IPTC SAS – SOFTWARE GUIDE

This is a general overview of the generated software. For a more detailed explanation of each function, read the annotations in the code and the flow diagrams in the thesis.

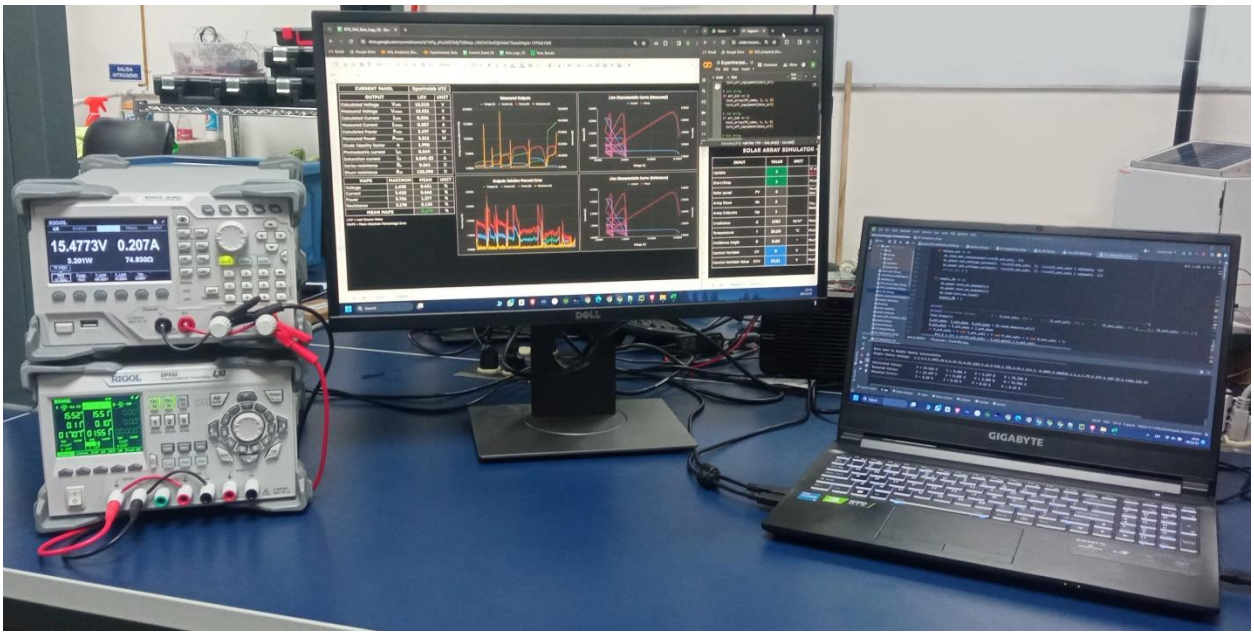
Component #1

Python File	Description
main	Uses functions from the other modules to extract datasets from pictures and calculate the single-diode parameters using the Hussein method. Alternatively, it can import .CSV files with already generated solar panels' datasheets. Also generates various plots from I-V and P-V characteristic curves under reference conditions and through a range of external conditions.
PV_Single_Diode	Contains functions related to the implementation of the Hussein method to calculate a solar panels' single-diode unknown parameters. Contains functions related to the scaling of parameters to different external conditions (i.e. irradiance, temperature of cell, incidence angle).
PV_functions	Contains the Computer Vision algorithm to extract datasets from images. Contains the primary functions called on the main, which use functions from PV_Single_Diode, PV_Plotting and PV_CSV. Creates classes for each logged solar panel.
PV_Plotting	Contains functions related to the generation of plots.
PV_CSV	Contains functions related to the generation of .CSV files that contain datasets or datasheets.

Component #2: Platform #1 (computer + RIGOL instruments)

Python File	Description
main	Runs an infinite loop where it constantly checks the IoT control panel for new inputs. If the user has set the *Start* to 1, and all the inputs are valid, it calculates the operational point of the solar panel and sets it in the instruments. Then, it saves the logs in both the IoT and local databases.
PV_Model	Contains the functions used to calculate an operational point on the panel by solving the single-diode equation. The single-diode parameters at reference conditions are known, and are scaled to different external conditions if needed.
RIGOL_lib	Contains functions that use PyVISA to send commands to the RIGOL instruments.
Turn_OFF_RIGOL	Simply executes a chain of commands to set the power supply outputs to zero.
Google_Sheets	Contains the necessary functions to read the IoT platform's control panel messages and to send logs to the cloud database.
Secrets	Contains the Google Apps Script's gas IDs for the control panel and the database.
SAS_Tests	Contains functions that generates datasets for experimental tests using this Google Colab Notebook.
Dataset_Diff_Array_Sizes_Test	Creates a dataset for different array sizes at the panel's reference conditions. You need to set the desired panel in the control panel.
Dataset_Diff_Conditions_Test	Creates a dataset in which the Maximum Power Point is tracked to the whole range of the external conditions. You need to set the desired panel in the control panel.
Plot_Curves_All_Arrays	Given you already have raw data for the arrays tests, it cleans it (meaning that for every single operational point that has multiple logs, it calculates the average values so that there's a single log for every point), then crates a .CSV file where the obtained errors are analyzed (maximum and minimum ARPE, MAPE) and then creates two plots for each test, one of the I-V/P-V curve and one of the ARPE curve.

Plot_Curves_Diff_Conditions	Does essentially the same that *Plot_Curves_All_Arrays* does but with the raw data of the different conditions tests.
Error_Analysis_All_Arrays	Analyzes the errors of every single test for different array sizes, generating tables that summarize the results.
Error_Analysis_Diff_Conditions	Analyzes the errors of every single test for different array sizes, generating tables that summarize the results.



Component #2: Platform #2 (microcontrollers)

.INO File	Description
IPTC_SAS_ESP8266	Receives data form the IoT control panel and sends it to the Nucelo 64. Receives logs from the Nucleo 64 and sends it to the IoT database.
IPTC_SAS_Nucleo64	Receives user's inputs and solar panel's data from the ESP8266. Calculates the operational point of the solar panel and sends it to the MCP pant analog. Receives "sensor readings" from the MCP analog. Sends logs to ESP8266 and saves them in a MicroSD card. Prints outputs in and LCD1602 display.
IPTC_SAS_MCP_Plant_Analog	Receives operational point from the Nucelo 64. Adds random noise equal to or below 1% of its reference value. Sends the noisy value to the Nucelo 64.