**Reminder:**

1. Please do not remove the Google Spreadsheet ‘Solar Irradiance Data’ in the IoT subgroup file.
2. Basically, we just only need to run the ‘Code1’ to obtain the data such as global irradiance, max global irradiance, temperature and humidity. Then, the data will be sent to Google Spreadsheet ‘Solar Irradiance Data’ around every 1 minute as long as the program doesn’t execute error and the escape key is not clicked.
3. Initially, ‘Code2’ should be used to obtain the measurements from online source ‘<https://www.ema.gov.sg/solarmap.aspx>’ by applying colour detection. Meanwhile, I found another webpage ‘<https://www.solar-repository.sg/local-weather>’ that can provide more accurate information. As such, I decide to obtain the measurements from this website by applying character recognition. It may cause errors when the captured image is blurred and damaged. After setting up well, ‘Code1’ perform in good condition so far.
4. All the codes will only be coordinated with Cayenne IoT. It has nothing to do with Arduino IDE and its board.
5. ‘Code1’ will be only worked well in 15.6’’ screen sized pc since the adjustment for the cropping of specific details are manually set.
6. The implementation of ‘Code1’ still have not completed yet since the connection to Cayenne Dashboard is not tested yet to check if the code performs well in terms of the connection.

**Procedure:**

1. Run the program using Pycharm or any Python IDE
2. Download the packages according to the instruction in the beginning of the codes
3. Click Run and wait until windows pop out, then simply observe it
4. Review the uploaded data on the Google Spreadsheet ‘Solar Irradiance Data’ but be careful that once you run the program, the data on the spreadsheet will be refreshed. If you want to record down the data, remember to download it as xlsx file to your pc
5. Only click ‘ESC’ to exit the program