Measuring Power Consumption in the Light of EnergyStar

Holger Macht

<holger@homac.de>

SUSE Linux Products GmbH

February 23, 2008
Revision: 0.1.4

Contents

1 Introduction

Measuring system power consumption on a laptop seems, at first, to be quite difficult and expensive. You may wonder what hardware or software is needed, what is relevant to be measured, and under which circumstances. This document tries to answer all of these questions. This document tries to give a summary about how notebook/laptop computers can be tested regarding their power consumption. In the course of that, it is heavily oriented towards the EnergyStar Specification (rev. 4.0). Actually, this document can also be viewed as an abstract of the EnergyStar Specification. However, it's not meant to be an official one, and thus, the information provided is supplied without liability. For reliable, official information, please consult the original specification[1].

Even if you get results which fulfill the criteria mentioned in this document, you are still not allowed to call the test system "EnergyStar Specified". Please redo all the tests along to the official specification. This summary should serve as an indication of whether the tested system would have a chance to meet the criteria.

This document focuses on mobile computers, basically notebooks, where the power consumption influences the daily way to work. Power consumption on workstation systems is not less important, however, just not the explicit focus of this document. For anything mentioned here, a general use case is considered, that is, think of a business workstation where a graphical desktop environment, or some office application, is running.

1.1 Common Definitions

Here are some common definitions to know about test procedure. These terms are used throughout this document.

Off Mode State where the system is completely powered off. The working state is lost. The EnergyStar specification also refers to this state as "Standby".

Sleep Mode Sleep mode is meant to be a situation where the system saves its current state to memory and goes into a low power mode. It's usually quite fast (both sleep and resume usually only last a few seconds). Other names are "Suspend To RAM" or S3 according to the ACPI specification[2].

Idle System An environment where all one-shot tasks have already completed and are not running anymore. No foreground applications are producing main CPU or hard disk activity. This state can usually be ensured with doing a complete boot and to wait about 15 minutes afterwards. In turn of that, the display will be switched off, more on that below.

[1] http://www.energystar.gov/ia/partners/product_specs/program_reqs/Computer_Spec_Final.pdf
[2] http://www.acpi.info

Unit Under Test (UUT) All test systems will be referred to as UUT. This is the system for which the power consumption is monitored and recorded.

Wake On LAN (WOL) Most recent systems are capable of Wake On LAN. This feature wakes a system through network traffic, which is either powered off or in a suspended state. Most computers require that you turn this feature on in the BIOS.

2 EnergyStar Specification

"Energy Star is a United States government program to promote energy efficient consumer products." [3]

It is most likely that many people have already seen the EnergyStar logo (Fig. 3) referring to EnergyStar. It can be found on monitors, notebooks, but although on technical devices apart from the computer industry. Its main target is to reduce power consumption, and thus, to reduce greenhouse gas emissions in general.

It is still a voluntary label manufacturers can refer to. However, nobody really polices compliance. If a company has test results showing that one of their systems follows the EnergyStar limits regarding power consumption and environment friendliness, they submit the results to the EnergyStar consortium, and then can start labeling their systems with the EnergyStar (Fig. 3) logo. Nobody will validate the results.

Figure 1: EnergyStar

However, contrary to expectations, the EnergyStar approach seems to work out, at least to a certain degree.

The EnergyStar specification consists of two major milestones. The first, Tier 1, is July 20, 2007. Tier 2 is targeted for January 2nd, 2009. So anyone who wants to label their hardware/software components as "EnergyStar certified" has to fulfill the Tier 1 criteria. Tier 2 should just be taken into account to get a good start in preparing for the upcoming year, 2009.

The specification[4] provides a whole set of requirements that a technical system needs to fulfill. The hardware prerequisites and software requirements which seem to be most important are summarized here.

[3] http://en.wikipedia.org/wiki/Energy_Star
[4] http://www.energystar.gov/index.cfm?c=revisions.computer_spec

2.1 Requirements by EnergyStar

There are many prerequisites the tests have to meet to conform to EnergyStar. This section tries to identify the must haves and tries to avoid being excessive about the requirements.

2.1.1 Power Meter

To perform any tests, you must have a suitable power meter. No! It is not enough to monitor the actual power consumption by looking at the values your battery reports in the operating system (such as /proc/acpi/*). These values may be inaccurate.

The power meter has to be connected between the main outlet and the UUT's AC power supply.

Also, the measurement frequency has to meet a frequency of at least 1 reading per second.

The following resolutions should be fulfilled:

- <= 10 W : 0.01 W or better
- > 10 W : 0.1 W or better
- > 100 W : 1 W or better

For the evaluation of all test results, rounding off to the second decimal place is needed.

2.1.2 Test Setup

- All tests have to be done with a live network connected.
- External devices, such as external hard disks, mice, or usb dongles must not be attached to the UUT.
- Remove the battery from the UUT. Otherwise, the charging of the battery would distort the real figures. If this is not possible, at least fully load the battery before doing any tests.
- UUT should be capable of Wake On LAN (if not, you can still do some of the tests, but not all).

2.1.3 Software Settings

- Whatever operating system and software are used, make sure that they came shipped with defaults. For instance, this includes the BIOS, which still has to be set to the defaults.
- Turn the wireless network off. Many systems have a hardware button or a switch so you can make sure it's off. Otherwise, disable the device by using software to unload the module. You can't be sure that this will really disable the antenna, but a lot devices do behave this way.

- Sleep mode has to work. Many systems have issues where the suspend works, but the resume does not. Often the display does not come back. To perform the mentioned tests, it should be sufficient if the system goes to sleep properly. Just make sure that it really does. It can be turned off altogether afterwards.
- The system has to be configured so that the display turns into sleep mode after 15 minutes of idle time. As a chain of reasoning, all idle tests are done with the display switched off.
- Automatic system sleep has to be activated after 30 minutes.
- It is beneficial to be able to turn Wake On LAN off and on while in sleep state.

2.2 Hardware Categories

There are different hardware categories a UUT might fit into. Notebooks fall into two of them. Category B laptops are all systems which include a video card with at least 128 MB dedicated/non-shared memory. All other laptops are consolidated into category A.

2.3 Criteria

To meet the requirements of the specification, the test results must not exceed certain limits as listed below.

Test	Requirement
Off Mode: Sleep Mode Idle State: Wake On LAN (WOL) enabled:	<= 1.0 W <= 1.7 W Category A notebooks: <= 14.0 W Category B notebooks: <= 22.0 W + 0.7 W for Sleep + 0.7 W for Standby (Off Mode)

2.4 Tier 2

To meet the tier 2 requirements:

- Tier 1 requirements must be met.
- Systems have to provide full network connectivity while in sleep mode.
- The link speed of the network has to be reduced when there is only low traffic. As an example, link speed has to be reduced to 10 MBit.

3 Test Procedure

Here are concrete descriptions of the software and test setup needed to actually perform the measurements.

3.1 Operating System

The whole procedure described here is based on GNU/Linux. So all scripts and software provided are written for this operating system. People using something else will have to look for equivalents on their base system and will have to adjust some of the test steps to apply to their own software environment.

3.2 Test Setup

Record System There might be power meters out there which have built-in functionality for recording and accumulating measurements, but usually they do not. The measurements cannot be done on the UUT itself, because this may falsify some measurements, especially in regard to the idle test. So, you must use another workstation or notebook, not the UUT, to record and evaluate the values read out. The only thing this system has to be capable of is an interface to connect the power meter to, quite often this is a serial line.

Software Requirements You must be able to accumulate measurements over a period of time. If the power meter has this functionality built in, you can skip this section. Otherwise, you need to run an application on the recording system. Below (Fig. 2) is a brief description of QtDMM[5].

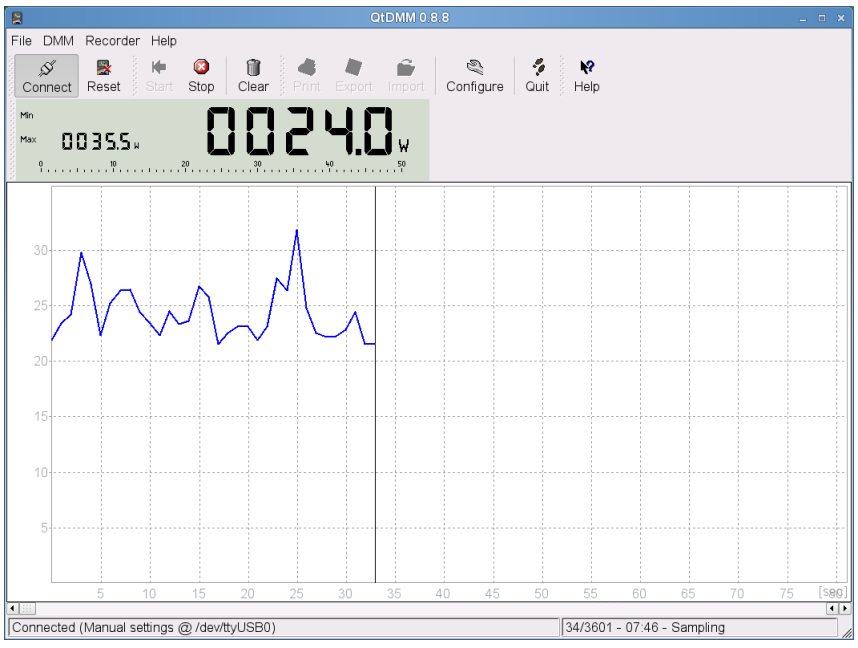

Figure 2: QtDMM in action

1. Download and install it.

[5]http://www.mtoussaint.de/qtdmm.html

2. Connect the serial cable from the power meter to the recording system.

3. Configure it to sample every 1 second (Configure → Sample every:). After each recording, QtDMM provides a functionality to export the read data to a file.

The content from the exported file can be further processed by a script (Fig. 3.2) provided in the openSUSE Wiki[6]. Executing the script with the exported data file results in the accumulated power consumption for one specific test:

```
power@UUT:~/bin> ./arithmetic_mean.sh /tmp/X60_load
calculating arithmetic mean...done
Result: 22.10 W
```

Power Meter A description of a suitable power meter is provided in section 2.1.1.

3.3 Preparations

This checklist provides detailed steps describing how to properly prepare a UUT for each individual test. Make sure that this section is properly processed before starting the individual tests (Fig.**??**).

A sample sheet for documenting the collected information can be found in section 4.

Note: *The order of the test procedure elements is mandatory!*

1. Record manufacturer and the model name of the UUT.

2. Connect the UUT to a wired live network.

3. Connect the power meter to an AC source.

4. Plug the UUT into the power meter (*no* power strips or UPS units (Uninterruptible Power Supply) between the meter and the UUT).

5. Record the AC voltage.

6. Boot the UUT and wait until it is has fully loaded. It should be in an idle state afterwards.

7. Record basic information like the operating system's name and version; the processor's type, speed, and available physical main memory; etc.

8. Record basic information about the video card, like the amount of memory, if the memory is onboard or not, the bits per pixel or the resolution used.

[6] http://www.openSUSE.org/PowerMeasurements

9. Make sure that the UUT is configured as shipped by default.
10. Configure the system to power down the display after one minute.
11. Shut down the UUT.

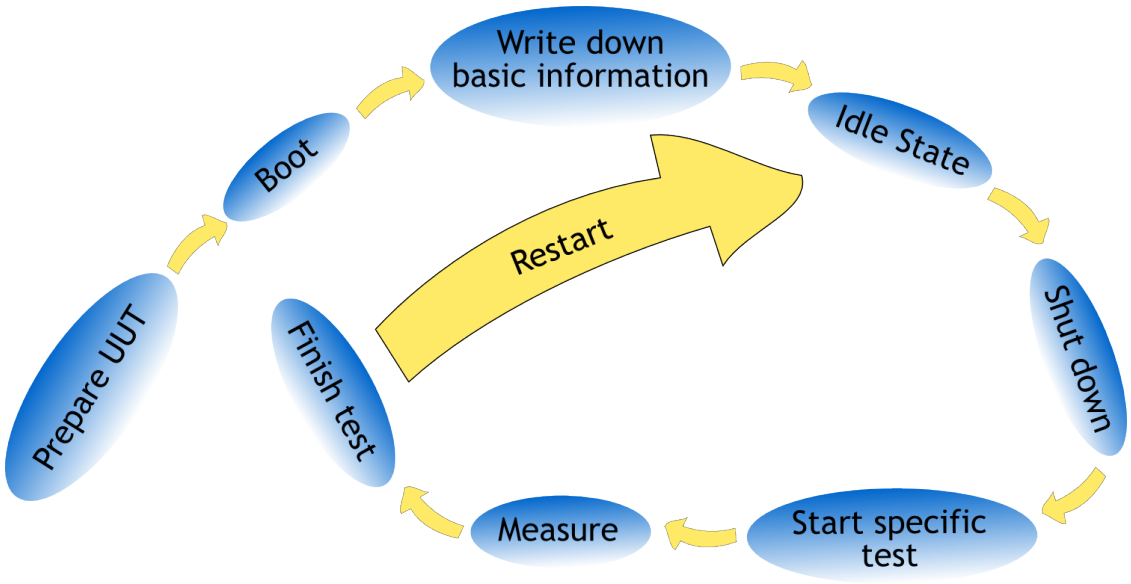

Figure 3: Test Procedure

3.4 Tests

The following test scenarios are mentioned in the EnergyStar Specification. They are not meant to deviate from it. However, additional tests, without any EnergyStar dependency, could be added in future, as part of another section.

3.4.1 Off Mode Tests

1. With the UUT still shut down and in off mode, start measurements. Accumulate power values for 5 minutes at an interval of one reading per second.
2. If testing with WOL, power up the UUT, enable WOL, power it down, and repeat the previous step.
3. Stop measuring. Export the data, calculate the arithmetic mean, and write down the result.

3.4.2 Idle Mode Test

1. Switch on the UUT and log in.
2. Wait until the machine has fully loaded. Close any windows so that the default desktop is displayed.
3. Wait 15 minutes to ensure no special tasks are running.
4. Display should be off (automatic display shut down). If that is not the case, it has to be switched off manually (such as with "xset dpms force off").
5. Start measurement for 5 minutes.
6. Stop measuring. Export the data, calculate the arithmetic mean, and write down the result.

3.4.3 Sleep Mode Tests

1. Put the computer in sleep mode.
2. Start measurement for 5 minutes.
3. If testing with both WOL enabled and WOL disabled, wakeup the UUT, enable/disable WOL, and repeat the previous two steps.
4. Stop measuring. Export the data, calculate the arithmetic mean, and write down the result.

4 Power Consumption Test Sheet

The example below shows what the results could look like. An empty result sheet for printing or editing can be found in the openSUSE wiki[7]. There is also a plain text file template available.

General

Model:	Example system
Manufacturer:	PowerManufacturer
CPU:	Mobile 64 bit, 800 Mhz
RAM:	480MB
Operating System:	openSUSE 10.3
AC Voltage:	230 V
Category:	A ☐ B ☐

GFX Card

Model:	Onboard graphics with shared memory
Memory:	32 MB shared
Resolution:	1024x768
Bits per pixel:	24

System Capabilities and Configuration

Configured as shipped:	x
Wireless network off:	x
Sleep mode:	x
Wake On LAN:	☐
Auto display sleep (15'):	x
Auto system sleep (30'):	x

Miscellaneous

Not able to determine if WOL was on or off when the system is shut down

Results

What	Result (W)	Required (W)	Passed
Off (WOL disabled)	0.5	<= 1.0	x
Off (WOL enabled)	1.9	<= 1.7	☐
Sleep (WOL disabled)	1.0	<= 1.7	x
Sleep (WOL enabled)	2.7	<= 2.4	☐
Idle	17	A: <= 14, B: <= 22	☐

EnergyStar compliant:	x

5 Final Notes

Although this document is based on the rudiments of EnergyStar, it is not meant to be a hard dependency. In fact, it is explicitly open to deviate from it where sensible. For instance, there could be a test which takes wireless networking and/or external devices into account, which of course would make sense. The whole document can be considered a “work in progress”. Any feedback or extensions are welcome. Add tests, notes, or any other comments. Just speak up! And not to be forgotten, special thanks goes to Noel Arnold for his review of this document.