

POWER CONSUMPTION MONITORING

INTRODUCTION

Power Consumption Monitoring is a Python-based script that uses PyQt5 framework and PowerLog, a command line version of Intel Power Gadget, to accurately measure the power consumption of individual applications on a computer. This tool functions by comparing power usage of the system at idle and during the application's operation.

DIRECTORY

All the files are under Measure directory. The entrance point of the application is main.py.

CODE BASE

1. Main.py

The entrance of the application. Contains all the User Interface (UI) design of the application, as well as the UI logic.

2. Intel.py

Intel.py contains the code to use Intel Power Gadget (PowerLog) to monitor the power consumption and generate reports, while also contains the code to read and process the data in the report.

3. Utility.py

Utility.py contains utility codes which helps to process data in the application.

4. Design.ui

Contains the design of the UI. It can only be opened using Qt Designer, which can be downloaded [here](#).

FUNCTIONAL DESCRIPTION

User Input Page

1. The application has a status bar, which will reflect as "Idle" when the application starts or when no tests are running.
2. The application will autofill the Intel power gadget directory when the application starts if it is installed.
3. The user can also click "Find Dependencies" button to let application fill in Intel power gadget directory if it is installed.
4. The user can input the intel power gadget directory and application directory. These directories must be a executable and must not contain quotation mark, for example: C:\Program Files\Mozilla Firefox\firefox.exe
5. If the user does not input the valid directory for both gadget directory and application directory, after the user clicks run, the application will not run the test, instead, the status bar will be updated to "Invalid Directory!".
6. The user can input how long they want the application to monitor the power consumption (**Measure time**). As the application will run two monitoring session in one test (one to monitor system at idle and one to monitor application's operation), the total time taken for the application to run will be 2x Measure Time.
7. The **measure time** is set to 60s by default.
8. The user can select whether they want, as well as how long they want to wait for the application to startup before monitoring the power consumption.
9. The application will not wait for startup if the checkbox is disabled.
10. If the user input valid directories and click run, the test will start, and the "Run" button will be disabled.

During Testing

1. The application will update status bar to reflect the current progress of the test. The test flow are as follows:

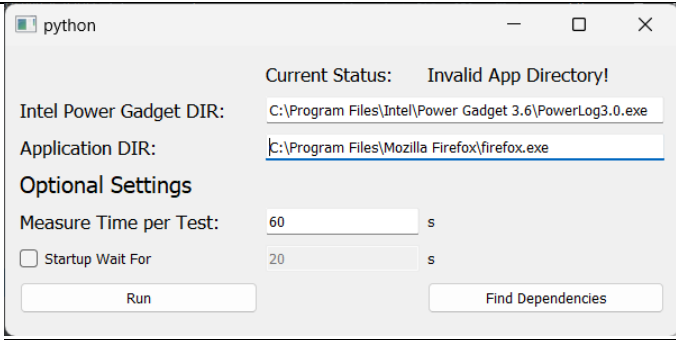
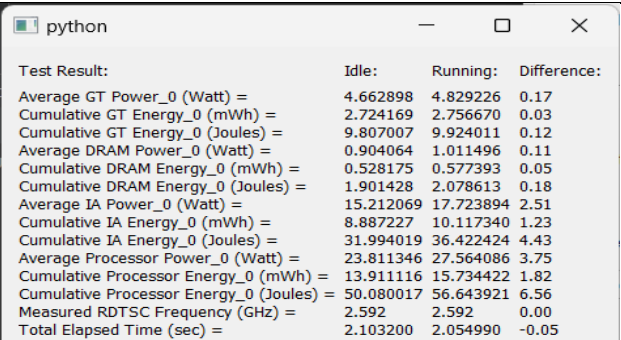
No.	Test Flow	Status Bar
1.	Measure the power consume <u>in idle state</u> in one measure time	"Start Measuring Power Consume in Idle State..."
2.	Start the test application	"Starting Test Application..."
3.	Measure the power consume <u>during application's operation</u> in one measure time	"Start Measuring Power Consume in Running State..."
4.	Processing the data in reports generated by Intel Power Gadget	"Generating Report..."
5.	Application will generate test report in a pop-up screen	(Pop Up Screen)
6.	Test finish. The application back to idle state	"Idle"

- The application will display power consumption data during idle state, power consumption data during application's operation and the difference between the two in a pop-up screen.
- If the difference is positive, it means the power consumption during application's operation is higher, vice versa.

Error code

- If the application that is being tested cannot start and is not executing, the status bar will be updated to "Start Test Application Fail. Terminating Test..."
- If the file that is supposed to be generated by the Intel Power Gadget does not exist in the same directory as the script, the status bar will be updated to "Log file is not found, analysis cannot be done!"

UI DESIGN

	
User Input Page	Pop-up Screen

USAGE

Prerequisite

- The code must be run on Windows 10/11. Mac may be possible, but it is not tested.
- Must have Python 3.10 installed on the machine. ([Installation Guide](#))
- Must have Intel Power Gadget installed on the machine. ([Installation Guide](#))
- Recommended to have Poetry installed on the machine. This is to install the dependencies mentioned in the pyproject.toml ([Installation Guide](#))

Steps (Assuming Poetry is installed)

- Unzip the file.
- Make sure that the machine is using Python 3.10
- In the terminal, type `poetry install` to install the dependencies.
- In the terminal, type `poetry run python main.py` to start the program.

*If you do not have Poetry install, you can still refer to pyproject.toml and download the dependencies manually using pip or something similar, just download all the dependencies listed under the section [tool.poetry.dependencies].