklets shall be executed by clicking “Configure worklets”. The default setting is to run all worklets and determine the number of repetitions automatically.
- Click “Run test” to execute the selected worklets, lasting 1 to 3 hours in case the default selection of worklets is used.
- You can follow the test progress in the text window which is opened by clicking “Show detailed status information”.
- After the complete suite of tests has been run, power data will be transferred from the controller and merged with the performance data. The test result will be transferred back to the controller computer. A message will appear once this is complete.

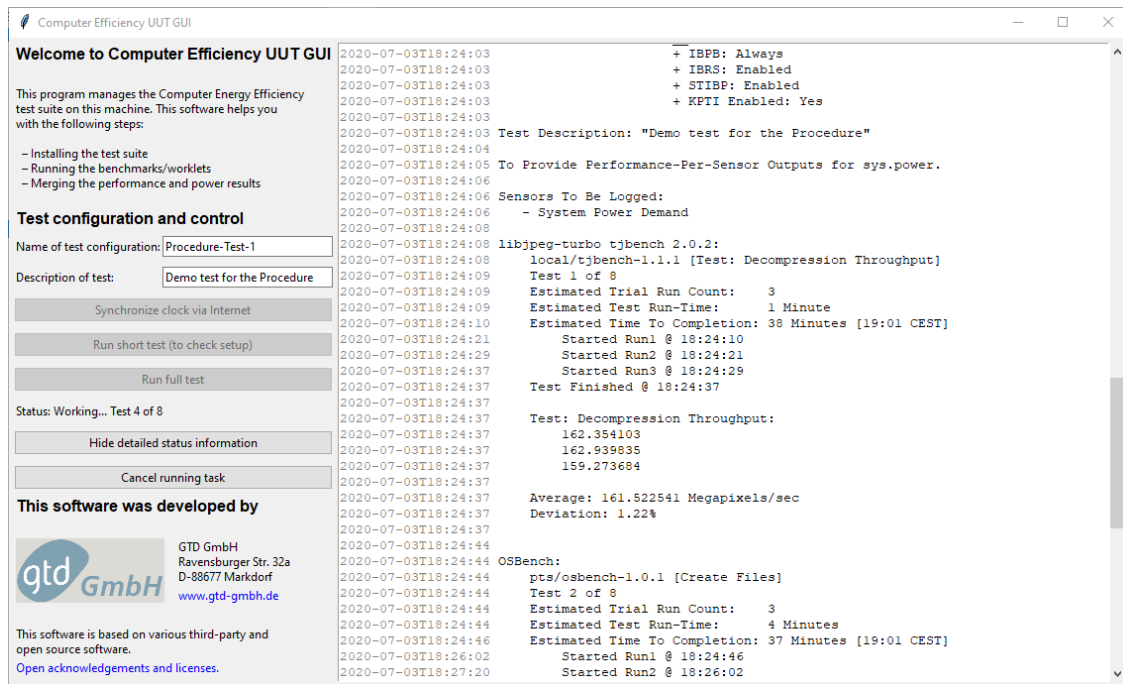


Figure 8.1: Test Suite execution GUI

### 8.3.1 Check the Tests Progress

When you run the tests and you select to display the detailed status information, the presented text box provides information on the progress of the tests together with some additional data (see Figure 8.1).

In particular, the main shown data is:

- Worklet name
- Worklet ID (Name of the test if the Worklet can run in different configurations)
- Test number and total number of tests
- Estimated Trial Run Count: number of times the Worklet will be initially repeated (more if necessary to decrease deviation)
- Estimated test run time: estimated duration of the Worklet
- Estimated time to test-suite completion: estimated duration of the rest of the Worklets
- Timestamp of each Worklet repetition
- Timestamp of Worklet execution end
- Performance result for each of the Worklet repetitions
- Calculated average of performance results
- Deviation in performance result

The same information is saved to a log file, so that you can check the output again after the test has finished and the software was closed. The log files are located on your Desktop in the folder `energy-efficiency-testing/log`.



### 8.3.2 Troubleshooting

- Repeating test runs on same EUT after configuration or hardware changes
  - If hardware configuration is changed (i.e., CPU or GPU) post installation of the PCEET test software, the PCEET test software must be re-installed (“Install Pre-requisites”).
  - When the PCEET test software installs, it checks for CPU & GPU installed in the gls{EUT} to determine which versions of programs to run.
- Chrome browser issue during installation:
  - Uninstall the Chrome browser on the gls{EUT} before installing the PCEET test software.
  - If an incorrect version of the Chrome browser is installed on the gls{EUT} the Selenium Kraken Chrome worklet will not run.
- Unity Spaceship doesn’t produce a result:
  - For some systems with low end graphics, Unity Spaceship rendering at 4k resolution doesn’t produce a result because the Frames/Sec value is too low. After analyzing data, some system with a <1 Frames/sec don’t produce a result.
- Wildlife Extreme doesn’t produce a result:
  - For some systems, the installed graphics drivers do not support the Vulkan graphics API, which causes Wildlife Extreme to no being able to run. Contact the manufacturer of the graphics card present in the system to check for appropriate drivers.

## 9 Complete List of Worklets

In the table below, we include a complete list of the Worklets which are included with this version of the software tool. The table lists them grouped by the performance or hardware configuration they are intended to test.

Table 9.1: Test Suite software Worklets and description

#	Worklet Name	Category	Description / Test Purpose
1	LibreOffice Writer	System <sup>1</sup>	Create a text document with images and tables
2	LibreOffice Impress	System	Create a presentation with several slides
3	LibreOffice Calc	System	Create a spreadsheet with formulas and diagrams
4	SQLite 1 thread	Storage	Create and query a database
5	Encode MP3	CPU	Encode WAV Audio to MP3
6	Selenium Kraken Chrome	System	Run the Kraken Javascript Benchmark in Chrome
7	PyBench	System	Test performance of python programs
8	TJBench	System	JPG image decompression
9	LibreOffice Convert	System	Convert Documents to PDF
10	FreeCAD	System	Create several 3D models in FreeCAD
11	Decompress ZSTD	Storage	Decompress a file with the ZSTD algorithm
12	Compress ZSTD	CPU	Compress a file with the ZSTD algorithm
13	Compress 7zip	CPU	Compress a file with the 7zip algorithm
14	Stockfish	CPU	Chess AI simulation
15	Blender	GPU	Render a scene in Blender using available GPUs
16	3DMark Wild Life Extreme	GPU	Graphics Benchmark provided by UL
17	Unity Spaceship	GPU	Graphics Benchmark provided by Unity
18	Handbrake	GPU	Hardware-accelerated video transcoding

<sup>1</sup>Worklets of the System category test multiple subsystems (such as CPU, GPU, Storage) at the same time producing a performance score for the full computer system.

## 10 Results Analysis

The final combined test results are stored in the “test-results” folder that was created on the USB key plugged into the controller computer.

Open the Controller Computer GUI, click on “Open result viewer” and a web browser with a list of all the executed tests will be opened.

By default test results are sorted by date, the newest listed first. The name of the test result includes the test configuration name given through the EUT GUI and the date and time of when the test was started (see Figure 10.1).

The screenshot shows the 'ptest Result Viewer' interface. At the top, there is a search bar labeled 'Search Test Results', a dropdown menu set to 'Sort By Date', and a green 'Update' button. Below this, a summary line reads '60 RESULT FILES CONTAINING A COMBINED 4824 TEST RESULTS'. The main content area lists two test entries:

- pceet-my-computer-test-2-2020-07-03-10-52**  
 145 TESTS 1 SYSTEMS FRIDAY 3 JULY 15:58  
 Testing our new GenericBrand GenericModel computer, second run
- pceet-my-computer-test-1-2020-07-02-13-04**  
 145 TESTS 1 SYSTEMS FRIDAY 3 JULY 09:59  
 Testing our new GenericBrand GenericModel computer, first run

Figure 10.1: List of example test results

To open a single result, click the test result entry to display all available metrics produced during the corresponding test run.

To compare several test results, click the checkbox below each result entry’s title, then click the appearing button to compare the results.

The opened result page consists of several sections, which will be briefly explained in the following sections.

## 10.1 Technical Details about the EUT(s)

Technical details, such as CPU model, motherboard model, amount of memory, graphics card model, operating system versions and much more is listed in a table. This helps to clearly identify the EUT and gives an overview of the tested hardware.

## 10.2 Performance and Efficiency Comparison

This section is only shown when several systems are selected for comparison. When comparing two EUTs, a bar graph will be shown, when comparing three or more systems a circular graph (also called a “spider graph”) will be shown. An example of such a graph is shown below (see Figure 10.2).

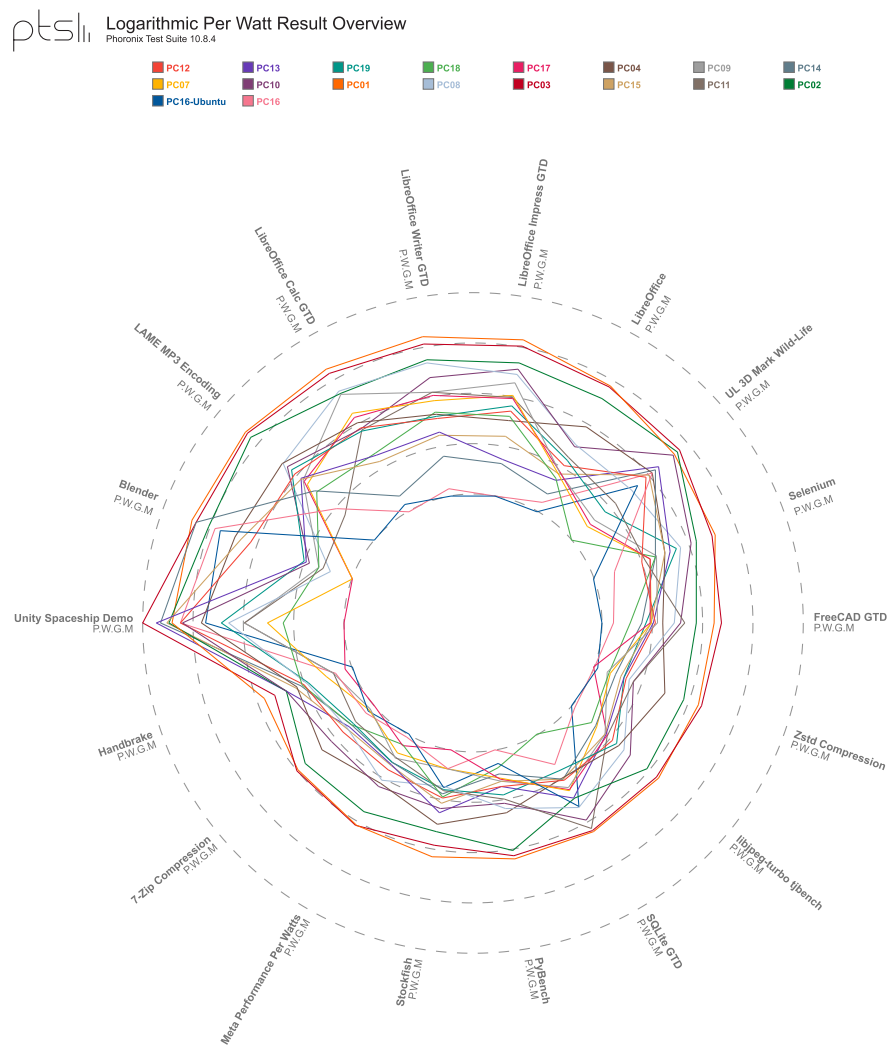


Figure 10.2: Circular Efficiency Graph

In this graph, better (i.e. more efficient) results are located further to the outside of the concentric circles. The scale of the graph is logarithmic.

### 10.3 Table of Performance Results

For each Worklet the performance results are listed in a table. This table does not provide a measure of power consumption, but gives an assessment of the performance of the computer conducting these tests.

### 10.4 Detailed Test Results for each Worklet

For each Worklet at least three graphs are shown, detailing the performance (Figure 10.3), efficiency (Figure 10.5) and power demand (Figure 10.4) measurements. Additional graphs may be generated depending on what the benchmark provides in terms of additional results.

Choose between the three graphs by clicking the orange buttons above the performance graph which is shown by default. See screenshots below for an example for each of these graphs when four test runs are selected for comparison.

Each graph defines the unit of the measured values, such as MB/s and whether higher values are better or lower values are better. This depends on the type of benchmark and is important to consider.

#### UL 3D Mark Wild-Life 1.0

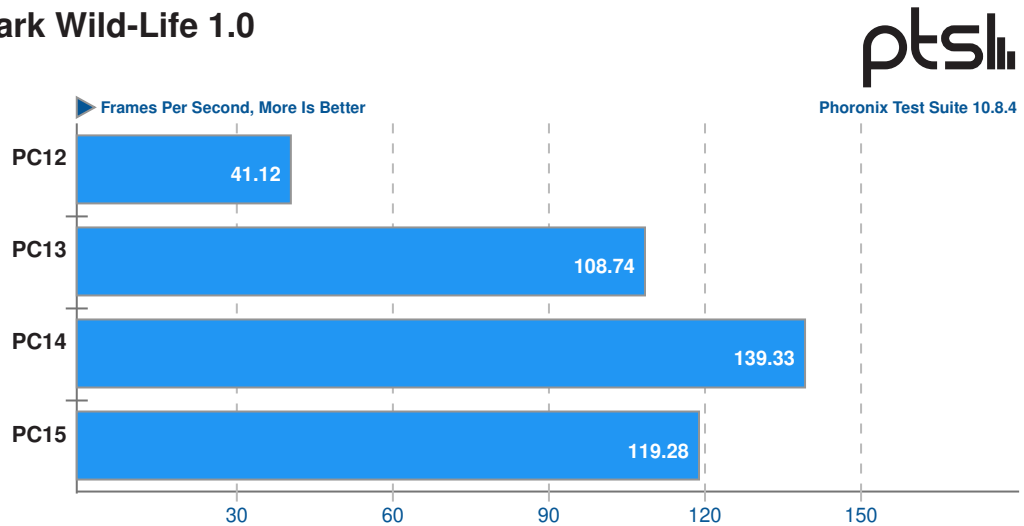


Figure 10.3: Example Worklet performance

### UL 3D Mark Wild-Life 1.0

#### System Power Demand

	Min	Avg	Max
PC12 (try 1)	10.61	65.4	72.6
PC13 (try 1)	10.04	127.3	155.81
PC14 (try 1)	18.42	178.4	206.17
PC15 (try 1)	31.35	174.4	250.95

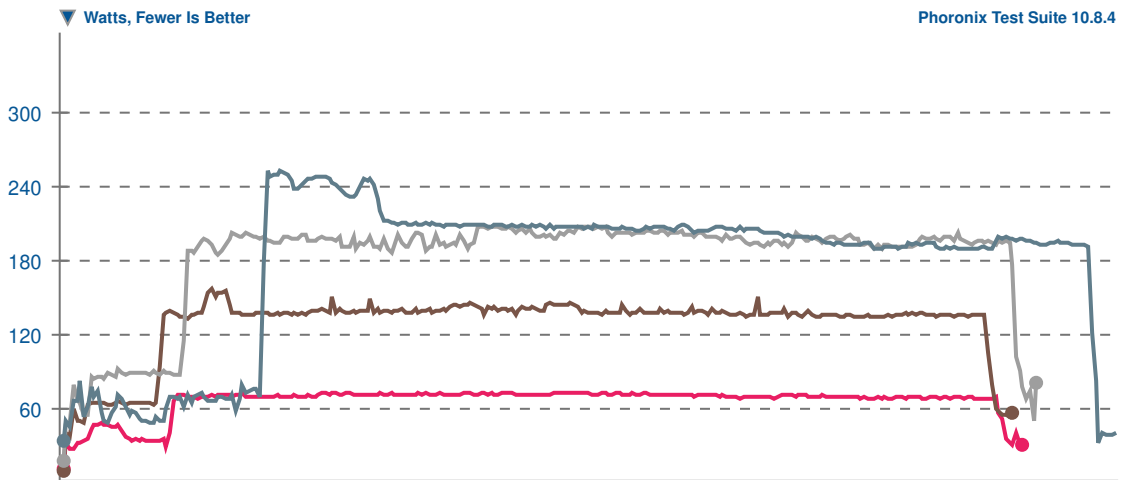


Figure 10.4: Example Worklet power

### UL 3D Mark Wild-Life 1.0

#### System Power per Performance (Efficiency)

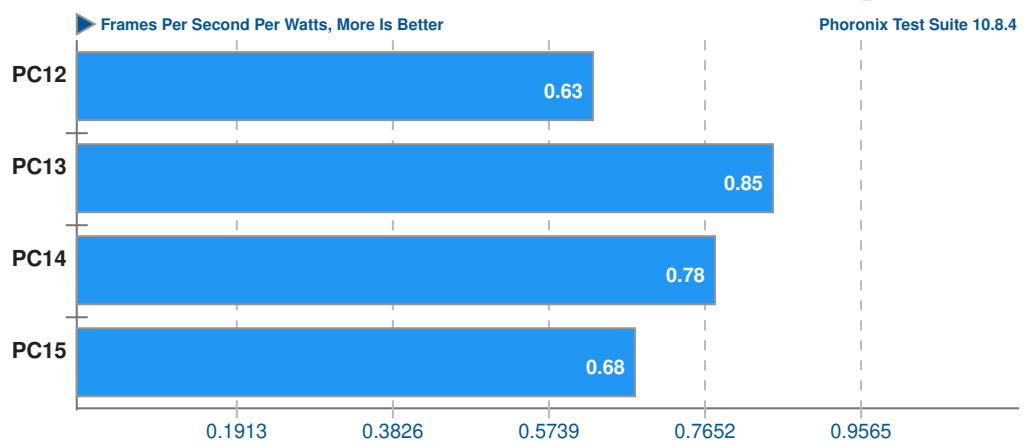


Figure 10.5: Example Worklet efficiency

For the power measurements each automatic repetition of the Worklet during a test run is shown as a separate line in the result graph. Comparing several systems with many repetitions can lead a very complex graph.

**Note:** There are worklets that produce more than one score depending on system configuration or number of hardware accelerators it supports on the platform. For example, Handbrake could have multiple scores and the best score based on Performance Per Watt is picked for final score calculation.

## 10.5 Meta Values and System Power Monitor

All the way to the end of the report, two graphs are showing the meta efficiency calculated over all Worklets and the full series of power measurements logged during the test run.

The meta efficiency is created by calculating the geometric mean of the individual efficiency results of each Worklet and represents the efficiency result of the EUT fused into a single number. These numbers are only comparable between test runs if exactly the same selection of Worklets has been run, so be sure to only compare meta values of test runs with the same worklets having successfully been run. See Figure 10.6 below.

### Meta Performance Per Watts

Performance Per Watts

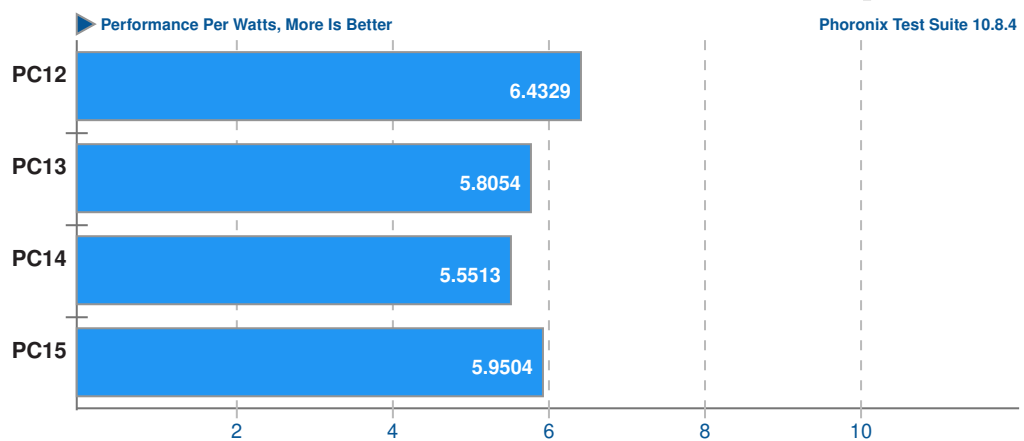


Figure 10.6: Example Meta Efficiency

The system power monitor includes the full series of power measurements acquired during the test runs. These line graphs are not directly comparable to each other in terms of average or length as Worklets are dynamically repeated during the test run as needed to improve statistical accuracy.

## 10.6 Exporting Results

At the top of the result page, all benchmark results can be exported to various formats, e.g. CSV. This can then be loaded into a spreadsheet to continue analysis and calculate custom averages or geometric means for example based on a specific subset of the full suite of Worklets.

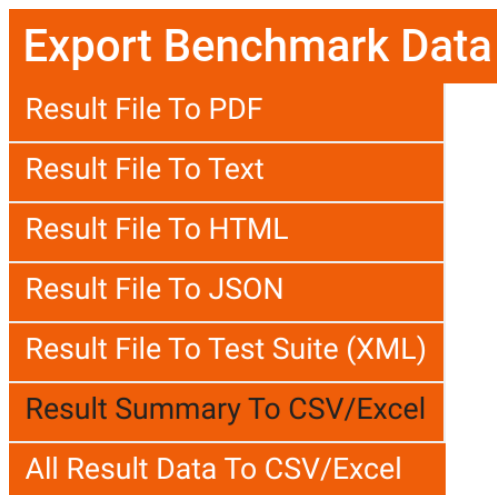


Figure 10.7: Export Data Menu

The following steps show how to export and retrieve the raw data:

- Click “Result Summary to CSV/Excel” in the “Export Benchmark Data” menu, see Figure 10.7.
  - The “Result Summary to CSV/Excel” is a file that contains all overall scores as well as 5 data points for each worklet.
- Click “All Results Data to CSV/Excel” in the “Export Benchmark Data” menu.
  - The “All Results Data to CSV/Excel” has much more detailed view of the data. This is only needed for detail review of the data which includes performance and power data for each “try” of each worklet.
- The “Result Summary to CSV/Excel” and “All Results Data to CSV/Excel” files are downloaded in the Downloads folder of the PCEET Controller Computer.
- Open the “Result Summary to CSV/Excel” file in a spreadsheet program and the data looks like as shown in Figure 10.8. To copy the data into a summary sheet, select all the data in column B and copy the data to the clipboard.



	A	B
1	Meta Performance Performance (PC01) [Performance]	210.3078
2	Meta Power Power in Watts (PC01) [Power in Watts]	10.2761
3	Meta Performance Per Watts Performance Per Watts (PC01) [Performance Per Watts]	20.4629
4	Meta Performance Per Watthours Performance Per Watthours (PC01) [Performance Per Watthours]	1230.6401685772
5	System Power Demand Monitor (PC01) [Watts]	13.500199965622
6	System Energy Demand Monitor (PC01) [Wattseconds]	58904.7475
7	LibreOffice Writer GTD (PC01) [Seconds]	166.252
8	LibreOffice Writer GTD System Power per Performance (Efficiency) (PC01) [Performance Per Watts]	14.3
9	LibreOffice Writer GTD System Energy per Performance (PC01) [Performance Per Watthours]	311.52798789713
10	LibreOffice Writer GTD System Power Demand (PC01) [Watts]	4.2074432677761
11	LibreOffice Writer GTD System Energy Demand (PC01) [Wattseconds]	695.28
12	LibreOffice Impress GTD (PC01) [Seconds]	93.164
13	LibreOffice Impress GTD System Power per Performance (Efficiency) (PC01) [Performance Per Watts]	26.25
14	LibreOffice Impress GTD System Energy per Performance (PC01) [Performance Per Watthours]	1018.8679245283
15	LibreOffice Impress GTD System Power Demand (PC01) [Watts]	4.0893530997305
16	LibreOffice Impress GTD System Energy Demand (PC01) [Wattseconds]	379.2875
17	LibreOffice Calc GTD (PC01) [Seconds]	31.915
18	LibreOffice Calc GTD System Power per Performance (Efficiency) (PC01) [Performance Per Watts]	50.97
19	LibreOffice Calc GTD System Energy per Performance (PC01) [Performance Per Watthours]	5779.2755905512
20	LibreOffice Calc GTD System Power Demand (PC01) [Watts]	6.147874015748
21	LibreOffice Calc GTD System Energy Demand (PC01) [Wattseconds]	195.195
22	SQLite GTD Threads / Copies: 1 (PC01) [Seconds]	36.263
23	SQLite GTD System Power per Performance (Efficiency) (PC01) [Performance Per Watts]	58.85
24	SQLite GTD System Energy per Performance (PC01) [Performance Per Watthours]	5905.5052264808

Figure 10.8: Imported Data into Spreadsheet

## 11 Efficiency Metric Calculation and Example Data

This chapter shows how the performance and power results are used to calculate the various result metrics produced by the test software. First we show how to calculate normalized performance and efficiency for a single worklet, and then how to combine all single worklet results to get meta performance and meta efficiency.

This document only contains the information necessary to understand how the values are calculated, but doesn't go into detail why the metrics are defined as they are. To get more information about this see the White Paper on the Definition of Efficiency Metrics for Computers, [RD03].

### 11.1 Calculation of Performance, Power and Efficiency Scores of a Worklet

Each worklet may be repeated multiple times during a test run, to make sure the results are stable and reproducible, and to also average out measurement errors or disturbances introduced by e.g. operating system background activity. Refer to Table 11.1 for minimum and maximum numbers how often a worklet might be retried. Given the minimum and maximum number of retries, the test software automatically decides the exact number of retries needed based on the standard deviation of the tries done so far.

Raw performance results of worklets are divided into two categories: Some worklets have result metrics of the category Lower is better (LIB), e.g. a result in seconds, and others have metrics of category Higher is better (HIB), e.g. a result in frames per second. Refer to Table 11.1 to see which kind of category a worklet belongs to.

Table 11.1: Worklet Measurement Details

#	Worklet Name	Performance Unit	Metric Category	Number of Tries
1	LibreOffice Writer	Seconds	LIB	Min: 1, Max: 1
2	LibreOffice Impress	Seconds	LIB	Min: 1, Max: 1
3	LibreOffice Calc	Seconds	LIB	Min: 1, Max: 1
4	SQLite 1 thread	Seconds	LIB	Min: 3, Max: 10
5	Encode MP3	Seconds	LIB	Min: 3, Max: 5
6	Selenium Kraken Chrome	Milliseconds	LIB	Min: 3, Max: 5
7	PyBench	Milliseconds	LIB	Min: 3, Max: 5
8	TJBench	Mega Pixel/Second	HIB	Min: 3, Max: 5
9	LibreOffice Convert	Seconds	LIB	Min: 3, Max: 10

10	FreeCAD	Seconds	LIB	Min: 3, Max: 10
11	Decompress ZSTD	MB/Second	HIB	Min: 3, Max: 10
12	Compress ZSTD	MB/Second	HIB	Min: 3, Max: 10
13	Compress 7zip	MIPS	HIB	Min: 3, Max: 5
14	Stockfish	Nodes/Second	HIB	Min: 3, Max: 5
15	Blender	Seconds	LIB	Min: 1, Max: 1
16	3DMark Wild Life Extreme	Frames/Second	HIB	Min: 1, Max: 1
17	Unity Spaceship	Frames/Second	HIB	Min: 1, Max: 1
18	Handbrake	Seconds	LIB	Min: 1, Max: 1

After running the worklet multiple times, the average performance and average power results are calculated using eq. 11.1 and eq. 11.2.

$$\overline{Perf(W_i)} = \frac{\sum_{n=1}^L Perf(W_i)_n}{L} \quad (11.1)$$

$$\overline{Power(W_i)} = \frac{\sum_{n=1}^L \frac{\sum_{m=1}^J Power(W_i)_{m,n}}{J}}{L} \quad (11.2)$$

where:

- $\overline{Perf(W_i)}$  = Average Performance of Worklet i
- $\overline{Power(W_i)}$  = Average Power of Worklet i
- $Perf(W_i)$  = Performance result of worklet i in a single try
- $Power(W_i)$  = Power used by the EUT when running worklet i
- $L$  = Number of Tries done for this Worklet
- $J$  = Number of Power Measurements for Try

Creating an efficiency score per worklet requires to calculate a normalized performance, which is always a HIB value. This is only done for LIB performance results, by inverting these results with eq. 11.3, while for HIB values, average and normalized performance are the same.

$$NPerf(W_i) = \frac{10000}{Perf(W_i)} \quad (11.3)$$

where:

- $NPerf(W_i)$  = Normalized Performance of Worklet i
- $\overline{Perf(W_i)}$  = Average Performance of Worklet i
- 10000 = Arbitrarily chosen factor to yield not too small values

The last step for the per Worklet results is to calculate the efficiency, defined as Performance per Power which is calculated according to eq. 11.4.

$$Eff(W_i) = \frac{NPerf(W_i)}{Power(W_i)} \quad (11.4)$$

where:

$Eff(W_i)$  = Energy Efficiency of Worklet i  
 $NPerf(W_i)$  = Normalized Performance of Worklet i  
 $Power(W_i)$  = Average Power of Worklet i

## 11.2 Meta Values Calculation

Meta values are used to compress all individual worklet results into a single value which can be easily compared between different systems or test runs. Meta values are calculated for performance, power and efficiency results.

Some worklets, such as Handbrake, do produce more than one result, e.g. for different available hardware accelerators present in the EUT. For these worklets only the most efficient result is included in the meta value calculation.

Each meta value is calculated as the geometric mean of all individual values, see eq. 11.5, eq. 11.6, eq. 11.7.

$$Meta\ Performance = \sqrt[n]{NPerf(W_1) \cdot NPerf(W_2) \cdot \dots \cdot NPerf(W_n)} \quad (11.5)$$

$$Meta\ Power = \sqrt[n]{Power(W_1) \cdot Power(W_2) \cdot \dots \cdot Power(W_n)} \quad (11.6)$$

$$Meta\ Efficiency = \sqrt[n]{Eff(W_1) \cdot Eff(W_2) \cdot \dots \cdot Eff(W_n)} \quad (11.7)$$

where:

$n$  = Number of Worklets  
 $Power(W_i)$  = Average Power of Worklet i  
 $NPerf(W_i)$  = Normalized Performance of Worklet i  
 $Eff(W_i)$  = Energy Efficiency of Worklet i

## 11.3 Other Metrics

The results exported from the result viewer also contain a few additional metrics which define Efficiency as Performance per Energy. Those are calculated similarly to the metrics explained above, except that Power is replaced by Energy. These metrics are only provided for convenience and are not officially supported by the PCEET software.

## 11.4 Sample Data for a Gaming Computer

Sample scores of a Gaming PC with discrete graphics card is shown in Table 11.2 and Table 11.3 for all 18 worklets. Note that this system has both a discrete GPU and an integrated GPU which

is the reason that there are three handbrake scores (third Handbrake score, reading “None” for the hardware accelerator is software transcoding using CPU).

Table 11.2: Test Results of a Gaming PC

Worklet	Raw Performance	Normalized Performance	Power [W]	Efficiency [Perf/W]	Time [s]	Energy [Wh]	Efficiency [Perf/Wh]
Libreoffice-writer-gtd-2.0.0	170.32	58.71	58.39	1.01	170.00	2.76	21.25
Libreoffice-impress-gtd-2.0.0	105.21	95.05	54.96	1.73	105.00	1.61	59.18
Libreoffice-calc-gtd-2.0.0	35.02	285.53	162.11	1.76	35.00	1.58	181.05
Sqlite-gtd-0.0.1	10.27	974.09	85.36	11.41	10.40	0.24	4001.72
Encode-mp3-2.0.0	28.17	354.97	100.65	3.53	28.33	0.79	450.70
Selenium-local-0.0.8 [Kraken Chrome]	346.80	28.84	85.44	0.34	12.50	8.23	3.50
Pybench-1.1.4	519.00	19.27	109.31	0.18	12.00	15.76	1.22
Tjbench-1.2.0 [Test: Decompression Throughput]	328.76	328.76	99.63	3.30	25.33	9.10	36.13
Libreoffice-3.0.0 [Test: 20 Documents To PDF]	31.42	318.29	117.26	2.71	31.67	1.02	311.03
Freecad-gtd-0.0.3	100.37	99.63	108.71	0.92	101.29	3.03	32.87
Zstd-1.5.1 [Decompression Throughput]	531.65	531.65	141.26	3.76	57.33	20.86	25.48
Zstd-1.5.1 [Compression Throughput]	68.49	68.49	253.86	0.27	30.00	4.83	14.18
Compress-7zip-2.0.0	164195.00	164195.00	249.93	656.98	20.00	11399.02	14.40
Stockfish-1.5.0	53117865.00	53117865.00	250.37	212155.22	74.00	3694239.13	14.38
Blender-3.4.1 [Blend File: Monster Under the Bed]	15.31	653.17	246.64	2.65	20.00	1.05	622.71
3dmark-wildlife-1.1.0	335.81	335.81	418.42	0.80	66.00	39.03	8.60
Unity-spaceship-1.1.1	80.28	80.28	412.80	0.19	183.00	9.21	8.72
Handbrake-1.1.0 [HW accelerator #1]	30.26	330.51	225.59	1.47	30.00	1.90	174.32
Handbrake-1.1.0 [HW accelerator #2]	26.92	371.42	79.86	4.65	27.00	0.60	621.86
handbrake-1.1.0 [None]	124.00	80.64	262.98	0.31	124.00	9.06	8.90

Table 11.3: Meta Test Results of a Gaming PC

Meta Performance	Meta Power	Meta Efficiency [Perf/W]	Meta Efficiency [Perf/Wh]
515.58	139.58	3.11	46.45