



ANUBIS-IDE DETAILED STUDY



AUGUST 20, 2020

AIN SHAMS UNIVERSITY – FACULTY OF ENGINEERING
1 Al-Sarayyat st, Abbasiya, Cairo 11517, Egypt

FACULTY OF ENGINEERING

AIN SHAMS UNIVERSITY

CSE426: Maintenance and Evolution



ANUBIS-IDE DETAILED STUDY

Submitted By:

Abdelrahman Ibrahim ELGhamry

16P3043@eng.asu.edu.eg

Ghamry1998@gmail.com

Submitted to:

Prof. Dr. Ayman Bahaa

AUGUST 20, 2020

A REPORT FOR SOFTWARE MAINTENANCE AND EVOLUTION COURSE CODDED CSE426 WITH THE
REQUIREMENTS OF AIN SHAMS UNIVERSITY

TABLE OF CONTENTS

1. PURPOSE.....	3
2. APPLICATION OVERVIEW	3
3. SCOPE	3
4. PROJECT INSTALLATION.....	4
4.1 Part (1).....	4
4.2 Part (2).....	4
5. TEST ANALYSIS AND ISSUES	5
6. RECOMMENDATIONS	6
7. SUMMARY AND CONCLUSIONS	6

1. PURPOSE

This document explains the various set of activities performed to investigate, analyze, and identify the key features, details and issues presented in Anubis-IDE.

2. APPLICATION OVERVIEW

Anubis-IDE is an open source desktop-based text editor, which aims to provide a simple environment to write, edit, compile, and run python scripts on various micro-controllers.

3. SCOPE

- **In-Scope:** Functional testing for the following modules are in scope of testing:
 - Scanning and analyzing (parsing) different input python tokens.
 - Compilation of the input code sequence.
 - Running of the compiled code on a micro-controller.
 - Saving input code to a new file.
 - Opening and editing an existing python file.
 - Port connectivity and code flushing.
- **Out of Scope:** Performance Testing and Stress Testing was not done for this application.
- **Items not tested:** Verification of external packages as QT5, which is responsible for building the UI, and PySerial which is responsible for the connectivity between the host and our micro-controller.

4. PROJECT INSTALLATION

Firstly, we need to get our environment ready to run the project, then we need to clone the project repo. Follow up the following steps for more details:

(Ps: if your environment is ready to run a python project, skip part (1))

4.1 Part (1)

1. Install any text editor (VS Code is recommended)
2. Install Python (3.0 or above is recommended)
3. Add Python's extension to your vs code.

4.2 Part (2)

1. Clone the project repo from the following link
<https://github.com/a1h2med/Anubis-IDE>
2. Open the project directory using any text editor (VS Code is recommended).
3. Open the terminal and run the following command(s) to download the project dependencies (packages):

```
$ pip install -r requirements.txt
```

Or

```
$ pip install pyserial
```

```
$ pip install PyQt5
```

4. Run "Anubis.py" using your text editor.

5. TEST ANALYSIS AND ISSUES

1. The code does not perform its main task (compiling and flushing code to the micro-controller).

Steps:

- a. Attach a micro-controller to your device.
- b. Go to the Ports Menu and select your plugged micro-controller port.
- c. Click Run.
- d. The app gives the corresponding message
"Sorry, there is no attached compiler."

2. App crashes on opening any file other than text files.

Steps:

- a. Select a directory or an image using the tree view or the finder.
- b. The app freezes for 1 second then it crashes.

3. "Requirements.txt" file which is responsible for saving dependencies (used packages) names and versions includes unused packages as (certify, chardet, future, idna, iso8601, PyYAML, requests and urllib3).

4. The user is always able to use save button even if the text editor is empty (no code is entered).

5. Tab name does not change to the selected file name (from the tree view) or even after saving the new code to the "main.py" file. Instead, it remains "tab1".

6. There is no chance to add another tabs.

7. Code is unstructured, unreadable and contains unused variables. (eg: *vbox* at line 152 *Anubis.py*) Which makes this code very hard to maintain and evolve contradicting FOSS rules or guidelines.

6. RECOMMENDATIONS

1. Use an external package (as Python Compiler) to compile the entered code.
2. Validate and disable opening any file other than text files either by restricting the viewed files in the tree view to “.txt” and “.py” files or by checking the selected file extension before opening it.
3. Delete unused packages from “Requirements.txt”.
4. Check if the file is empty or not before saving it.
5. Set the tab name to the selected file name instead of “tab1”.
6. Unify variable names or change them to descriptive ones and use only one naming case (*eg: camel case*).
7. Split your code classes on different files to assist scalability.

7. SUMMARY AND CONCLUSIONS

The mentioned software needs more work on various aspects, its code must be refactored, requirements must be fulfilled, and design documents must be provided in order to achieve readability, maintainability, reliability and scalability.