**Infineon IFAP DC - AMSA File Processing User Documentation**

**Table of Contents**

1. Introduction
2. Getting Started
3. Generating the Executable
4. Using the Program
5. Main Window
6. Program Workflow
7. Modifiable Variables
8. Troubleshooting

**Introduction**

This program is a graphical user interface (GUI) application designed to process ACC Folders and perform various data manipulations for calibration purposes and to create a new ACC Folder with the calibrated data.

This documentation provides step-by-step instructions on how to use the program effectively and also code snippets that are to be changed if needed.

The program runs a virtual environment of Python without the use of any Integrated Development Environment (IDE) such as VSCode or Python IDLE through the use of PyInstaller to create an executable (.exe) file.

**Getting Started**

The executable does not require any installation of software or setup of any installation files. Essentially, there are no prerequisites to run the software. The executable is a standalone and will run alone without any dependencies.

The executable is located in the below directory in the shared folder:

**Z:\rfs\_simic\ACC Folder Generator\AMSA\_ACCFolderGeneration.exe**

However, if you are running it through an IDE (eg. VSCode), the script requires the following libraries:

**tkinter, pandas, numpy, matplotlib, xlsxwriter, shutil, pyinstaller**

which can be installed using the below command in Anaconda Prompt/VSCode Terminal:

**pip install tk, pandas, numpy, matplotlib, xlsxwriter, shutil, pyinstaller**

The core directory structure requirement should contain the following:

1. **AMSA\_ACCFolderGeneration.exe**
2. **Acoustic\_Chambers\_Calibration\_Data**

This is because the program uses the contents of the ACC folder and copies it into a cache before creating the new ACC Folder directory.

**Generating the Executable**

Generating the executable requires the use of the **pyinstaller** library, and a Command Line to generate the .exe file of the Python script.

PyInstaller bundles a Python application and all its dependencies into a single package. Hence, the user can run the packaged app without installing a Python interpreter or any modules. PyInstaller supports Python 3.7 and newer, and correctly bundles many major Python packages such as numpy, matplotlib, PyQt, wxPython, and others.

**Please ensure that the Source Codes are in different directories (folders).**

To enable the use of PyInstaller, run the following command in the VSCode terminal/Anaconda Prompt:  
  
**pip install pyinstaller**

To generate the .exe file from the .py file, run the following command with the directory of the source code open in the left sidebar:  
  
**pyinstaller --onefile <filename>.py**

With this, PyInstaller will build several directories within the parent directory where the .py file is located. Within the **dist** subdirectory, there will be a file named <filename>.exe

<filename.exe> is the standalone executable file that will be distributed and run the Python script for Data Analysis and Automation.

The pathlib library might interfere with the generation of the file, hence you may run the below command to remove the conflicting package:

**conda remove pathlib**

**Using the Program**

**Main Window**

Upon launching the program, the main window will appear, displaying the program logo, description, and instructions.

**Product Selection**

In the main window, a dropdown menu labeled "Select Product" is available. Use this dropdown to choose the product you are working with (e.g., Akari or Fuji).

**AMSA File Processing**

Click the "Upload Raw Data (GD AMSA) file (.xlsx)" button to upload the raw AMSA data file (in Excel format) from the Golden Devices.

The selected file path will be displayed below the button.

**Loading GD DMC Code**

Click the "Upload GD DMC Code (.xlsx)" button to upload the DMC Code of Golden Devices file (in Excel format).

The selected file path will be displayed below the button.

**Generating the Temp AMSA1.xlsx file**

Click the "Process Raw Data and GD Selection" button to initiate AMSA file processing.

**Loading Calibration Setup File**

Click the "Upload CalibrationSetupFile (.csv)" button to upload the CalibrationSetupFile (in CSV format).

The selected file path will be displayed below the button.

Click the "Process CalibrationSetupFile" button to start processing the CalibrationSetupFile.

**AMSA2 Graph Window**

After the "Process CalibrationSetupFile" button is clicked, a new window should open showing the graph plotted from the first two columns of AMSA2.xlsx. This shows the variance in the sensitivity from the target value per site.

The target sensitivity is also highlighted with a blue line for greater visibility and the Deviation is calculated using the difference between the maximum and the minimum points.

An illustration of the graph window can be seen in the next page (Figure 1).

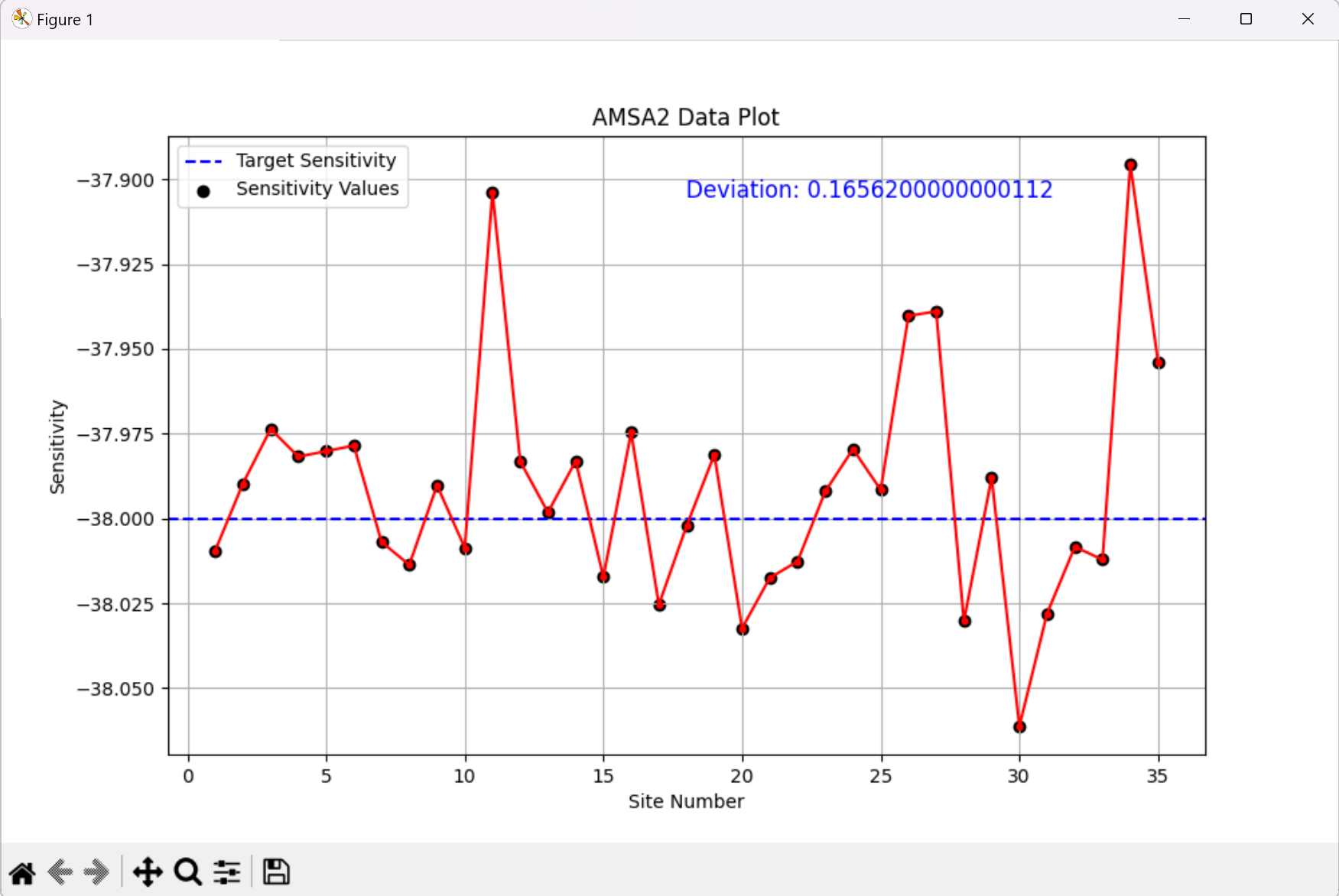


Figure : AMSA2 Data Plot

**Program Workflow**

**AMSA File Processing Workflow**

1. Select the product (Akari or Fuji).
2. Upload the raw AMSA data file.
3. Upload the GD file.
4. Click the "Process Raw Data and GD Selection" button.
5. The program will split the data, create output files, and generate a temporary AMSA1.xlsx file, which is the cleaned version of the raw data file.

**Calibration Setup File Processing Workflow**

1. Upload the CalibrationSetupFile (CSV format).
2. Click the "Process CalibrationSetupFile" button.
3. The program will generate new target values for different sheets in the AMSA2 file and will also perform amplitude and phase compensation.

Ultimately, the program produces a new folder called **“Acoustic\_Chambers\_Calibration\_Data\_new”** which is to be used for calibration purposes.

**Modifiable Variables**

There are several variables that can be modified from the code and code snippets will be provided below to illustrate the change in variables and its effects.

**deleteTrigger**

At line 16 - 17, there is a global Boolean variable called deleteTrigger. When set to True, it will delete all intermediate and temporary files that are generated.

The code snippet involved is the below function:

def delete\_intermediate\_files():

os.remove("AMSA.xlsx")

os.remove("AMSA1.xlsx")

os.remove("AMSA2.xlsx")

os.remove("New\_ini\_Target.xlsx")

os.remove("output\_Phase.xlsx")

os.remove("output\_spk.xlsx")

**Product Addition and Management**

At **Line 39**, there is a function called update\_data\_columns. This function enables the developer to add more raw data column headers dictionary pairs to loc/iloc using pandas.

The developer can use more **“elif (else if) == ‘<productName>’:**” to create more sets of data columns.

If there is any uncertainty, do refer to the initial “**if product == ‘Akari’:”** statement as a reference point for the addition of other products.

Additionally, at **Line 518**, there is a line that controls the number and names of products that can be selected from the drop down box:

**values=['Select Product', 'Akari', 'Fuji'], # Adding a placeholder**

The developer is able to modify the values by adding more strings in this manner coloured in red:

**values=['Select Product', 'Akari', 'Fuji', ‘product1’, ‘product2’],**

This enables the developer to use this program to process more products.

**Troubleshooting**

If you encounter any issues while using the program, please check the following:

1. Ensure you have met the system requirements and prerequisites.
2. Verify that the uploaded files are in the correct format.
3. Check for error messages in the program console.
4. Ensure that the folder directory contains the ACC Folder.